Knowledge-Based Systems 133 (2017) 294–313

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

# An entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood

Shifei Ding[a,*], Mingjing Du[a], Tongfeng Sun[a], Xiao Xu[a], Yu Xue[b]

[a] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China
[b] School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China

ARTICLE INFO

*Article history:*
Received 27 October 2016
Revised 4 June 2017
Accepted 20 July 2017
Available online 21 July 2017

*Keywords:*
Entropy
Density peaks clustering
Mixed type data
Fuzzy neighborhood

ABSTRACT

Most clustering algorithms rely on the assumption that data simply contains numerical values. In fact, however, data sets containing both numerical and categorical attributes are ubiquitous in real-world tasks, and effective grouping of such data is an important yet challenging problem. Currently most algorithms are sensitive to initialization and are generally unsuitable for non-spherical distribution data. For this, we propose an entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood (DP-MD-FN). Firstly, we propose a new similarity measure for either categorical or numerical attributes which has a uniform criterion. The similarity measure is proposed to avoid feature transformation and parameter adjustment between categorical and numerical values. We integrate this entropy-based strategy with the density peaks clustering method. In addition, to improve the robustness of the original algorithm, we use fuzzy neighborhood relation to redefine the local density. Besides, in order to select the cluster centers automatically, a simple determination strategy is developed through introducing the $\gamma$-graph. This method can deal with three types of data: numerical, categorical, and mixed type data. We compare the performance of our algorithm with traditional clustering algorithms, such as K-Modes, K-Prototypes, KL-FCM-GM, EKP and OCIL. Experiments on different benchmark data sets demonstrate the effectiveness and robustness of the proposed algorithm.



## 1. Introduction

Clustering analysis is aimed at finding correlations within subsets of the dataset and assessing similarity among elements within these subsets [1,2]. Clustering has many applications in various domains including biology, economics and medicine. Its applications include data mining, document retrieval, image segmentation, and pattern classification [3,4]. Traditional clustering methods, e.g., K-Means [5], can only handle numerical values. Nevertheless, in some real world applications, one has to deal with features, such as gender, color, and type of disease that are categorical attributes. In other words, data sets containing both numerical and categorical attributes are ubiquitous in real-world tasks. Designing an effective clustering algorithm for this type of data is a challenging problem. For convenience, we use the "mixed type" data to denote this type of data with numerical and categorical attributes in this paper. But the mixed type data may contain ordinal attribute or some other attributes in other literatures.

A straightforward way to deal with mixed type data has a preprocessing that is able to convert categorical attributes to new forms, e.g. the binary strings, and then apply the aforementioned numerical value based clustering methods. However, binary encoding has three drawbacks. First and foremost, this method destructs the original structure of categorical attributes. In other words, transformed binary attributes are meaningless and their values are hard to interpret [6]. Second, if the domain of a categorical attribute is large, then transformed binary attributes will have a much larger dimensionality. The last disadvantage is the difficult of maintenance. If an attribute value is added into a categorical attribute, all of the objects will be changed. In order to better solve the problem, numerous researchers study on clustering based on similarity metrics dealing with categorical values directly, during the last decade. Based on a similarity (or dissimilarity) metric that takes into account both numeric and categorical attributes, some methods, e.g., K-prototypes (KP) [7] and its variations which are applicable to numerical and categorical data are presented. In order to circumvent parameter adjustment between categorical and numerical values, some works, e.g., Similarity-Based Agglomerative Clustering (SBAC) algorithm [6], based on a new similarity metric for mixed type data, are presented. However, SBAC is high compu-

* Corresponding author.
*E-mail address:* dingsf@cumt.edu.cn (S. Ding).

http://dx.doi.org/10.1016/j.knosys.2017.07.027
0950-7051/

tational complexity. Some methods based on a parameter-free similarity metric, e.g. OCIL [8], are proposed. But this metric only can measure the similarity between an object and a cluster. And like KP and its variations, OCIL uses the K-Means paradigm to cluster mixed type data and is an iterative clustering algorithm. Hence, such kind of algorithms is sensitive to initialization and is more suitable for spherical distribution data.

In this paper, we propose a novel density peaks clustering algorithm for mixed type data. Firstly, we propose a new similarity measure for either categorical or numerical attributes which has a uniform criterion. This similarity measure is proposed to avoid feature transformation and parameter adjustment between categorical and numerical values. Subsequently, to better handle data with the non-spherical distribution, a density-based data clustering, peak density (DP) clustering algorithm [9], is introduced. The algorithm is able to detect non-spherical distribution data and does not need to pre-assign the number of clusters. We integrate this entropy-based strategy with the DP clustering method. Moreover, to improve the robustness of DP, we use fuzzy neighborhood relation to redefine the local density, which integrates the speed of DP clustering algorithm with the robustness of FJP algorithm [10]. We also develop an automatic cluster center selection method. On the basis of these strategies, we develop a density-based clustering algorithm for mixed type data employing fuzzy neighborhood (DP-MD-FN). Additionally, in order to demonstrate the feasibility, robustness and scalability, we conduct some experiments on synthetic data sets. In order to assess the performance of the proposed algorithm, we compare the proposed algorithm with other algorithms on some UCI data sets. As a result, our algorithm has achieved satisfactory results in most data sets.

The rest of this paper is organized as follows. Related works are introduced in Section 2. Section 3 sets down notations and describes the original DP clustering method. We make a detailed description of DP-MD-FN in Section 4. Section 5 presents our experimental results on synthetic data sets and real data sets. Finally, Section 6 presents conclusions and future works.

## 2. Related works

This section introduces the related works on: (1) clustering for mixed type data and (2) density peaks clustering and (3) fuzzy joint points.

Firstly, we review traditional clustering algorithms for mixed type data. Some methods have a pre-processing that is able to convert categorical attributes to new forms and facilitates processing. For example, Ralambondrainy's algorithm [11] transforms categorical attributes into a set of binary attributes. Then, new forms are treated as numeric in the K-Means algorithm. Hence, we can directly adopt most traditional distances which are often used in numerical clustering, such as Euclidean distance, to define similarity between transformed objects. However, as stated before, the primary disadvantage of the binary encoding is that it cannot reveal the structure of the data sets. Apart from binary encoding, there are also other pre-processing methods. For example, Hsu [12] presents a new mechanism, distance hierarchy, which encodes a data set into a weighted tree structure. In addition, Hsu et al. [13] introduce this mechanism into hierarchical clustering for mixed type data. But it has a drawback that both the assignment of weights and the construction of distance hierarchies rely on domain knowledge.

Unlike the pre-processing methods, some works try to find a unified similarity metric for categorical and numerical attributes. Along this line, some clustering algorithms based on a unified similarity metric for categorical and numerical attributes are proposed, during the last decade. Among them, K-prototypes (KP) algorithm [7] is one of the most famous clustering algorithms for mixed type data. The algorithm is an extension of K-Modes [14] which handles categorical data by using a simple matching dissimilarity measure for categorical objects. Simple matching is compared with two categorical values according to a matching function. The result is 1 if the two values are different or 0 otherwise. To avoid favoring either type of attribute, Huang introduces a weight $\gamma$ into K-Prototypes. Some variations of K-Prototypes, such as KL-FCM-GM [15], WFK-prototypes [16], perform fuzzy partition by combining K-Prototypes with fuzzy c-means (FCM)-type clustering. In addition, Zheng et al. [17] integrate evolutionary computation framework with KP and propose an unsupervised evolutionary clustering algorithm for mixed type data, evolutionary K-Prototypes algorithm (EKP). Experimental results show that the parameter $\gamma$ has a great influence on these algorithms. Hence, it comes out that choosing the parameter is a delicate and difficult task for users that may be a roadblock for using K-Prototypes and its variations efficiently. In order to circumvent parameter adjustment between categorical and numerical values, OCIL [8] gives a unified similarity metric which can be applied to mixed type data using the entropy-based criterion. This similarity metric is also based on the concept of object-cluster similarity. In other words, it can measure the similarity between an object and a cluster rather than that between objects. Li et al. propose a Similarity-Based Agglomerative Clustering algorithm [6] based on a new similarity metric which deals with the mixed type data. This similarity metric is proposed by Goodall for biological taxonomy. It assigns a greater weight to uncommon feature value matching in similarity computations without assuming the underlying distributions of the feature values. But the method is high computational complexity. Most of these methods use the K-Means paradigm to cluster mixed type data and have an iterative process. Hence, they are sensitive to initialization and are generally unsuitable for non-spherical distribution data.

Secondly, Density peaks clustering [9] and its variations are introduced. Density peaks clustering, a density-based algorithm, is proposed by Rodriguez and Laio. Unlike traditional density-based clustering methods, the algorithm can be considered as a combination of the density-based and the centroid-based. It starts by determining the centroids of clusters according to two important quantities: $\rho$ and $\delta$. The second step is to determine which objects to merge in a cluster. Unlike traditional density-based clustering methods, it is based on the local density of objects. All objects are in descending order in terms of the local density. An unclassified object is assigned to the cluster that contains a certain classified object satisfying a condition. It is the nearest of all the classified objects to the unclassified object. Similarly to other density-based methods, DP clustering algorithm is able to recognize clusters with arbitrary shape. Density peaks clustering algorithm has been applied to a variety of applications, such as anomaly detection, image segmentation, community detection, and so on [18–20], because it has the advantage of being convenient to implementation and computation. For example, inspired by this, Ma et al. propose the LED algorithm [21], which is based on Structural Clustering, which converts structural similarity between vertices to weights of network. However, there are still some shortcomings. For example, DP algorithm cannot find the correct number of clusters automatically. In order to overcome this difficulty, Liang et al. [22] propose the 3DC clustering based on the divide-and-conquer strategy and the density-reachable concept. DP only has taken the global structure of data into account, which leads to missing many clusters. In order to overcome this problem, Du et al. [23] propose a density peaks clustering based on k nearest neighbors (DPC-KNN) which introduces the idea of k nearest neighbors (KNN) into the original and has another option for the local density computation. To improve the running speed of DP algorithm, Xu et al. [24] propose a novel approach based on grid, called density peaks clustering algorithm based on grid (DPCG).

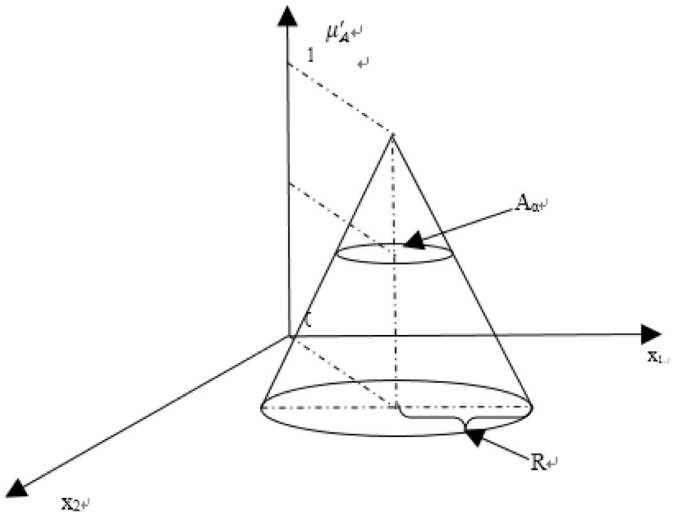



**Fig. 1.** Fuzzy point $A = (a, R) \in F(\mathbb{R}^2)$ in the space $\mathbb{R}^2$.

Finally, we introduce the Fuzzy Joint Points (FJP) method. Fuzzy clustering has constituted, in the past, a domain of research in the framework of cluster analysis. In these methods, the Fuzzy c-Means (FCM) algorithm is perhaps the most important and widely used method. Nasibov and Ulutagay [10] propose a different approach of fuzziness based on a new Fuzzy Joint Points method which perceives the neighborhood concept from a level-based viewpoint which means that the objects are considered in how much detail in construction of homogenous classes. It means that the fuzzier the objects are, more similar they are. Based on this approach, many clustering methods [25,26] are proposed. In these methods, FJP algorithm is robust since it uses fuzzy relation in neighborhood analysis.

## 3. Preparation

### 3.1. Notations

$\mathbf{x}_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,M}]$ is an object with $M$ numerical attributes. Let $\mathrm{d}(\mathbf{x}_i, \mathbf{x}_j)$ denote the Euclidean distance between the object $\mathbf{x}_i$ and the object $\mathbf{x}_j$, as follows:

$$\mathrm{d}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2. \tag{1}$$

From information theory [27], entropy has generally been associated with the amount of order or disorder in the system [28]. It means that entropy is more for a room with socks strewn all over the floor, and less for a room with socks placed in an underwear drawer [29]. More formally, entropy is the highest in the case that the probability of the variable obeys the uniform distribution. By contrast, the entropy are low when the bell-shaped histogram of the probability is "tall" and "thin" [30].

We assume that $x$ is a discrete random variable belonging to a finite set $V = \{x_1, x_2, \cdots, x_r\}$ and $p(x)$ is the probability mass function of the discrete random variable $x$. The entropy $E(x)$ is defined as

$$H(x) = -\sum_{x \in V} p(x) log(p(x)). \tag{2}$$

Some basic concepts about the fuzzy joint points method will be given later. Let $F(\mathbb{R}^m)$ denote the set of $m$-dimensional fuzzy sets of the space $\mathbb{R}^m$. $\mu : \mathbb{R}^m \to [0, 1]$ denotes the membership function of the fuzzy set $A \in F(\mathbb{R}^m)$.

A conical fuzzy point $A = (a, R) \in F(\mathbb{R}^m)$ of the space $\mathbb{R}^m$ is a fuzzy set with membership function (Fig. 1)

$$\mu_A^{'}(x) = \begin{cases} 1 - \dfrac{d(a,x)}{R}, & if\ d(a,x) < R \\ 0, & otherwise \end{cases}, \tag{3}$$

where $a \in \mathbb{R}^m$ is the center of fuzzy point $A$, and $R$ is the radius of its support supp A, where

$$\text{supp A} = \left\{ x \in \mathbb{R}^m \middle| \mu_A^{'}(x) 0 \right\}. \tag{4}$$

The $\alpha$-level set of conical fuzzy point $A = (a, R)$ is calculated as

$$A_\alpha = \{x \in \mathbb{R}^m > a\} = \{x \in \mathbb{R}^m | d(a,x) < R \cdot (1 - a)\}. \tag{5}$$

In this study, the short term "fuzzy point" is used, instead of conical fuzzy point defined in Formula (3).

### 3.2. Density peaks clustering

Unlike DBSCAN, the DP clustering finds the cluster centers before data points are assigned. Determining the cluster centers is of vital importance to guarantee good clustering results. Because it determines the number of the clusters and affects the assignation indirectly. In the following, we will describe the calculation of $\rho_i$ and $\delta_i$ in much more detail.

DP represents data objects as points in a space and adopts a distance metric, such as Formula (1), as a dissimilarity between objects. Let $\mathrm{D} = \{ds_1, ds_2, \cdots, ds_{N_d}\}$ be a set of all the distances between every two points in data set, where all the distances from smallest to greatest. $N_d = \binom{n}{2}$, where $N$ is the number of points in the dataset. Unlike DBSCAN, the neighborhood radius is determined by the percentage rather than by the direct value. $d_c$ indicates a percentage and is the only input parameter, which is called a cutoff distance. The neighborhood radius $\varepsilon$ is defined as:

$$\varepsilon = ds_{\lceil ds^{max} \cdot d_c \rceil}, \tag{6}$$

where $\lceil \cdot \rceil$ is the ceiling function and $ds^{max} = \max(d(\boldsymbol{x}_i, \boldsymbol{x}_j))$.

The neighborhood set of point $\boldsymbol{x}_i \in \mathrm{X}$ with parameter $\varepsilon$ ($\varepsilon$-neighborhood set) is as follows:

$$\mathrm{N}(\boldsymbol{x}_i, \varepsilon) = \left\{ \boldsymbol{x}_i \in \mathrm{X} | d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) < \varepsilon \right\}. \tag{7}$$

$\mu_{\boldsymbol{x}_j}(\boldsymbol{x}_j)$ denotes the membership degree of the point $\boldsymbol{x}_j$ to the neighborhood set of the point $\boldsymbol{x}_i$, as follows:

$$\mu_{\boldsymbol{x}_j}\left(\boldsymbol{x}_j\right) = \begin{cases} 1, & if\ d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) < \varepsilon \\ 0, & otherwise \end{cases}. \tag{8}$$

The local density $\rho_i$ of a point $\boldsymbol{x}_i$ is defined as:

$$\rho_i = \sum_j \mu_{\boldsymbol{x}_j}\left(\boldsymbol{x}_j\right). \tag{9}$$

The calculation of the delta value, again, is quite simple. The minimum distance between the point of $\boldsymbol{x}_i$ and any other points with higher density, denoted by $\delta_i$ is defined as

$$\delta_i = \begin{cases} \min\limits_{j:\rho_i < \rho_j} d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right), & if \exists j\ s.t. \rho_i < \rho_j \\ \max\limits_j d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right), & otherwise \end{cases}. \tag{10}$$

When the local density and delta values for each point have been calculated, this method identifies the cluster centers by anomalously large $\rho_i$ and $\delta_i$. On the basis of this idea, cluster centers always appear on the upper-right corner of the decision graph. After cluster centers have been found, the DP clustering assigns remaining points to the cluster to which its nearest neighbors with higher density belong.

## 4. The proposed algorithm

We present a new similarity measure as a unified framework for handling mixed type data with numerical and categorical attributes. To be more suitable for non-spherical distribution data, we introduce the similarity metric into the density peaks clustering algorithm for clustering data. Furthermore, to improve the robustness of the original DP method, we use fuzzy neighborhood relation to redefine the local density. Finally, to correctly find the number of clusters, we develop an automatic cluster center selection method.

### 4.1. Similarity measure

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ denote a dataset of $N$ mixed data objects, where for each $\mathbf{x}_i$, $i$ subjects to $1 \le i \le N$. A mixed data $\mathbf{x}_i$ is represented by $M$ $(M = M_n + M_c)$ attributes $A_1^{(n)}, A_2^{(n)}, \cdots, A_{M_n}^{(n)}, A_{M_n+1}^{(c)}, A_{M_n+2}^{(c)}, \cdots, A_{M_n+M_c}^{(c)}$, where $A_1^{(n)}, A_2^{(n)}, \cdots, A_{M_n}^{(n)}$ are the $M_n$ numerical attributes and $A_1^{(c)}, A_2^{(c)}, \cdots, A_{M_c}^{(c)}$ are the $M_c$ categorical attributes. $x_{i,k}^{(n)}$ denotes the $k$th attribute of $\mathbf{x}_i^{(n)}$, where $\mathbf{x}_i^{(n)}$ is the numerical part. $x_{i,k}^{(c)}$ denotes the $k$th attribute of $\mathbf{x}_i^{(c)}$, where $\mathbf{x}_i^{(c)}$ is the categorical part. $x_{i,k}^{(n)}$ $(1 \le k \le M_n)$ belongs to $\mathbb{R}$. And $\mathrm{DOM}(A_k^{(c)})$ $(1 \le k \le M_n)$ denotes the domain of $x_{i,k}^{(c)}$. That is, the categorical domain is denoted by $\mathrm{DOM}(A_k^{(c)}) = \{a_{k,1}, a_{k,2}, \cdots, a_{k,r_k}\}$, where $r_k$ is the number of category values of the $k$th categorical feature. Therefore, we represent $\mathbf{x}_i$ as a vector $.[\mathbf{x}_i^{(n)}, \mathbf{x}_i^{(c)}] = [x_{i,1}^{(n)}, x_{i,2}^{(n)}, \cdots, x_{i,M_n}^{(n)}, x_{i,M_n+1}^{(c)}, \cdots, x_{i,M}^{(c)}]$.

Unlike conventional similarity measures, we treat the similarity on the numerical part as whole (i.e., the numerical attributes are treated as a vector and handled together), but calculate the similarity on the categorical part individually. As a result, the dimensionality of $\mathbf{x}_i$ is $M_n + M_c$, but the number of attributes that is used for measuring similarity between objects is $1 + M_c$. Thus we use 1 numerical vector and $M_c$ categorical attributes to compute similarity. Thus the similarity between objects $\mathbf{x}_i$ and $\mathbf{x}_j$ can be denoted as:

$$\begin{aligned} \mathrm{S}(\mathbf{x}_i, \mathbf{x}_j) &= \frac{1}{M_c+1}\Big(\mathrm{S}_n(\mathbf{x}_i^{(n)}, \mathbf{x}_j^{(n)}) + S'_c(x_{i,1}^{(c)}, x_{j,1}^{(c)}) \\ &\quad + S'_c(x_{i,2}^{(c)}, x_{j,2}^{(c)}) + \cdots + S'_c(x_{i,M_c}^{(c)}, x_{j,M_c}^{(c)})\Big) \\ &= \frac{1}{M_c+1}\mathrm{S}_n(\mathbf{x}_i^{(n)}, \mathbf{x}_j^{(n)}) + \frac{1}{M_c+1}\sum_{k=1}^{M_c} S'_c(x_{i,k}^{(c)}, x_{j,k}^{(c)}) \end{aligned} \tag{11}$$

Let $\mathrm{S}_c(\mathbf{x}_i^{(c)}, \mathbf{x}_j^{(c)})$ denote the similarity between $\mathbf{x}_i^{(c)}$ and $\mathbf{x}_j^{(c)}$. If we assume that each categorical attribute has the same contribution to the calculation of similarity on categorical part, we have

$$\mathrm{S}_c(\mathbf{x}_i^{(c)}, \mathbf{x}_j^{(c)}) = \frac{1}{M_c}\sum_{k=1}^{M_c} S'_c(x_{i,k}^{(c)}, x_{j,k}^{(c)}) = \sum_{k=1}^{M_c}\frac{1}{M_c} S'_c(x_{i,k}^{(c)}, x_{j,k}^{(c)}) \tag{12}$$

Thus we can get

$$\begin{aligned} \mathrm{S}(\mathbf{x}_i, \mathbf{x}_j) &= \frac{1}{M_c+1}\mathrm{S}_n(\mathbf{x}_i^{(n)}, \mathbf{x}_j^{(n)}) + \frac{1}{M_c+1} M_c \sum_{k=1}^{M_c}\frac{1}{M_c} S'_c(x_{i,k}^{(c)}, x_{j,k}^{(c)}) \\ &= \frac{1}{M_c+1}\mathrm{S}_n(\mathbf{x}_i^{(n)}, \mathbf{x}_j^{(n)}) + \frac{M_c}{M_c+1}\mathrm{S}_c(\mathbf{x}_i^{(c)}, \mathbf{x}_j^{(c)}) \end{aligned} \tag{13}$$

#### 4.1.1. Similarity measure for numerical values

In this sub-section, we focus on the similarity metric on numerical values. For numerical values, we easily obtain the distance between $\mathbf{x}_i^{(n)}$ and $\mathbf{x}_j^{(n)}$. Thus, according to the measure of distance, we can define the similarity between numerical vectors [31,32]. $\mathrm{S}_n(\mathbf{x}_i^{(n)}, \mathbf{x}_j^{(n)})$ is related via Shepard's formulation as follows [33]:

$$\mathrm{S}_n(\mathbf{x}_i^{(n)}, \mathbf{x}_j^{(n)}) = \exp\left(-\mathrm{d}(\mathbf{x}_i^{(n)}, \mathbf{x}_j^{(n)})^2/2\right). \tag{14}$$

where $\mathrm{d}(\cdot, \cdot)$ stands for the Euclidean distance, i.e., Formula (1). Here, we note that the value of the similarity measure on numerical attributes cannot be 0. The closer the value of $S_n$ is to 1, the more similar the two objects are. If $\mathbf{x}_i^{(n)} = \mathbf{x}_j^{(n)}$, the value of it will be equal to 1.

#### 4.1.2. Similarity measure for categorical values

This sub-section studies the similarity for categorical part.

Firstly, we define the similarity between $x_{i,k}^{(c)}$ and $x_{j,k}^{(c)}$. Contrary to the simple matching [34], $S'_c(x_{i,k}^{(c)}, x_{j,k}^{(c)})$ is defined as

$$S'_c(x_{i,k}^{(c)}, x_{j,k}^{(c)}) = \begin{cases} 1, & if\ x_{i,k}^{(c)} = x_{j,k}^{(c)} \\ 0, & if\ x_{i,k}^{(c)} \ne x_{j,k}^{(c)} \end{cases}. \tag{15}$$

In previous discussion, we assume that the weight of each categorical attribute is the same. However, in practice, each categorical attribute has different contribution to the calculation of similarity on categorical part. One of the main reasons for this is that different attribute value has the different distribution. Thus Formula (12) can further be modified, as follows:

$$\mathrm{S}_c(\mathbf{x}_i^{(c)}, \mathbf{x}_j^{(c)}) = \sum_{k=1}^{M_c} w_k S'_c(x_{i,k}^{(c)}, x_{j,k}^{(c)}), \tag{16}$$

where $0 \le w_k \le 1$ and $\sum_{k=1}^{M_c} w_k = 1$. Obviously, $w_k$ is the weight of categorical attribute $A_k^{(c)}$. In other words, $w_k$ is the importance of categorical attribute $A_k^{(c)}$contributing to the calculation of the similarity on the categorical part.

Then we discuss how to calculate the weight $w_k$ of each categorical attribute $A_k^{(c)}$. We apply the notion of entropy to the calculation of the weights. The larger the inhomogeneity of the data set with respect to a categorical attribute, the larger the entropy of this categorical attribute is [35]. Besides, the inhomogeneity of the data set with respect to a categorical attribute corresponds to the importance of this categorical attribute. Therefore, according to Formula (2), we can calculate the entropy of a categorical attribute $A_k^{(c)}$ with $\mathrm{DOM}(A_k^{(c)}) = \{a_{k,1}, a_{k,2}, \cdots, a_{k,r_k}\}$ by

$$H_{A_k^{(c)}} = -\sum_{a_{k,t} \in DOM\left(A_k^{(c)}\right)} p(a_{k,t}) log\left(p(a_{k,t})\right), \tag{17}$$

where the probability $p(a_{k,t})$ of attribute value $a_{k,t}$ can be calculated by $\frac{\sum_{i=1}^{N} S'_c(x_{i,k}^{(c)}, a_{k,t})}{N}$. The function $S'_c(\cdot, \cdot)$ is similar to Formula (15). Obviously, the numerator denotes the number of objects whose value of the categorical attribute $A_k^{(c)}$ equals to $a_{k,t}$. And, $N$ is the total number of objects in the data set. Observing Formula (17) carefully, we notice the fact that if the number of values chosen by $A_k^{(c)}$, $r_k$, is very large, then the entropy of this categorical attribute, $H_{A_k^{(c)}}$, is also high. This is not the same as the actual case. In order to lower the impact of the categorical attributes with too many different values or even unique values, such as the ID number, we redefine the entropy of a categorical attribute $A_k^{(c)}$ as

$$H'_{A_k^{(c)}} = -\frac{1}{r_k}\sum_{t=1}^{r_k} p(a_{k,t}) log\left(p(a_{k,t})\right). \tag{18}$$

Hence, we can quantify the importance of a categorical attribute $A_k^{(c)}$ as

$$w_k = \frac{H'_{A_k^{(c)}}}{\sum_{k=1}^{M_c} H'_{A_k^{(c)}}}. \tag{19}$$

Substituting Formula (19) into Formula (16), we obtain the final similarity measure on the categorical attributes, as follows:

$$S_c\left(\mathbf{x}_i^{(c)}, \mathbf{x}_j^{(c)}\right) = \sum_{k=1}^{M_c} \left( \frac{H'_{A_k^{(c)}}}{\sum_{k=1}^{M_c} H'_{A_k^{(c)}}} \cdot S'_c\left(x_{i,k}^{(c)}, x_{j,k}^{(c)}\right) \right). \tag{20}$$

Notice that, similar to $S_n(\mathbf{x}_i^{(n)}, \mathbf{x}_j^{(n)})$, the value of $S_c(\mathbf{x}_i^{(c)}, \mathbf{x}_j^{(c)})$ also falls into the interval $[0, 1]$. The closer the value of $S_c$ is to 1, the more similar the two objects are. If $\mathbf{x}_i^{(c)} = \mathbf{x}_j^{(c)}$, the value of it will be equal to 1. And if $x_{i,k}^{(c)} \neq x_{j,k}^{(c)}$, for every $k$, $1 \le k \le M_c$, then the value of $S_c$ will be equal to 0.

#### 4.1.3. Similarity measure for mixed values

From the above content, it is easy to discover the proposed similarity measure for the numerical part (the whole numerical part) is expressed via an exponential function, whereas, for the categorical values, the similarity measure makes use of the entropy notion to compute the weight of each categorical attribute. Then, the two similarity measures are added together. Hence, this uniform similarity between two mixed type objects $\mathbf{x}_i$ and $\mathbf{x}_j$, denoted as $S(\mathbf{x}_i, \mathbf{x}_j)$, is defined by

$$\begin{aligned} S\left(\mathbf{x}_i, \mathbf{x}_j\right) = & \frac{1}{M_c + 1} \exp\left(-d\left(\mathbf{x}_i^{(n)}, \mathbf{x}_j^{(n)}\right)^2 / 2\right) \\ & + \frac{M_c}{M_c + 1} \sum_{k=1}^{M_c} \left( \frac{H'_{A_k^{(c)}}}{\sum_{k=1}^{M_c} H'_{A_k^{(c)}}} \cdot S'_c\left(x_{i,k}^{(c)}, x_{j,k}^{(c)}\right) \right), \end{aligned} \tag{21}$$

where the first term is the weighted similarity measure on the numerical attributes and the second term is the weighted similarity measure on the categorical attributes. Because the ranges of these two similarities $S_n(\mathbf{x}_i^{(n)}, \mathbf{x}_j^{(n)})$ and $S_c(\mathbf{x}_i^{(c)}, \mathbf{x}_j^{(c)})$ are the interval from 0 to 1, the value of $S(\mathbf{x}_i, \mathbf{x}_j)$ using the above weighting scheme also falls into the interval (0, 1]. Notice that the value of the similarity measure cannot reach to 0. If $\mathbf{x}_i = \mathbf{x}_j$, the value of it will be equal to 1.

To satisfy the requirement of the computation of the DP clustering algorithm, we convert the judged similarity $S(\cdot, \cdot)$ back into the distance $d_u(\cdot, \cdot)$. The smaller the distance is, the more similar the two objects are. Hence, the distance measure finally can be defined as

$$d_u\left(\mathbf{x}_i, \mathbf{x}_j\right) = -\log\left(S\left(\mathbf{x}_i, \mathbf{x}_j\right)\right). \tag{22}$$

### 4.2. Fuzzy neighborhood relation

The points $\mathbf{x}_1$ and $\mathbf{x}_2$ have the same number of neighbors within $\varepsilon \le d^{max}$ radius Fig. 2). There is an obvious difference between these points. The points $\mathbf{x}_1$ and $\mathbf{x}_2$ in Fig. 2 are the same according to the crisp neighborhood relation used in the DP clustering method. Because in classical case there is no difference with respect to membership degrees between points within the same neighborhood radius of core point (Fig. 3). This study expands the neighborhood set determined in Formula ((8) to the fuzzy neighborhood case. Utilizing fuzzy neighborhood function brings an advantage that there are different values of the neighborhood membership degrees of the points with respect to different distances from core point.

Note that, in order to form a fuzzy relation $\mu$: $X \times X \to [0, 1]$, the idea of the membership function of the fuzzy set defined in

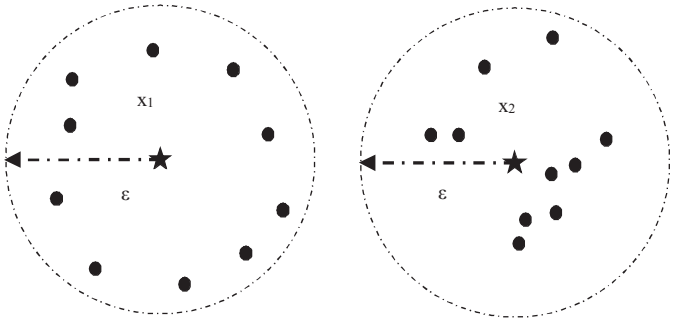



**Fig. 2.** Classical and fuzzy neighborhood relations.

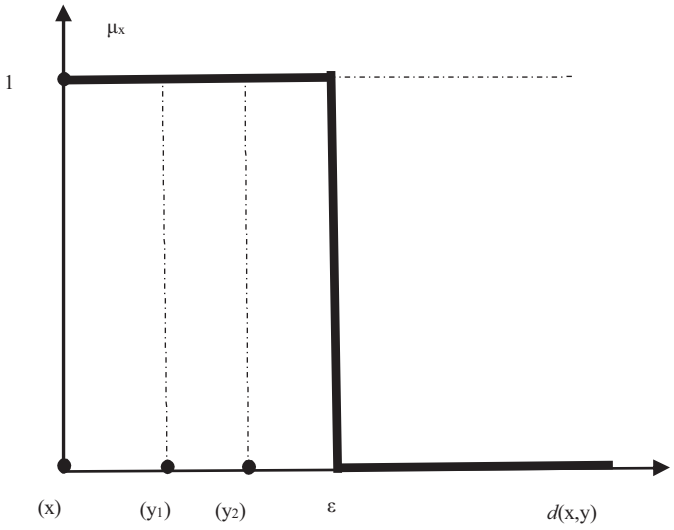



**Fig. 3.** The crisp neighborhood relation of the DP clustering.

Formula (8) is introduced. In order to be consistent with the parameter of the original DP clustering method, the radius of the considered fuzzy points is calculated as Formula (6) in the proposed algorithm.

Thus, such a neighborhood membership function is defined as

$$\mu'_{\mathbf{x}_i}\left(\mathbf{x}_j\right) = \begin{cases} 1 - \frac{d\left(\mathbf{x}_i, \mathbf{x}_j\right)}{\varepsilon}, & if\ d\left(\mathbf{x}_i, \mathbf{x}_j\right) < \varepsilon \\ 0, & otherwise \end{cases}. \tag{23}$$

In the previous example, if fuzzy neighborhood function is used, point $\mathbf{x}_1$ will have a higher membership degree of being a core point than that of point $\mathbf{x}_2$.

In neighborhood relation determined by Formula (8), neighborhood degrees of points with varying distances to the core point will be different from each other (Fig. 3).

As it is seen from Fig. 4, points $y_1$ and $y_2$ have different neighborhood membership degrees to the point x. Hence, the membership degree of $y_1$, i.e. $\alpha_1$, is higher than the membership degree of $y_2$, i.e. $\alpha_2$.

### 4.3. Algorithm description

Based on the definitions given above, a new local density is redefined for fuzzy logic approach, as follows:

$$\rho_i = \sum_j \mu'_{\mathbf{x}_i}\left(\mathbf{x}_j\right). \tag{24}$$

where $\mu'$ denotes the fuzzy neighborhood function defined by Formula (23).

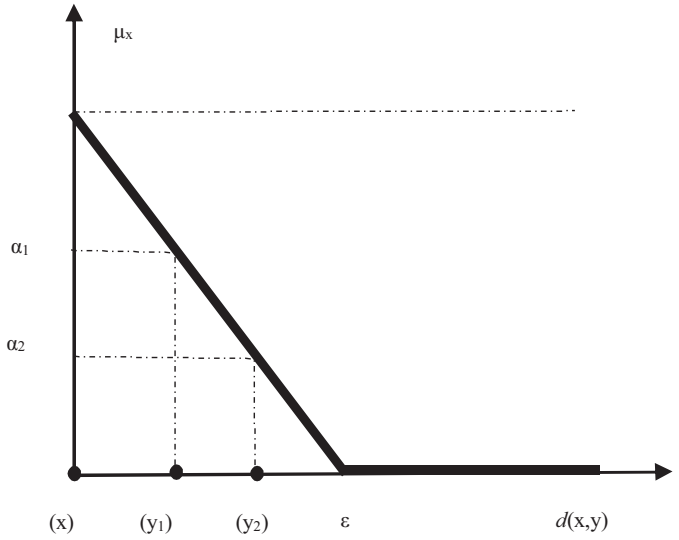



**Fig. 4.** The fuzzy neighborhood relation of the FN-DP clustering.

In order to select the cluster centers automatically, we introduce the $\gamma$-graph. Firstly, the local density $\rho_i$ and the separation distance $\delta_i$ are normalized to the scale of [0, 1]:

$$\overline{\rho_i} = \frac{\rho_i - \min\limits_i \rho_i}{\max\limits_i \rho_i - \min\limits_i \rho_i}. \tag{25}$$

$$\overline{\delta_i} = \frac{\delta_i - \min\limits_i \delta_i}{\max\limits_i \delta_i - \min\limits_i \delta_i}. \tag{26}$$

Secondly, the parameter $\gamma_i$ is defined as the product of these parameter $\overline{\rho_i}$ and $\overline{\delta_i}^2$.

$$\gamma_i = \overline{\rho_i} \cdot \overline{\delta_i}^2. \tag{27}$$

Once the parameter $\gamma_i$ has been calculated, we sort the data points in a descending order of $\gamma_i$. With very high probability the points with anomalously large $\gamma$ value are cluster centers. We need to determine cluster centers in a quantitative way. Therefore we define a threshold to determine cluster centers in terms of the parameter $\gamma_i$. The threshold $\alpha$ is defined by

$$\alpha = \mathrm{t} \cdot \mathrm{mean}(\gamma_i). \tag{28}$$

where $t$ is a new variable; mean($\gamma_i$) is the mean of all values of $\gamma_i$.

Therefore, cluster centers are the points that have $\gamma_i$ greater or equal to the threshold $\alpha$. As a rule of thumb, the parameter $\alpha$ can choose 5–40.

In this sub-section, we introduce the similarity metric presented in Section 4.1 into the original DP clustering algorithm in order to handle mixed type data. We can calculate the distance matrix of mixed type data by the proposed similarity measure. In addition, we use fuzzy neighborhood relation to redefine the local density. We also develop an automatic cluster center selection method.

The following algorithm is a summary of the proposed DP-MD-FN (Algorithm 1).

### 4.4. Performance analysis

The computational complexity is an important indicator of the algorithm. If the complexity is too high, it will limit the application of the algorithm in complex scenes [36]. Now, we give the time complexity of the DP-MD-FN algorithm. To be consistent with the above notations, we assume that $N$ is the number of objects in the data set; $M_n$ is the number of numerical attributes; $M_c$ is the number of categorical attributes; $r$ is the average number of different categorical attribute values. The cost of the similarity matrix is $O((M_nN)^2 + (rM_cN)^2)$. DP-MD-FN also needs $O(N^2)$ to compute the new local density. In addition, the cost of the sorting process with quick sort is $O(N\log N)$. As the complexity in assignment procedure is $O(N)$, the total time cost of our proposed algorithm is $O((M_nN)^2 + (rM_cN)^2) + O(N^2) + O(N\log N) + O(N) \sim O((r^2M_c^2 + M_n^2)N^2)$.

## 5. Experiments and results

In this section, we use experimental results to exhibit the robustness, the scalability and the clustering performance of our algorithm. In order to reveal the robustness and the scalability of the proposed algorithm, we design some synthetic data sets. To demonstrate the clustering performance of DP-MD-FN, we use it in some benchmark data sets obtained from the UCI repository. On the categorical data sets, we compare the proposed algorithm with clustering, K-Modes [14], KL-FCM-GM [15], EKP [17] and OCIL [8]. On the mixed type data sets, we compare the proposed algorithm with K-Prototypes [7], KL-FCM-GM, EKP and OCIL.

We conduct experiments in a work station with a core i7 DMI2-Intel 3.6 GHz processor and 18GB RAM running MATLAB 2012B.

In DP and DP-MD-FN, we select the parameter $d_c$ from {0.1% 0.5% 1% 2% 4% 6%}. K-Prototypes is an extension of K-Modes which can only handle categorical data. When we deal with categorical data sets, both the two clustering method results are consistent. Thus, we only run K-Modes on categorical data sets. The parameter $\gamma$ of the K-Prototypes varies from 0.1 to 2.1 in 0.1 increments. The parameter $\lambda$ of KL-FCM-GM varies from 0.1 to 2.1 in 0.1 increments. EKP contains 6 tunable parameters, they are: crossover probability, mutation probability, initial population, $\eta$ in SBX, n in polynomial mutation, the weight of categorical part, i.e., $\gamma$. The first five parameters are set to default according to Parameter Setting Table in [17]. $\gamma$ is an important tunable parameter and varies from 0.1 to 2.1 in 0.1 increments. As K-Modes, K-Prototypes, KL-FCM-GM, EKP and OCIL are stochastic, we run every algorithm 10 times on each data set and get the average.

### 5.1. Evaluation method

Before presenting the experimental results, we first discuss the evaluation of cluster quality. We use four well-known validity indexes (ACC, NMI, ARI and $F_1$). These indexes are widely used to measure clustering quality.

This paper uses clustering accuracy (ACC) [37] to measure the quality of clustering results. For $N$ distinct samples $\mathbf{x}_i \in \mathbb{R}^j$, $Y = (y_1, y_2, \cdots, y_k,)$ denotes the true category labels and $C = (c_1, c_2, \cdots, c_k)$ denotes the predicted cluster labels obtained by the clustering algorithm. The calculation formula of ACC is

$$\mathrm{ACC} = \sum_{i=1}^{N} \sigma(\mathrm{y_i}, map(\mathrm{c_i}))/N, \tag{29}$$

where $map(\cdot)$ maps each cluster label to a category label by the Hungarian algorithm [38] and this mapping is optimal, let $\sigma(\mathrm{y_i}, map(\mathrm{c_i}))$ equal to 1 if $\mathrm{y_i} = map(\mathrm{c_i})$ or 0 otherwise. In addition, $N$ is the number of objects in the data set. The higher the *ACC* value, the better the clustering performs.

In addition, we introduce the Normalized Mutual Information. The normalized mutual information between two variables $Y$ (category labels) and $C$ (cluster labels) is defined as

$$NMI(Y;C) = -\frac{I(Y;C)}{\sqrt{H(Y)H(C)}} \tag{30}$$

**Algorithm 1**
DP-MD-FN algorithm.

**Inputs:**
The samples $\boldsymbol{X} \in \mathbb{R}_{N\times M}$
The parameter $d_c, t$
**Outputs:**
The label vector of cluster index: $\boldsymbol{y} \in \mathbb{R}_{N\times\ 1}$
**Method:**
Step 1: Calculate distance matrix according to Formula (22)
Step 2: Calculate $\rho_i$ for point $\mathbf{x}_i$ according to Formula (24)
Step 3: Calculate $\delta_i$ for point $\mathbf{x}_i$ according to Formula (10)
Step 4: Select cluster centers according to Formula (28)
Step 5: Assign each remaining point to the cluster, which has its nearest neighbor of higher local density
Step 6: **Return y**

where $I(Y; C)$ is the mutual information between $Y$ and $C$. The entropies $H(Y)$ and $H(C)$ are used to normalize the mutual information in the range of [0, 1]. In practice, we made use of the following formulation to estimate the NMI score [39]:

$$NMI = -\frac{\sum_{i=1}^{K}\sum_{j=1}^{K} N_{i,j} log\left(\frac{N\cdot N_{i,j}}{N_i\cdot N_j}\right)}{\sqrt{\left(\sum_i N_i log\left(\frac{N_i}{N}\right)\right)\left(\sum_j N_j log\left(\frac{N_j}{N}\right)\right)}}. \tag{31}$$

where $N$ is the number of objections, $N_i$ and $N_j$ denote the number of objects in category $y_i$ and cluster $c_j$, respectively, and $N_{i,j}$ denotes the number of objects in category $y_i$ as well as in cluster $c_j$. The *NMI* score is 1 if the clustering results perfectly match the category labels, and the score is close to 0 if data is randomly partitioned. The higher the *NMI* score, the better the clustering quality.

The Rand index mainly evaluates the clustering results according to the relationship of pairwise data points. The adjusted Rand index (ARI) [40,41] is the corrected-for-chance version of the Rand index. $Y$ (category labels) and $C$ (cluster labels) are regard as two different partitions of the dataset. ARI uses data pairs to measure the agreement between two partitions. Let $a$ be the number of pairs of objects that are placed in the same class in $Y$ and in the same cluster in $C$, $b$ be the number of pairs of objects in the same class in $Y$ but not in the same cluster in $C$, $c$ be the number of pairs of objects in the same cluster in $C$ but not in the same class in $Y$, and $d$ be the number of pairs of objects in different classes and different clusters in both partitions. The quantities $a$ and $d$ can be interpreted as agreements, and $b$ and $c$ as disagreements. ARI is defined by:

$$ARI = \frac{2(ad-bc)}{(a+b)(b+d)+(a+c)(c+d)}, \tag{32}$$

where ARI $\in$ [0, 1], the higher the value of ARI, the better the clustering quality. Note that the adjusted Rand index can yield negative values if the index is less than the expected index.

F-Measure [42,43] is the ratio between recall and precision measurements. P($i$) denotes the precision rate:

$$\mathrm{P}(i) = \frac{N_i^{(Y,C)}}{N_i^{(C)}}, \tag{33}$$

where $N_i^{(Y,C)}$ is the size of the intersection of class $y_i$ and cluster $c_i$; $N_i^{(C)}$ is the size of cluster $c_i$.

R($i$) denotes the recall rate:

$$\mathrm{R}(i) = \frac{N_i^{(Y,C)}}{N_i^{(Y)}}, \tag{34}$$

where $N_i^{(Y,C)}$ is the size of the intersection of class $y_i$ and cluster $c_i$; $N_i^{(Y)}$ is the size of class $y_i$.

$\mathrm{F}_1$ measure is the harmonic mean of precision and recall. The $\mathrm{F}_1$-score of class $y_i$ is defined by:

$$\mathrm{F}_1(i) = 2\cdot\frac{1}{\frac{1}{\mathrm{R}(i)}+\frac{1}{\mathrm{P}(i)}} = 2\cdot\frac{\mathrm{R}(i)\cdot\mathrm{P}(i)}{\mathrm{R}(i)+\mathrm{P}(i)} \tag{35}$$

The total $\mathrm{F}_1$-score of the clustering results is the weighted average of each class's $\mathrm{F}_1$-score:

$$\mathrm{F}_1 = \frac{1}{N}\sum_{i=1}^{k} N_i^{(Y)}\cdot\mathrm{F}_1(i) \tag{36}$$

where $N$ is the number of objects; $k$ is the class number of data set; $N_i^{(Y)}$ is the size of class $y_i$. $\mathrm{F}_1$-score reaches its best value at 1 and worst at 0.

In order to find significant differences among the results obtained by the clustering algorithms, statistical analysis is used. The Friedman test [44–46] is a non-parametric statistical test that determines whether there are significant differences in the results of the clustering algorithms. The first step in calculating the test statistic is to convert the original results to ranks. It ranks the algorithms for each data set separately. For the $i$th of $\mathrm{n}_{ds}$ data sets, rank values from 1 (best result) to $\mathrm{k}_a$ (worst result). Denote these ranks as $r_j^i (1 \le j \le k_a,\ 1 \le i \le n_{ds})$. Then the Friedman test computes the average ranks of algorithms, $R_j = \frac{1}{n_{ds}}\sum_i r_j^i$. Under the null-hypothesis, which states that all the algorithms are equivalent and so their ranks $R_j$ should be equal, the Friedman statistic

$$\chi_F^2 = \frac{12 n_{ds}}{k_a(k_a+1)}\left[\sum_j {R_j}^2 - \frac{k_a(k_a+1)^2}{4}\right] \tag{37}$$

is distributed according to $\chi_F^2$ with $k_a - 1$ degrees of freedom.

Iman and Davenport showed that Friedman's $\chi_F^2$ is undesirably conservative and derived a better statistic

$$F_F = \frac{(n_{ds}-1)\chi_F^2}{n_{ds}(k_a-1)-\chi_F^2} \tag{38}$$

which is distributed according to the F-distribution with $k_a - 1$ and $(k_a - 1)(n_{ds} - 1)$ degrees of freedom. We use $\alpha = 0.05$ as the level of confidence in all cases. A wider description of these tests is presented in [47,48].

### 5.2. Experiments on synthetic data sets

#### 5.2.1. Results for synthetic data sets

The first base data set, BaseOne data set, has 360 data points, equally distributed into three classes: $G_1$, $G_2$, $G_3$. Each data point consists of three categorical attributes. The categorical attribute values are assigned to each class in equal proportion. Categorical attribute 1 has a unique categorical value for each class. Categorical attribute 2 has two distinct categorical values assigned to each class. And categorical attribute 3 has three distinct categorical values assigned to each class. Based on the BaseOne data set,

**Table 1**
Clustering accuracy of the evaluated algorithms on BaseOne and its variations.

| Data set | | BaseOne | IrrOne1 | IrrOne2 | CorOne1 | CorOne2 | CorOne3 |
|---|---|---|---|---|---|---|---|
| DP-MD-FN | ACC | **1.0** | **1.0** | **1.0** | **1.0** | **1.0** | **1.0** |
| | Para | $d_c = 6\%$, t = 30 | $d_c = 4\%$, t = 30 | $d_c = 4\%$, t = 30 | $d_c = 4\%$, t = 30 | $d_c = 4\%$, t = 30 | $d_c = 4\%$, t = 30 |
| K-Modes (K-Prototypes) | ACC | 0.8422 ± 0.2077 | 0.7661 ± 0.2006 | 0.7558 ± 0.1780 | 0.7503 ± 0.2149 | 0.7058 ± 0.1613 | 0.6675 ± 0.1593 |
| | Para | K = 3 | K = 3 | K = 3 | k = 3 | K = 3 | k = 3 |
| KL-FCM-GM | ACC | **1.0 ± 0.0** | **1.0 ± 0.0** | **1.0 ± 0.0** | **1.0 ± 0.0** | **1.0 ± 0.0** | **1.0 ± 0.0** |
| | Para | $\lambda = 0.7$ | $\lambda = 1.6$ | $\lambda = 1.6$ | $\lambda = 1.6$ | $\lambda = 1.6$ | $\lambda = 1.0$ |
| EKP | ACC | 1.0 ± 0.0 | 0.9653 ± 0.0087 | 0.9217 ± 0.0087 | 0.9992 ± 0.0026 | 0.9950 ± 0.0070 | 0.9950 ± 0.0049 |
| | Para | k = 3 | k = 3 | k = 3 | k = 3 | k = 3 | k = 3 |
| OCIL | ACC | 0.9017 ± 0.1979 | 0.9078 ± 0.1917 | 0.9500 ± 0.1508 | 0.8522 ± 0.2271 | 0.8233 ± 0.2371 | 0.7917 ± 0.2462 |
| | Para | k = 3 | k = 3 | k = 3 | k = 3 | k = 3 | k = 3 |

**Table 2**
Clustering accuracy of the evaluated algorithms on BaseTwo and its variations.

| Data set | | BaseTwo | IrrTwo1 | IrrTwo2 | CorTwo1 | CorTwo2 | CorTwo3 |
|---|---|---|---|---|---|---|---|
| DP-MD-FN | ACC | **1.0** | **1.0** | **1.0** | **1.0** | **1.0** | **1.0** |
| | Para | $d_c = 6\%$, t = 30 | $d_c = 4\%$, t = 30 | $d_c = 4\%$, t = 30 | $d_c = 4\%$, t = 30 | $d_c = 4\%$, t = 30 | $d_c = 4\%$, t = 30 |
| K-Prototypes | ACC | **1.0 ± 0.0** | **1.0 ± 0.0** | 0.9981 ± 0.0019 | 0.9889 ± 0.0719 | 0.9889 ± 0.0719 | 0.9556 ± 0.1405 |
| | Para | $\gamma = 2.0$ | $\gamma = 0.5$ | $\gamma = 0.9$ | $\gamma = 0.5$ | $\gamma = 0.5$ | $\gamma = 0.5$ |
| KL-FCM-GM | ACC | **1.0 ± 0.0** | **1.0 ± 0.0** | 0.3544 ± 0.030 | **1.0 ± 0.0** | 0.8200 ± 0.0029 | 0.7667 ± 0.0 |
| | Para | $\lambda = 0.7$ | $\lambda = 1.6$ | $\lambda = 2.1$ | $\lambda = 0.6$ | $\lambda = 2.1$ | $\lambda = 0.5$ |
| EKP | ACC | 0.8916 ± 0.1583 | 0.8961 ± 0.1583 | 0.8961 ± 0.1583 | 0.7869 ± 0.0676 | 0.6453 ± 0.0413 | 0.5417 ± 0.0 |
| | Para | $\gamma = 1.7$ | $\gamma = 1.8$ | $\gamma = 1.2$ | $\gamma = 1.3$ | $\gamma = 0.8$ | $\gamma = 1.2$ |
| OCIL | ACC | 0.9111 ± 0.1787 | 0.9868 ± 0.0648 | 0.9909 ± 0.0524 | 0.9876 ± 0.0715 | 0.9071 ± 0.1819 | 0.9084 ± 0.1420 |
| | Para | k = 3 | k = 3 | k = 3 | k = 3 | k = 3 | k = 3 |

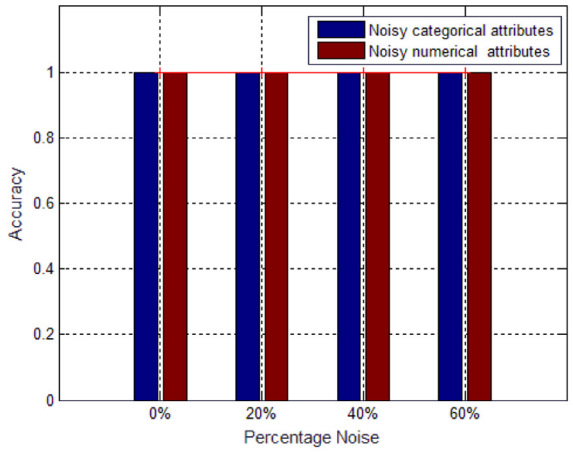
IrrOne1 adds a categorical attribute which is an irrelevant attribute and only has a random relationship with the predicted class. Based on IrrOne1 data set, IrrOne2 adds another irrelevant categorical attribute. In addition to IrrOne1, we generate three data sets which have some corrupted categorical attributes. CorOne1, CorOne2 and CorOne3 are generated based on the BaseOne data set by successively mixing up 20%, 40%, and 60% of the third attribute values for one class with the other two classes.

The second base data set, BaseTwo data set, has 360 data points, equally distributed into three classes: $G_1$, $G_2$, $G_3$. Each data point consists of two categorical attributes and two numerical attributes. The categorical attribute values are assigned to each class in equal proportion. Categorical attribute 1 has a unique categorical value for each class. And categorical attribute 2 has two distinct categorical values assigned to each class. The numerical attribute values are generated by sampling normal distributions with different means and standard deviations for each class. For the first class, the two numerical attributes are distributed as $N(\mu = 3, \sigma = 1)$ and $N(\mu = 5, \sigma = 3)$; for the second class, the distributions are $N(\mu = 9, \sigma = 1)$ and $N(\mu = 9, \sigma = 3)$; for the third class, $N(\mu = 15, \sigma = 1)$ and $N(\mu = 13, \sigma = 3)$. It means that the first numerical attribute can make the three classes apart from each other, and the first numerical attribute cannot. Based on BaseTwo data set, IrrTwo1 adds a numerical attribute which is an irrelevant attribute and only has a random relationship with the predicted class. Based on IrrTwo1 data set, IrrTwo2 adds another irrelevant numerical attribute. In addition to IrrTwo2, we generate three other data sets which have some corrupted numerical attributes. CorTwo1, CorTwo2 and CorTwo3 are generated based on the BaseTwo data set by successively mixing up 20%, 40%, and 60% of the second numerical attribute values for one class with the other two classes.

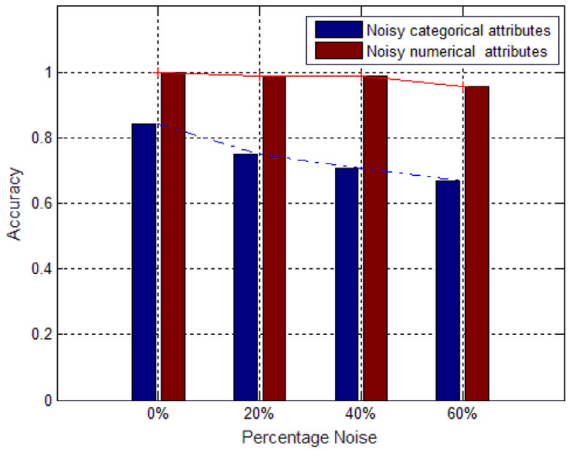
The partitional structures generated by the five clustering methods are evaluated using clustering accuracy measure. Table 1 shows the results of the evaluated algorithms on the BaseOne set and its variations. Table 1 shows the results of the evaluated algorithms over different synthetic data sets only containing categorical attributes. On the BaseTwo set and its variations, the ACC values generated by these methods are listed in Table 2. For every data set, the ACC values of the best performance method are in bold, which makes them really stand out. Comparisons with other methods illustrate the superior performance of the proposed approach.

In order to make results clear, the effects of the corrupted attribute are plotted in Fig. 5 and the effects of the irrelevant attributes are plotted in Fig. 6. It is obvious that results obtained by DP-MD-FN are not affected by the corrupted attributes. As seen in Fig. 5(a), the proposed method generates the optimal structure of clusters on these eight synthetic data sets. We conjecture that although the third attribute or the second numerical attribute is corrupted, the attribute has little contribution to the similarity value and the remaining attributes contributions remain high. Consequently, these noisy attributes have no effect on the partitional structures generated by our method. As seen in Fig. 5(b), as the percentage of the noise in the numerical attribute increases, there is little deterioration. But when the categorical attribute is corrupted, the ACC values generated by K-Modes (or K-Prototypes) decrease significantly. As seen in Fig. 5(c), KL-FCM-GM shows the opposite effect. Like DP-MD-FN, it is one of the most robust methods for noisy in the categorical attribute. But KL-FCM-GM is more sensitive to the corrupted numerical attribute. Analogous to KL-FCM-GM, EKP shows very little deterioration in the categorical attribute. But as the percentage of the noise in the numerical attribute increases, its performance deteriorates quite rapidly. Unlike other methods, OCIL does not perform very well on the two base data sets. It is interesting to observe that as the percentage of the noise in the numerical attribute increases, its performance improves. We conjecture that a certain level (about 20%) of the noise is introduced into the numerical attribute, the contribution of the numerical attributes to the similarity value is depleted. However, when we continue to increase the noise, the ACC values generated by OCIL descend. Of the five algorithms, the proposed method is the most robust and generates the optimal structure of clusters on data sets with noise.

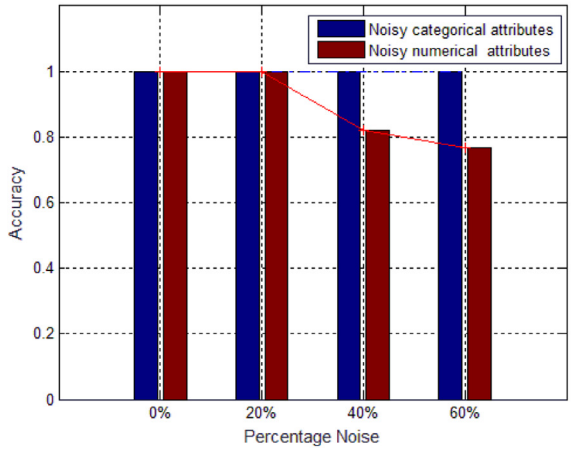
As seen in Fig. 6(a), the proposed method generates the optimal structure of clusters on these six synthetic data sets. It is obvious that results obtained by DP-MD-FN are not affected by these additional irrelevant attributes. As seen in Fig. 6(b), when the number of the irrelevant categorical attributes increases, the ACC val-



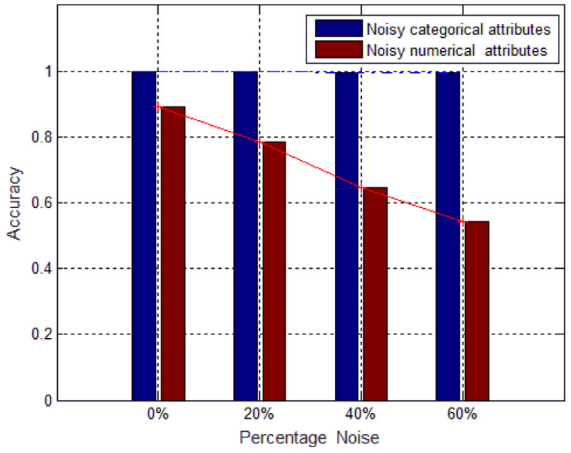

(a) DP-MD-FN



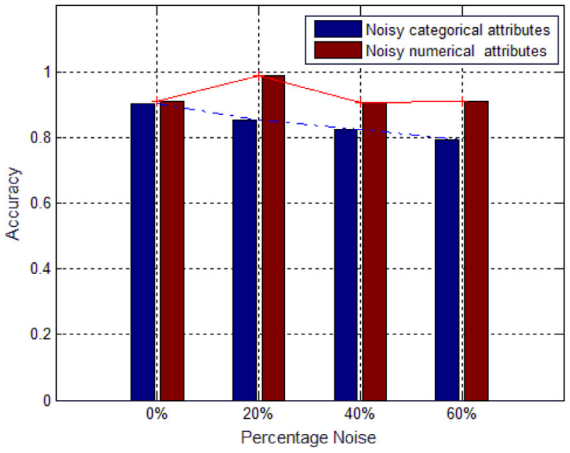

(b) K-Modes(K-Prototypes)




(c) KL-FCM-GM

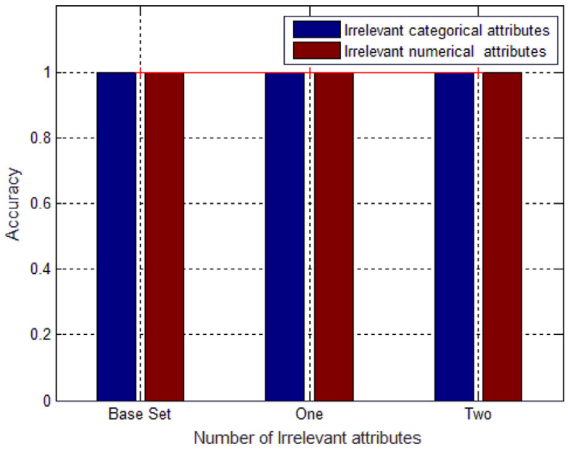



(d) EKP

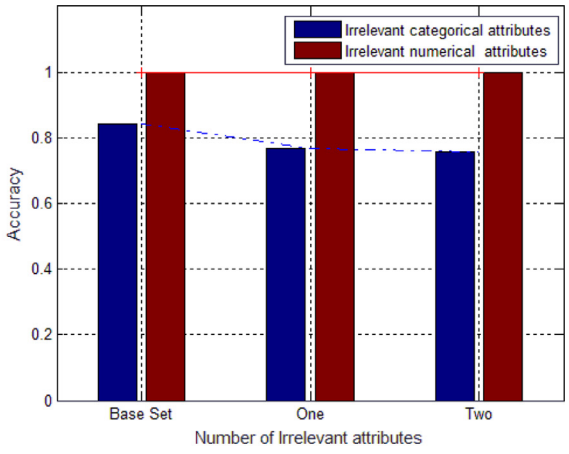



(e) OCIL

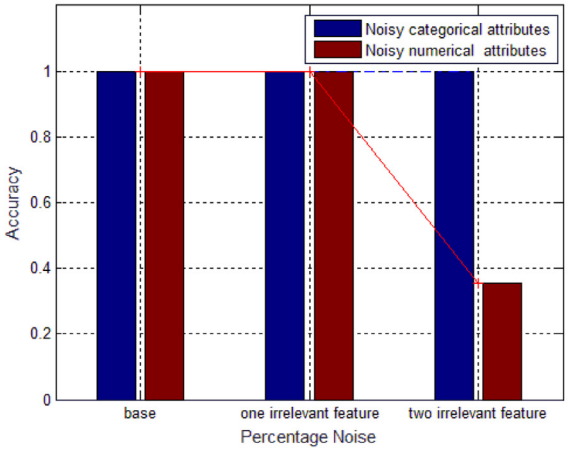
**Fig. 5.** The effects of the corrupted attributes.



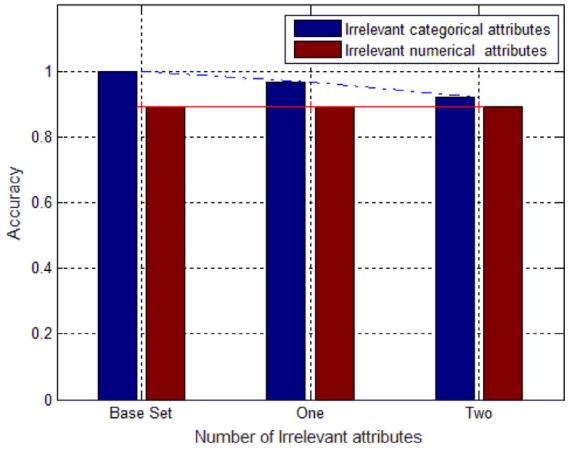

(a) DP-MD-FN



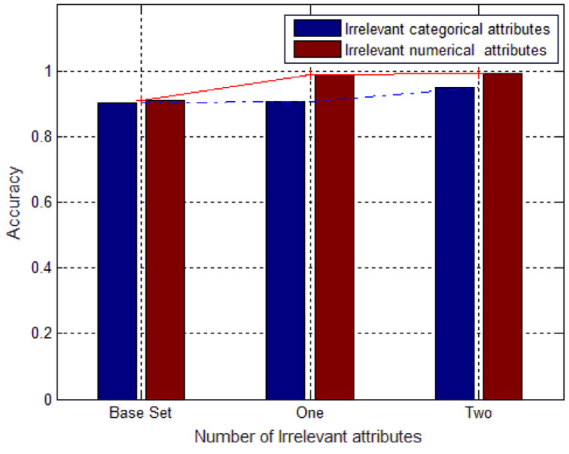

(b) K-Modes(K-Prototypes)




(c) KL-FCM-GM




(d) EKP




(e) OCIL

**Fig. 6.** The effects of the irrelevant attributes.

ues generated by K-Modes (or K-Prototypes) decrease significantly. However, as the number of the irrelevant numerical attributes increases, there is little deterioration. As seen in Fig. 6(c), KL-FCM-GM shows the opposite effect. Like DP-MD-FN, it generates the optimal structure of clusters on these data sets containing the irrelevant categorical attributes. But KL-FCM-GM is more sensitive to the irrelevant numerical attributes. Especially, it does an extremely poor job in clustering IrrTwo2 data set. Analogous to K-Modes, EKP shows deterioration when the number of the irrelevant categorical attributes increases. It generates the optimal structure of clusters on the BaseTwo data set. However, the irrelevant numerical attributes have no effect on EKP's results, as seen in Fig. 6(d). It is interesting to observe that as the number of the irrelevant attributes increases, its performance improves. We conjecture that as the number of the irrelevant attributes increases, the contribution of the relevant attributes to the similarity value rises. As a result, the ACC values generated by OCIL ascend. Of the five algorithms, the proposed method is the most robust and generates the optimal structure of clusters on data sets containing the irrelevant attributes.

Figs. 7 and 8 display the decision graph and $\gamma$-graph results of the proposed algorithm for clustering the IrrOne1 data set and the CorTwo3 data set in all $d_c$ parameter values, respectively. The $\gamma$-graph results generated by our algorithm have great performance. The $\gamma$ values of centers are far greater than those of the other points. The proposed automatic cluster center selection method can find the right number of clusters based on the $\gamma$-graph results. We can see in the $\gamma$-graph 2 or 3 points (Figs. 7 or 8) standing out the others. The selected cluster centers are represented in different color (yellow, blue and green). The corresponding decision graphs obtained by propagation of the label are also showed in Figs. 7 and 8. Note that the $\gamma$-graphs in Figs. 7 and 8 show the sorted gamma values. From the decision graphs in Figs. 7 and 8, it is easy to see that the selected cluster centers have anomalously large $\delta$ and relatively large $\rho$, as described in original paper [9]. Furthermore, it is worth noting that the $\gamma$ values of the selected cluster centers are far larger than those of the other points. Therefore, it is insensitive to the choice of the parameter $t$.

Fig. 9 shows the ACC values obtained by the proposed algorithm in the above cases. It is obvious that the proposed algorithm generates the optimal structure of clusters regardless of the value of the $d_c$ parameter $d_c$. In other words, the proposed method is robust with respect to choosing $d_c$.

Generally, of the five methods, DP-MD-FN is the most robust and does an excellent job when the degradation levels or the number of the irrelevant attributes increases. Also, it is insensitive to the choice of $d_c$.

#### 5.2.2. Scalability tests

In this sub-section, we test the scalability of the DP-MD-FN algorithm on some synthetic data sets. In these experiments, all synthetic data sets are generated by a synthetic data generator (http://www.datasetgenerator.com). From the earlier discussions, we can know that the proposed algorithm is the generalization of the DP clustering algorithm. It means that our algorithm and DP have the same scalability for handling numerical values. In order to make the research target-oriented, we generate these data sets containing only categorical attributes. The scalability tests of the proposed algorithm can fall into four categories. The first category tests the time varies with change of the number of objects when the numbers of attributes, attribute values and clusters are kept fixed. The second one is the scalability against the number of attributes for given numbers of objects, attribute values and clusters. The third one is the scalability against the number of attribute values for given number of objects and attributes and clusters. The last category of the tests is that the numbers of objects, attributes and attribute values are kept fixed and the number of clusters is varied.

**Table 3**
The details of the categorical data sets.

| Data Sets | Cluster | Dimension | N |
|---|---|---|---|
| LED Display Domain | 10 | 7 | 500 |
| Tic-Tac-Toe Game | 2 | 9 | 958 |
| Congressional Voting | 2 | 16 | 232 |
| Mushroom | 2 | 22 | 5644 |
| Soybean | 4 | 35 | 47 |

Fig. 10 shows the execution time of the proposed algorithm to cluster objects ($N = 1000,\ 2000,\ 3000,\ 4000,\ 5000$) with 5 categorical attributes, each of which contains 5 attribute values, into 5 clusters. We run the proposed algorithm 10 times on each data set and get the average. In Fig. 10, the y-axis shows the execution time of DP-MD-FN in seconds, and the x-axis shows the number of objects. To be convenient, each attribute includes the same number of values on each data sets. This is applied to the following cases. One important observation from Fig. 10 is that DP-MD-FN's running time is polynomial. The main reason for this that is the original method need to compute the pairwise distance between pairs of objects. The cost of this calculation itself is $O(N^2)$.

Fig. 11 shows the execution time of the proposed algorithm to cluster 1000 objects with categorical attributes ($M = 5,\ 10,\ 15,\ 20,\ 25$), each of which contains 5 attribute values, into 5 clusters. As described above, we run the proposed algorithm 10 times on each data set and get the average. In Fig. 11, the y-axis shows the execution time of DP-MD-FN in seconds, and the x-axis shows the number of attributes. Our algorithm scales linearly with the number of attributes, as shown in Fig. 11. This is distinct from the previous analysis in Section 4.4. The reason for this result may be that MATLAB has a very strong processing capability for matrix manipulations.

Fig. 12 shows the execution time of the proposed algorithm to cluster 1000 objects with 5 categorical attributes, each of which contains attribute values ($r = 5,\ 10,\ 15,\ 20,\ 25$), into 5 clusters. As previously described, we run the proposed algorithm 10 times on each data set and get the average. In Fig. 12, the y-axis shows the execution time of DP-MD-FN in seconds, and the x-axis shows the number of attribute values. Fig. 12 shows that the curve of the execution time using our algorithm is almost flat. It means that the running speed of DP-MD-FN is not dependent on the number of attribute values.

Fig. 13 shows the execution time of the proposed algorithm to cluster 1000 objects with 5 categorical attributes, each of which contains 5 attribute values, into clusters ($k = 5,\ 10,\ 15,\ 20,\ 25$). Like all previous tests, we run the proposed algorithm 10 times on each data set and get the average. In Fig. 13, the y-axis shows the execution time of DP-MD-FN in seconds, and the x-axis shows the number of clusters. Fig. 13 shows that the curve of the execution time using our algorithm is almost flat. It means that the running speed of DP-MD-FN is independent of the number of clusters.

### 5.3. Experiments on real-world data sets

The Categorical data sets used in the experiment are taken from the UCI Machine Learning Repository, including LED Display Domain, Tic-Tac-Toe Game, Congressional Voting Records, Mushroom and Soybean. The details of these data sets are listed in Table 3. It is important to note that the LED Display Domain data set is a sample of 500 objects obtained from the original data generator. Thus, the LED Display Domain data set consists of 500 objects with 7 categorical attributes. The data objects can be divided into ten classes. Also, there are a few missing values in the Congressional

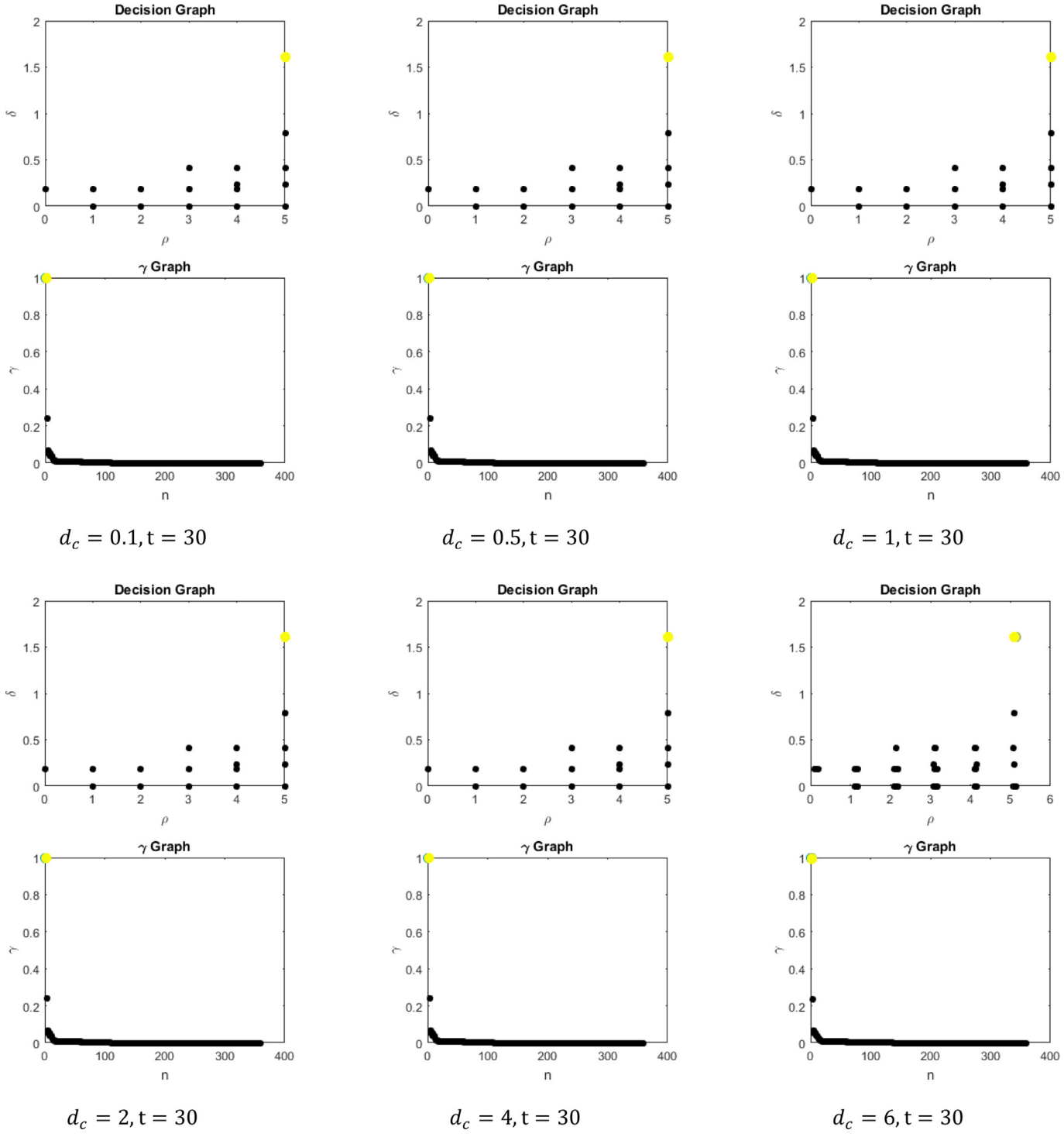



**Fig. 7.** The decision graphs of DP-MD-FN on the IrrOne1 data set.

Voting Records data set. A complete version of this data set has 435 objects. To facilitate handling this data set, we use a cleaned version (where objects with missing values are not included) consisting of 232 objects. The Congressional Voting Records data set consists of 232 samples with 16 categorical attributes. Similar to the Congressional Voting Records data set, there are a few missing values in the Mushroom data set. A complete version of this data set has 8124 objects. We use a cleaned version consisting of 5644 objects.

The mixed type data sets used in the experiment also are all taken from the UCI Machine Learning Repository, including South African Hearth, Heart, Australian Credit Approval, Credit Approval, Bank Marketing and KDD Cup 1999. The details of these data sets are listed in Table 4. Similar to the Congressional Voting Records data set, there are a few missing values in the Credit Approval data set. A complete version of this data set has 690. To facilitate handling this data set, we use a cleaned version with 653 objects. Thus, the Credit Approval data set consists of 653 samples with six numerical and nine categorical attributes. Also a complete version

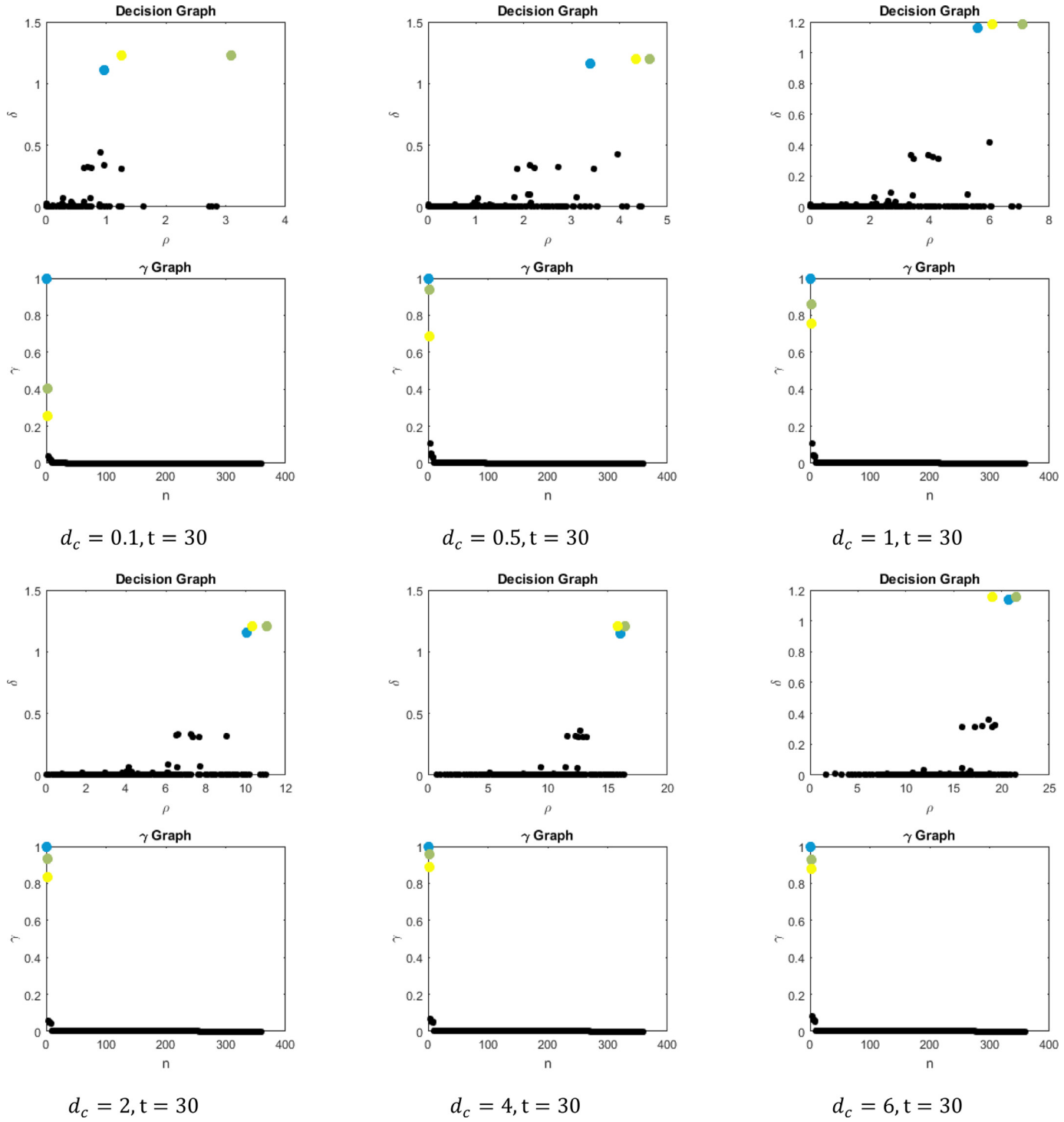



**Fig. 8.** The decision graphs of DP-MD-FN on the CorTwo3 data set.

of the KDD Cup 1999 data set has 4,000,000 objects. Our device cannot process such a large data set due to the limitations of the memory. Since we use a subset of the KDD Cup 1999 data set has 2000 objects, equally distributed into four classes. However, there is no difference in the attributes and each object has 26 numerical and 15 categorical attributes.

*5.3.1. Experiments on categorical data sets*

In Table 5, we list the clustering accuracy of our proposed algorithm, K-Modes (K-Prototypes), KL-FCM-GM, EKP and OCIL on categorical data sets. As we can see, DP-MD-FN results are better than those obtained by other methods for these four data sets. Also, the proposed method finds the optimal structure of clusters in clustering the Soybean data set, while others do not. The main reason for this is that these comparison partners are sensitive to initialization and is more suitable for spherical distribution data. Relatively, based on the DP clustering algorithm, DP-MD-FN can deal with non- spherical distribution data.

In Table 6, we list the five evaluated algorithms generating the NMI values on categorical data sets. Our method shows a small advantage compared with others on LED Display Domain and Tic-Tac-

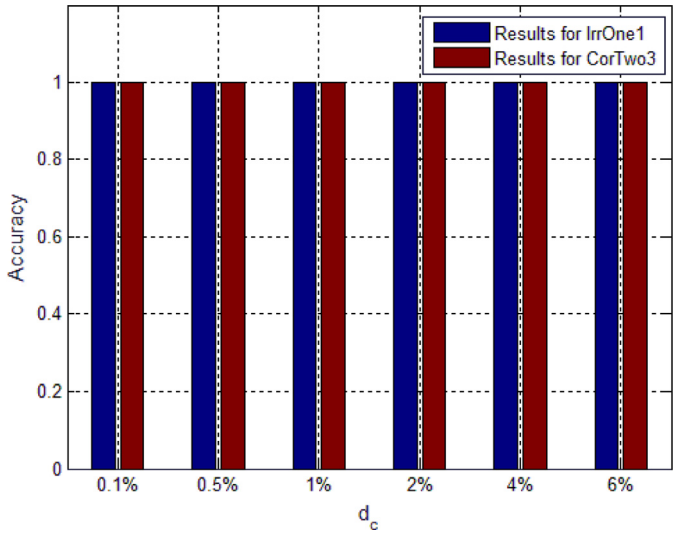



**Fig. 9.** The results of DP-MD-FN on IrrOne1 and CorTwo3 data sets.

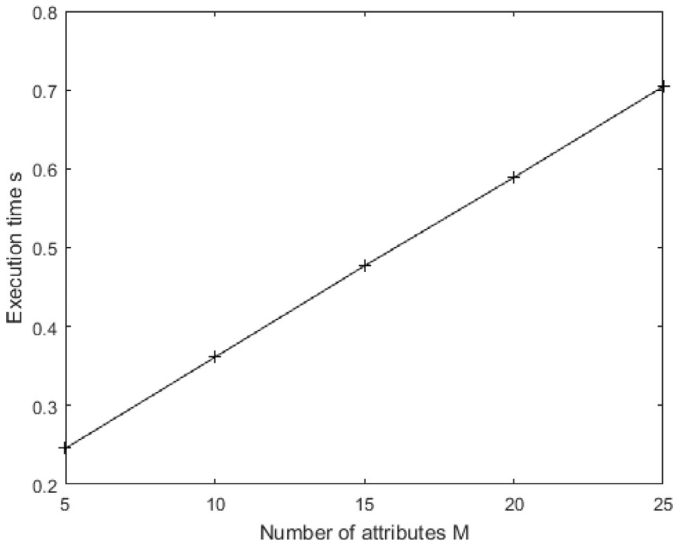



**Fig. 11.** The execution time vs. the number of attributes.

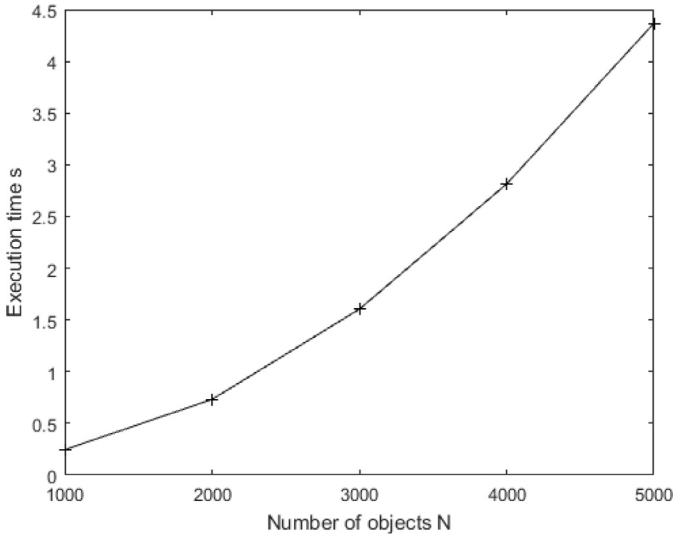



**Fig. 10.** The execution time vs. the number of objects.

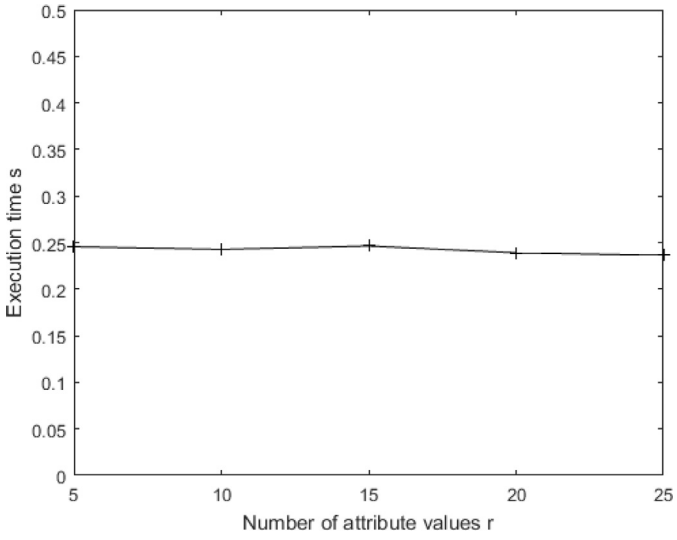



**Fig. 12.** The execution time vs. the number of attribute values.

**Table 4**
The details of the mixed type data sets.

| Data Sets | Cluster | Dimension ($M_n + M_c$) | N |
|---|---|---|---|
| South African Hearth | 2 | 8 + 1 | 462 |
| Heart | 2 | 6 + 7 | 270 |
| Australian Credit Approval | 2 | 6 + 8 | 690 |
| Credit Approval | 2 | 6 + 9 | 653 |
| Bank Marketing | 2 | 7 + 9 | 4521 |
| KDD Cup 1999 | 4 | 26 + 15 | 2000 |

Toe Game data sets. Beyond that, the NMI values obtained by DP-MD-FN are significantly superior to those obtained by other methods.

In Table 7, we list the five evaluated algorithms generating the ARI values on categorical data sets. DP-MD-FN yields the best results among the five algorithms on three out of the five categorical data sets. Although our method slightly inferiors to OCIL (the best performance) on the LED Display Domain and Congressional Vot-

**Table 5**
The ACC values of the evaluated algorithms on categorical data sets.

| Data set | | LED Display Domain | Tic-Tac-Toe Game | Congressional Voting | Mushroom | Soybean |
|---|---|---|---|---|---|---|
| DP-MD-FN | ACC | **0.6860** | **0.6628** | **0.9181** | **0.8540** | **1.0** |
| | Para | $d_c = 4\%,\ t = 20$ | $d_c = 1\%,\ t = 40$ | $d_c = 4\%,\ t = 20$ | $d_c = 2\%,\ t = 20$ | $d_c = 6\%,\ t = 5$ |
| K-Modes | ACC | 0.5382 ± 0.0583 | 0.5649 ± 0.0310 | 0.8694 ± 0.0054 | 0.8161 ± 0.1090 | 0.8404 ± 0.1522 |
| | Para | k = 10 | k = 2 | k = 2 | k = 2 | k = 4 |
| KL-FCM-GM | ACC | 0.5390 ± 0.0452 | 0.6534 ± 0 | 0.8879 ± 0 | 0.8436 ± 0.0186 | 0.9617 ± 0.0814 |
| | Para | $k = 10,\ \lambda = 0.8$ | $k = 2,\ \lambda = 1.5$ | $k = 2,\ \lambda = 1.5$ | $k = 2,\ \lambda = 1.8$ | $k = 4,\ \lambda = 1.8$ |
| EKP | ACC | 0.5768 ± 0.0410 | 0.5533 ± 0.0283 | 0.8664 ± 0 | 0.8522 ± 0 | 0.9723 ± 0.0144 |
| | Para | k = 10 | k = 2 | k = 2 | k = 2 | k = 4 |
| OCIL | ACC | 0.5872 ± 0.0737 | 0.5585 ± 0.0290 | 0.8931 ± 0.0018 | 0.5957 ± 0.0880 | 0.9362 ± 0.1346 |
| | Para | k = 10 | k = 2 | k = 2 | k = 2 | k = 4 |

**Table 6**
The NMI values of the evaluated algorithms on categorical data sets.

| Data set | | LED Display Domain | Tic-Tac-Toe Game | Congressional Voting | Mushroom | Soybean |
|---|---|---|---|---|---|---|
| DP-MD-FN | NMI | **0.5303** | **0.0183** | **0.5967** | **0.4396** | **1.0** |
| | Para | $d_c = 4\%,\ t = 20$ | $d_c = 1\%,\ t = 40$ | $d_c = 4\%,\ t = 20$ | $d_c = 2\%,\ t = 20$ | $d_c = 6\%,\ t = 5$ |
| K-Modes | NMI | 0.4952 ± 0.0444 | 0.0125 ± 0.0206 | 0.4587 ± 0.0264 | 0.3606 ± 0.1093 | 0.8441 ± 0.1310 |
| | Para | k = 10 | k = 2 | k = 2 | k = 2 | k = 4 |
| KL-FCM-GM | NMI | 0.5135 ± 0.0224 | 0.0017 ± 0 | 0.5111 ± 0 | 0.3992 ± 0.0615 | 0.9 ± 0.1009 |
| | Para | $k = 10,\ \lambda = 0.8$ | $k = 2,\ \lambda = 1.5$ | $k = 2,\ \lambda = 1.5$ | $k = 2,\ \lambda = 1.8$ | $k = 4,\ \lambda = 1.8$ |
| EKP | NMI | 0.5201 ± 0.0128 | 0.0094 ± 0.0081 | 0.4462 ± 0 | 0.4187 ± 0 | 0.9322 ± 0.0299 |
| | Para | k = 10 | k = 2 | k = 2 | k = 2 | k = 4 |
| OCIL | NMI | 0.5223 ± 0.0378 | 0.0072 ± 0.0020 | 0.5344 ± 0.0045 | 0.0598 ± 0.1150 | 0.9441 ± 0.1178 |
| | Para | k = 10 | k = 2 | k = 2 | k = 2 | k = 4 |

**Table 7**
The ARI values of the evaluated algorithms on categorical data sets.

| Data set | | LED Display Domain | Tic-Tac-Toe Game | Congressional Voting | Mushroom | Soybean |
|---|---|---|---|---|---|---|
| DP-MD-FN | ARI | 0.4248 | **0.0552** | 0.5737 | **0.4933** | **1.0** |
| | Para | $d_c = 4\%,\ t = 20$ | $d_c = 1\%,\ t = 40$ | $d_c = 4\%,\ t = 20$ | $d_c = 2\%,\ t = 20$ | $d_c = 6\%,\ t = 5$ |
| K-Modes | ARI | 0.3699 ± 0.0641 | 0.0165 ± 0.0188 | 0.5440 ± 0.0162 | 0.4297 ± 0.1403 | 0.7680 ± 0.2005 |
| | Para | k = 10 | k = 2 | k = 2 | k = 2 | k = 4 |
| KL-FCM-GM | ARI | 0.3928 ± 0.0256 | 0 ± 0 | 0.5869 ± 0 | 0.4656 ± 0.0472 | 0.9532 ± 0.0988 |
| | Para | $k = 10,\ \lambda = 0.8$ | $k = 2,\ \lambda = 1.5$ | $k = 2,\ \lambda = 1.5$ | $k = 2,\ \lambda = 1.8$ | $k = 4,\ \lambda = 1.8$ |
| EKP | ARI | 0.4105 ± 0.0052 | 0.0094 ± 0.0081 | 0.5349 ± 0 | 0.4885 ± 0 | 0.9179 ± 0.0415 |
| | Para | k = 10 | k = 2 | k = 2 | k = 2 | k = 4 |
| OCIL | ARI | **0.4305 ± 0.0492** | 0.0130 ± 0.0109 | **0.6165 ± 0.0058** | 0.0625 ± 0.1438 | 0.9185 ± 0.1719 |
| | Para | k = 10 | k = 2 | k = 2 | k = 2 | k = 4 |

**Table 8**
The $F_1$ values of the evaluated algorithms on categorical data sets.

| Data set | | LED Display Domain | Tic-Tac-Toe Game | Congressional Voting | Mushroom | Soybean |
|---|---|---|---|---|---|---|
| DP-MD-FN | $F_1$ | 0.2831 | **0.7866** | 0.8814 | **0.8944** | **1.0** |
| | Para | $d_c = 4\%,\ t = 20$ | $d_c = 1\%,\ t = 40$ | $d_c = 4\%,\ t = 20$ | $d_c = 2\%,\ t = 20$ | $d_c = 6\%,\ t = 5$ |
| K-Modes | $F_1$ | 0.1538 ± 0.1078 | 0.6436 ± 0.0199 | 0.8680 ± 0.0070 | 0.8575 ± 0.0945 | 0.7622 ± 0.2199 |
| | Para | k = 10 | k = 2 | k = 2 | k = 2 | k = 4 |
| KL-FCM-GM | $F_1$ | 0.4586 ± 0.0219 | 0.7069 ± 0 | 0.7957 ± 0 | 0.7729 ± 0.0252 | 0.9491 ± 0.1074 |
| | Para | $k = 10,\ \lambda = 0.8$ | $k = 2,\ \lambda = 1.5$ | $k = 2,\ \lambda = 1.5$ | $k = 2,\ \lambda = 1.8$ | $k = 4,\ \lambda = 1.8$ |
| EKP | $F_1$ | **0.4773 ± 0.0028** | 0.5303 ± 0.0079 | 0.7672 ± 0 | 0.7833 ± 0 | 0.9382 ± 0.0314 |
| | Para | k = 10 | k = 2 | k = 2 | k = 2 | k = 4 |
| OCIL | $F_1$ | 0.2182 ± 0.0986 | 0.6275 ± 0.0504 | **0.8926 ± 0.0016** | 0.6163 ± 0.0954 | 0.9143 ± 0.1807 |
| | Para | k = 10 | k = 2 | k = 2 | k = 2 | k = 4 |

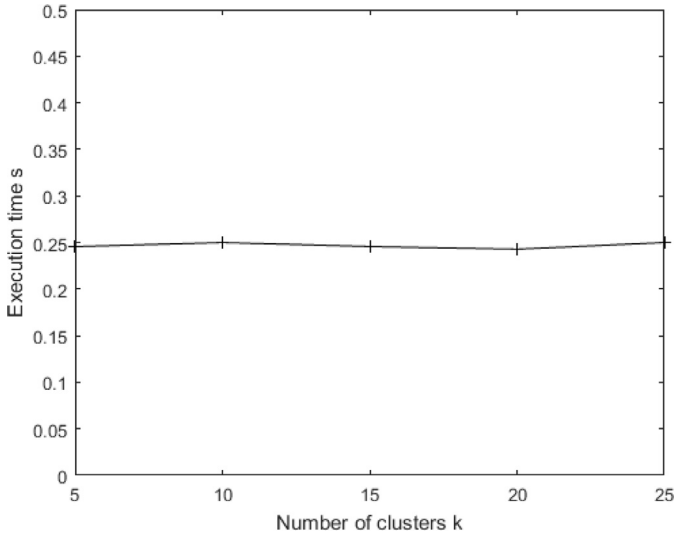



**Fig. 13.** The execution time vs. the number of clusters.

ing data sets, the ARI values of the clusters formed by these two methods are close, within a difference of 0.05 (within a difference of 0.01 on the LED Display Domain data set).

In Table 8, we list the five evaluated algorithms generating the $F_1$ values on categorical data sets. DP-MD-FN yields the best results among the five algorithms on three out of the five categorical data sets.

*5.3.2. Experiments on mixed type data sets*

In Table 9, we list the clustering accuracy of our proposed algorithm, K-Prototypes, KL-FCM-GM, EKP and OCIL on mixed type data sets. In Table 10, we list the five evaluated algorithms generating the NMI values on mixed type data sets. Tables 11 and 12 show the ARI and $F_1$ values of the five evaluated algorithms, respectively. In all cases, both the NMI values and the ARI values of the DP-MD-FN solutions are better than those found by other methods. DP-MD-FN yields the best results among the five algorithms on five out of the six categorical data sets according to the ACC and $F_1$ values. On the South African Hearth data set, our algorithm and KP tie for first place in term of the ACC and ARI values, while our method shows a small advantage compared with KP in term of the NMI and $F_1$ values. All validity indexes of the proposed method are significantly superior to those of other methods on the Credit Approval data set. Experimental results on the Credit Approval data set show that the ACC value of the proposed method is 6.51%, 27.54%, 31.55%, 20.34%, respectively higher than K-Prototypes, KL-FCM-GM, EKP and OCIL. And the $F_1$ value of the proposed method is 7.73%, 26.35%, 19.37%, 40.68%, higher than K-Prototypes, KL-FCM-GM, EKP and OCIL, respectively Both the NMI value and the ARI value of the proposed

**Table 9**
The ACC values of the evaluated algorithms on mixed type data sets.

| Data set | | South African Hearth | Heart | Australian Credit Approval | Credit Approval | Bank Marketing | KDD Cup 1999 |
|---|---|---|---|---|---|---|---|
| DP-MD-FN | ACC | **0.6537** | **0.8148** | **0.8551** | **0.8668** | 0.6357 | **0.9875** |
| | Para | $d_c = 1\%$, t = 20 | $d_c = 2\%$, t = 20 | $d_c = 4\%$, t = 15 | $d_c = 6\%$, t = 20 | $d_c = 0.5\%$,t = 40 | $d_c = 6\%$, t = 5 |
| K-Prototypes | ACC | **0.6537 ± 0** | 0.7830 ± 0.0445 | 0.7955 ± 0.0180 | 0.8017 ± 0.0122 | 0.6134 ± 0.0817 | 0.7500 ± 0 |
| | Para | k = 2, $\gamma = 0.3$ | k = 2, $\gamma = 0.2$ | k = 2, $\gamma = 1.0$ | k = 2, $\gamma = 0.1$ | k = 2, $\gamma = 0.2$ | k = 4, $\gamma = 1.2$ |
| KL-FCM-GM | ACC | 0.5584 ± 0 | 0.7926 ± 0 | 0.8319 ± 0 | 0.5914 ± 0.0847 | 0.5400 ± 0.0133 | 0.7714 ± 0.0565 |
| | Para | k = 2, $\lambda = 0.5$ | k = 2, $\lambda = 0.8$ | k = 2, $\lambda = 1.5$ | k = 2, $\lambda = 0.3$ | k = 2, $\lambda = 0.3$ | k = 4, $\lambda = 2.0$ |
| EKP | ACC | 0.6303 ± 0.0082 | 0.5926 ± 0 | 0.5590 ± 0.0014 | 0.5513 ± 0 | **0.8848 ± 0** | 0.5048 ± 0.0035 |
| | Para | k = 2, $\gamma = 0.1$ | k = 2, $\gamma = 0.7$ | k = 2, $\gamma = 1.1$ | k = 2, $\gamma = 1.3$ | k = 2, $\gamma = 0.7$ | k = 4, $\gamma = 0.6$ |
| OCIL | ACC | 0.6301 ± 0.0027 | 0.7411 ± 0.0678 | 0.6668 ± 0.0382 | 0.6634 ± 0.0407 | 0.6245 ± 0.0372 | 0.2500 ± 0 |
| | Para | k = 2 | k = 2 | k = 2 | k = 2 | k = 2 | k = 4 |

**Table 10**
The NMI values of the evaluated algorithms on mixed type data sets.

| Data set | | South African Hearth | Heart | Australian Credit Approval | Credit Approval | Bank Marketing | KDD Cup 1999 |
|---|---|---|---|---|---|---|---|
| DP-MD-FN | NMI | **0.0545** | **0.2791** | **0.4264** | **0.4482** | **0.0130** | **0.9637** |
| | Para | $d_c = 1\%$, t = 20 | $d_c = 2\%$, t = 20 | $d_c = 4\%$, t = 15 | $d_c = 6\%$, t = 20 | $d_c = 0.5\%$,t = 40 | $d_c = 6\%$, t = 5 |
| K-Prototypes | NMI | 0.0494 ± 0.0161 | 0.2549 ± 0.0600 | 0.2848 ± 0.0313 | 0.2894 ± 0.0291 | 0.0056 ± 0.0030 | 0.7391 ± 0.0011 |
| | Para | k = 2, $\gamma = 0.3$ | k = 2, $\gamma = 0.2$ | k = 2, $\gamma = 1.0$ | k = 2, $\gamma = 0.1$ | k = 2, $\gamma = 0.2$ | k = 4, $\gamma = 1.2$ |
| KL-FCM-GM | NMI | 0.0097 ± 0 | 0.2636 ± 0 | 0.3431 ± 0 | 0.0726 ± 0.0755 | 0.0096 ± 0.0046 | 0.7215 ± 0.0848 |
| | Para | k = 2, $\lambda = 0.5$ | k = 2, $\lambda = 0.8$ | k = 2, $\lambda = 1.5$ | k = 2, $\lambda = 0.3$ | k = 2, $\lambda = 0.3$ | k = 4, $\lambda = 2.0$ |
| EKP | NMI | 0.0031 ± 0.0002 | 0.0196 ± 0 | 0.0048 ± 0.0011 | 0.0053 ± 0 | 0.0013 ± 0 | 0.4056 ± 0.0028 |
| | Para | k = 2, $\gamma = 0.1$ | k = 2, $\gamma = 0.7$ | k = 2, $\gamma = 1.1$ | k = 2, $\gamma = 1.3$ | k = 2, $\gamma = 0.7$ | k = 4, $\gamma = 0.6$ |
| OCIL | NMI | 0.0460 ± 0.0013 | 0.1860 ± 0.0651 | 0.0904 ± 0.0308 | 0.0948 ± 0.0329 | 0.0056 ± 0.0029 | 0.0015 ± 0 |
| | Para | k = 2 | k = 2 | k = 2 | k = 2 | k = 2 | k = 4 |

**Table 11**
The ARI values of the evaluated algorithms on mixed type data sets.

| Data set | | South African Hearth | Heart | Australian Credit Approval | Credit Approval | Bank Marketing | KDD Cup 1999 |
|---|---|---|---|---|---|---|---|
| DP-MD-FN | ARI | **0.0903** | **0.3942** | **0.5036** | **0.5374** | **0.0393** | **0.9675** |
| | Para | $d_c = 1\%$, t = 20 | $d_c = 2\%$, t = 20 | $d_c = 4\%$, t = 15 | $d_c = 6\%$, t = 20 | $d_c = 0.5\%$,t = 40 | $d_c = 6\%$, t = 5 |
| K-Prototypes | ARI | **0.0903 ± 0** | 0.3227 ± 0.0793 | 0.3493 ± 0.0428 | 0.3635 ± 0.0300 | 0.0060 ± 0.0281 | 0.7098 ± 0.0003 |
| | Para | k = 2, $\gamma = 0.3$ | k = 2, $\gamma = 0.2$ | k = 2, $\gamma = 1.0$ | k = 2, $\gamma = 0.1$ | k = 2, $\gamma = 0.2$ | k = 4, $\gamma = 1.2$ |
| KL-FCM-GM | ARI | 0.0117 ± 0 | 0.3400 ± 0 | 0.4395 ± 0 | 0.0553 ± 0.1082 | 0.0052 ± 0.0055 | 0.6646 ± 0.0948 |
| | Para | k = 2, $\lambda = 0.5$ | k = 2, $\lambda = 0.8$ | k = 2, $\lambda = 1.5$ | k = 2, $\lambda = 0.3$ | k = 2, $\lambda = 0.3$ | k = 4, $\lambda = 2.0$ |
| EKP | ARI | 0.0183 ± 0.0064 | 0.0302 ± 0 | 0.4570 ± 0.0323 | 0.0019 ± 0 | 0 ± 0 | 0.3302 ± 0.0002 |
| | Para | k = 2, $\gamma = 0.1$ | k = 2, $\gamma = 0.7$ | k = 2, $\gamma = 1.1$ | k = 2, $\gamma = 1.3$ | k = 2, $\gamma = 0.7$ | k = 4, $\gamma = 0.6$ |
| OCIL | ARI | 0.0654 ± 0.0030 | 0.2455 ± 0.0872 | 0.1113 ± 0.0408 | 0.1086 ± 0.0393 | 0.0188 ± 0.0010 | 0 ± 0 |
| | Para | k = 2 | k = 2 | k = 2 | k = 2 | k = 2 | k = 4 |

**Table 12**
The $F_1$ values of the evaluated algorithms on mixed type data sets.

| Data set | | South African Hearth | Heart | Australian Credit Approval | Credit Approval | Bank Marketing | KDD Cup 1999 |
|---|---|---|---|---|---|---|---|
| DP-MD-FN | $F_1$ | **0.7752** | **0.8276** | **0.8503** | **0.8621** | 0.2602 | **0.9756** |
| | Para | $d_c = 1\%$, t = 20 | $d_c = 2\%$, t = 20 | $d_c = 4\%$, t = 15 | $d_c = 6\%$, t = 20 | $d_c = 0.5\%$,t = 40 | $d_c = 6\%$, t = 5 |
| K-Prototypes | $F_1$ | 0.7203 ± 0 | 0.8010 ± 0.0497 | 0.7781 ± 0.0047 | 0.7848 ± 0.0054 | 0.2018 ± 0.0587 | 0.6667 ± 0.0011 |
| | Para | k = 2, $\gamma = 0.3$ | k = 2, $\gamma = 0.2$ | k = 2, $\gamma = 1.0$ | k = 2, $\gamma = 0.1$ | k = 2, $\gamma = 0.2$ | k = 4, $\gamma = 1.2$ |
| KL-FCM-GM | $F_1$ | 0.5272 ± 0 | 0.6710 ± 0 | 0.7281 ± 0 | 0.5986 ± 0.0377 | 0.6172 ± 0.0014 | 0.7598 ± 0.0608 |
| | Para | k = 2, $\lambda = 0.5$ | k = 2, $\lambda = 0.8$ | k = 2, $\lambda = 1.5$ | k = 2, $\lambda = 0.3$ | k = 2, $\lambda = 0.3$ | k = 4, $\lambda = 2.0$ |
| EKP | $F_1$ | 0.6420 ± 0.0227 | 0.5306 ± 0 | 0.6703 ± 0.5217 | 0.6684 ± 0 | **0.8864 ± 0** | 0.5683 ± 0.0005 |
| | Para | k = 2, $\gamma = 0.1$ | k = 2, $\gamma = 0.7$ | k = 2, $\gamma = 1.1$ | k = 2, $\gamma = 1.3$ | k = 2, $\gamma = 0.7$ | k = 4, $\gamma = 0.6$ |
| OCIL | $F_1$ | 0.5295 ± 0.0962 | 0.7746 ± 0.0244 | 0.4584 ± 0.1527 | 0.4553 ± 0.1570 | 0.2280 ± 0.0019 | 0.4 ± 0 |
| | Para | k = 2 | k = 2 | k = 2 | k = 2 | k = 2 | k = 4 |

method is one order of magnitude higher than those of KL-FCM-GM, EKP and OCIL. On Bank Marketing data set, all objects are divided into two classes, "yes" (class #1), including 4000 objects, and "no" (class #2), including 521 objects. According to its class distribution, this data set is a typical imbalanced data set. Its imbalance ratio value is higher than 7. On this data set, our method shows a small advantage compared with others in terms of the NMI and ARI values. It is notable that EKP obtains the best performance in terms of the ACC and $F_1$ values, but performs very poor in terms of the NMI and ARI values. The ACC and $F_1$ indexes are unsuitable for imbalanced data sets. A method divides all objects into two classes, cluster #1 including only 1 object, cluster #2 including others. When the data set is imbalanced, the method obtains very high ACC and $F_1$ values. However, we think the result is not a good performance. As a result, we think EKP does not reveal good structure of clusters on Bank Marketing data set. Experimental results on the KDD Cup 1999 data set show that the ACC value of the proposed method is 21.61% higher than that of the second-best result (i.e., KL-FCM-GM). Actually, when the parameter $d_c$ is set to 8% or higher values, the proposed method can find the optimal structure of clusters on the data set. However, these values are out of range which is defined in the beginning of this section. As can be seen from Tables 5–12, all validity indexes obtained by DP-MD-FN are, in most cases, superior to those obtained by other methods. In ad-

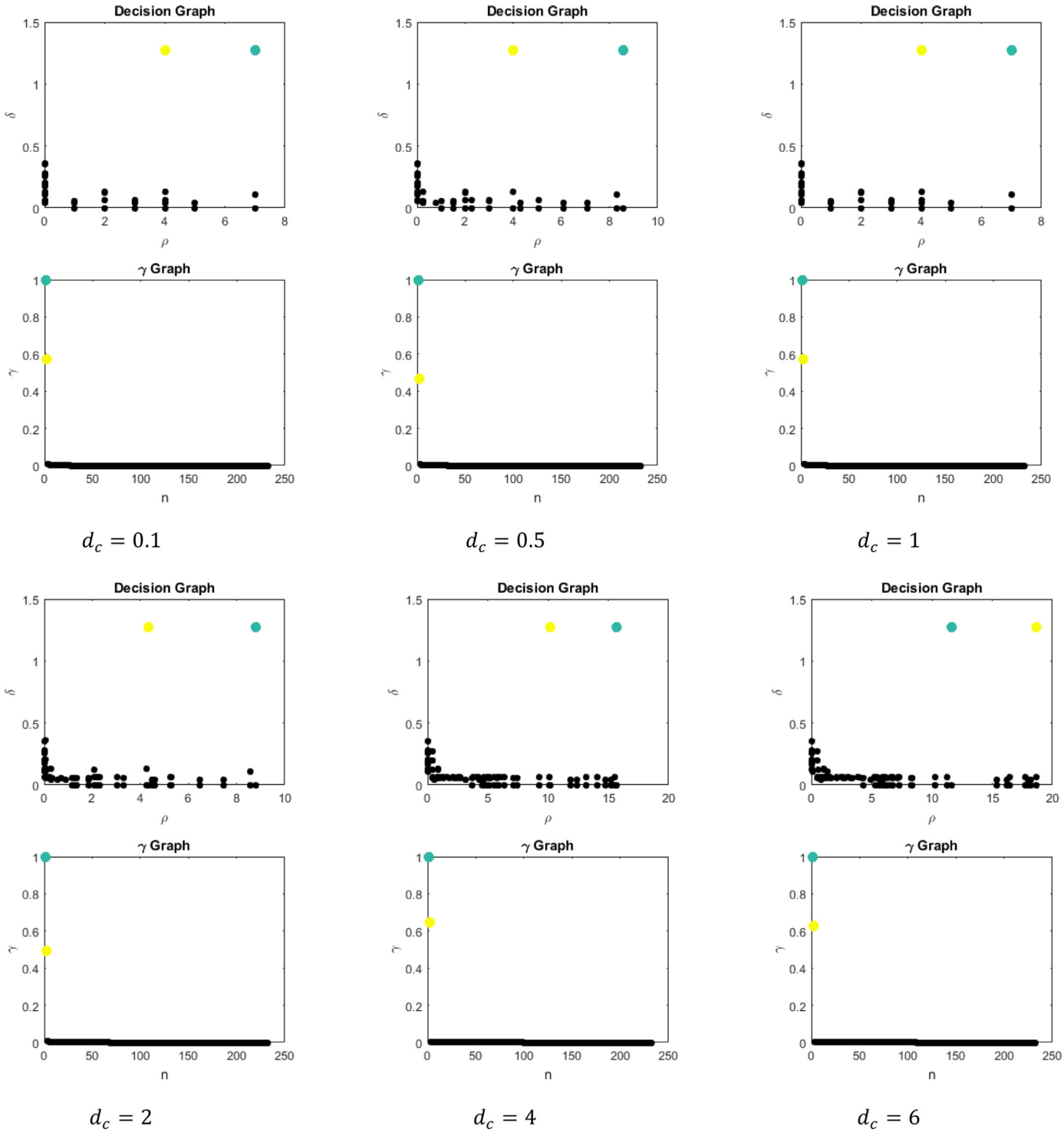



**Fig. 14.** The decision graphs of DP-MD-FN on the Congressional Voting data set.

dition, as the number of the attributes increases, the performance of the proposed method does not deteriorate. And it can even find the optimal structure of clusters on Soybean and KDD Cup 1999 data sets which have more number of attributes compared with others.

In Table 13, we list the average ranking of clustering algorithms obtained by the Friedman's test based on the ACC value. The proposed DP-MD-FN algorithm is ranked first. It is worth noting that K-prototypes is an extension of K-Modes. Therefore, we treat K-prototypes and K-Modes as one algorithm. As a consequence, the number of algorithms $k_a$ is 5 and the number of data sets $n_{ds}$ is 11. The $p$-value computed by the Friedman test and the Iman–Davenport test are given in Table 14, which both are smaller than

**Table 13**
Average ranking of clustering algorithms based on the ACC value.

| Algorithm | DP-MD-FN | K-Prototypes | KL-FCM-GM | EKP | OCIL |
|---|---|---|---|---|---|
| Ranking | 1.0909 | 3.4545 | 3.1818 | 3.6364 | 3.6364 |

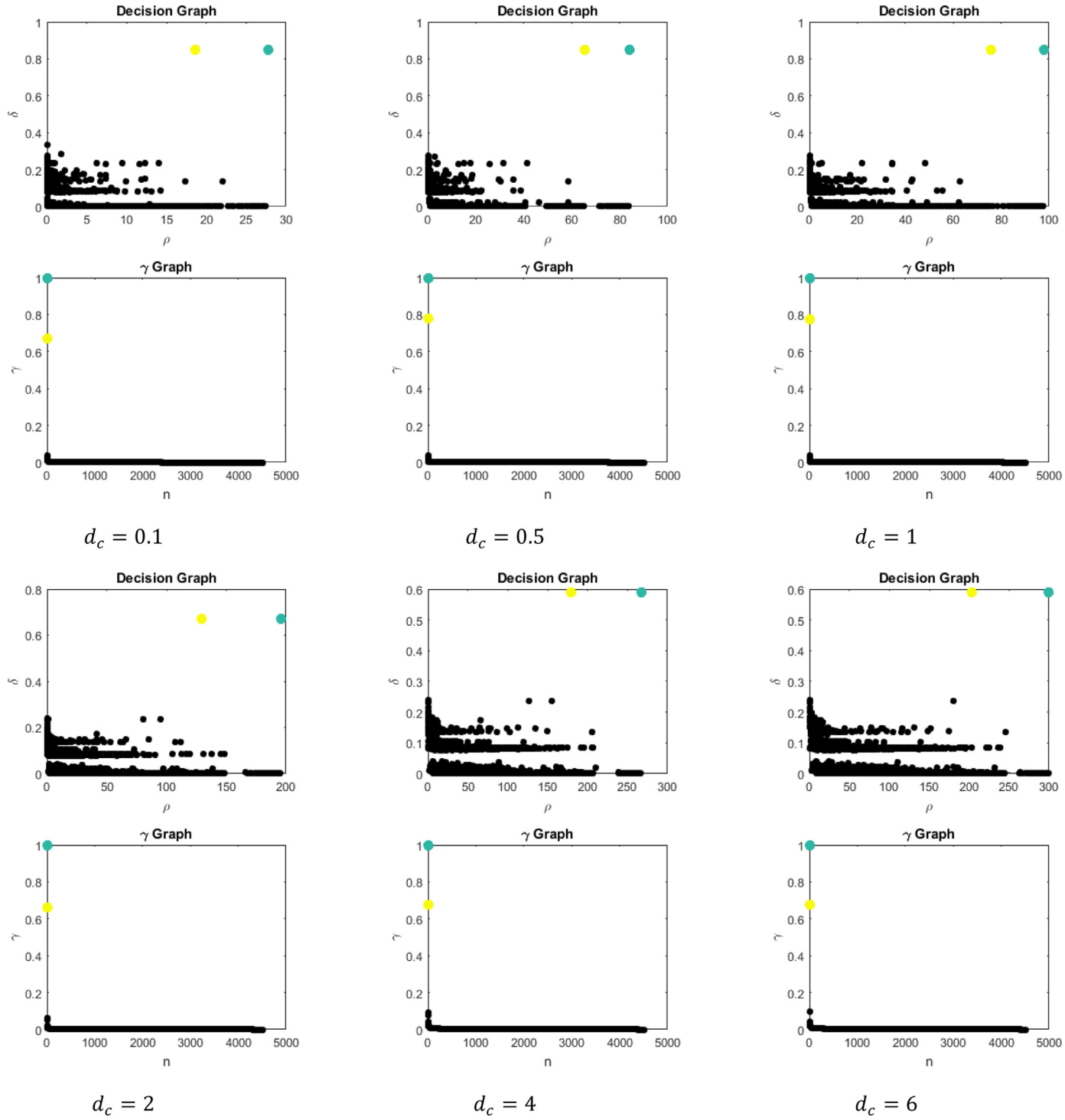



**Fig. 15.** The decision graphs of DP-MD-FN on the Bank Marketing data set.

**Table 14**
Results of Friedman's and Iman–Davenport's tests based on the ACC value.

| Method | Statistical value | *p*-value | Hypothesis |
|---|---|---|---|
| Friedman | 20.6546 | 0.0004 | Rejected |
| Iman–Davenport | 8.8474 | 0.00003 | Rejected |

0.05. Thus the null hypothesis of equivalent performance are rejected. That means significant differences among the performance of all the clustering algorithms do exist.

The same procedure is performed to check whether there are significant differences in terms of the other validity indexes (NMI, ARI and $F_1$). Tables 15–20 show that the proposed algorithm is ranked first and there are significant differences in the results of the algorithms.

Figs. 14 and 15 display the decision graph and $\gamma$-graph results of the proposed algorithm for clustering the Congressional Vot-

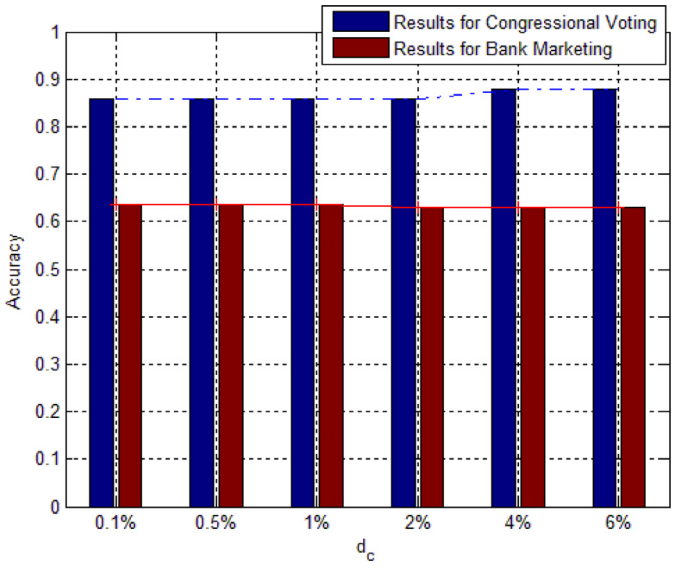



**Fig. 16.** The results of DP-MD-FN on Congressional Voting and Bank Marketing.

**Table 15**
Average ranking of clustering algorithms based on the NMI value.

| Algorithm | DP-MD-FN | K-Prototypes | KL-FCM-GM | EKP | OCIL |
|---|---|---|---|---|---|
| Ranking | 1 | 3.1818 | 3.2727 | 4 | 3.5454 |

**Table 16**
Results of Friedman's and Iman–Davenport's tests based on the NMI value.

| Method | Statistical value | *p*-value | Hypothesis |
|---|---|---|---|
| Friedman | 23.7818 | 0.00008 | Rejected |
| Iman–Davenport | 11.7626 | 0.000002 | Rejected |

**Table 17**
Average ranking of clustering algorithms based on the ARI value.

| Algorithm | DP-MD-FN | K-Prototypes | KL-FCM-GM | EKP | OCIL |
|---|---|---|---|---|---|
| Ranking | 1.2727 | 3.2727 | 3.3636 | 3.8182 | 3.2727 |

**Table 18**
Results of Friedman's and Iman–Davenport's tests based on the ARI value.

| Method | Statistical value | *p*-value | Hypothesis |
|---|---|---|---|
| Friedman | 17.3091 | 0.0017 | Rejected |
| Iman–Davenport | 6.4850 | 0.0004 | Rejected |

**Table 19**
Average ranking of clustering algorithms based on the $F_1$ value.

| Algorithm | DP-MD-FN | K-Prototypes | KL-FCM-GM | EKP | OCIL |
|---|---|---|---|---|---|
| Ranking | 1.4545 | 3.0909 | 3.1818 | 3.3636 | 3.9091 |

**Table 20**
Results of Friedman's and Iman–Davenport's tests based on the $F_1$ value.

| Method | Statistical value | *p*-value | Hypothesis |
|---|---|---|---|
| Friedman | 14.9091 | 0.0049 | Rejected |
| Iman–Davenport | 5.1250 | 0.0020 | Rejected |

ing data set and the Bank Marketing data set in all $d_c$ parameter values, respectively. The $\gamma$-graph results generated by our algorithm have a great performance. The $\gamma$ values of centers are far greater than those of the other points. From the decision graphs, we can see that the global distribution of the points is a "straight line", but the centers of the clusters are the outliers deviating from the global distribution. The proposed automatic cluster center selection method can find the right number of clusters based on the $\gamma$-graph results. The selected cluster centers are represented in different color (yellow and green). The corresponding decision graphs obtained by propagation of the label are also showed in Figs. 14 and 15. In addition, Fig. 16 shows the ACC values obtained by the proposed algorithm in the above cases. We can see that the curves of the ACC values of our algorithm are almost flat. It means that DP-MD-FN is still robust with respect to choosing $d_c$ when it deals with real world data sets.

## 6. Discussion and conclusions

Similarity measure and clustering algorithm are the two primary steps in clustering process. From the two aspects, we explore the differences between the existing methods and DP-MD-FN.

Firstly, we discuss similarity measure. Compared by some preprocessing methods, the proposed similarity measure can better reveal the structure of the data sets. K-Prototypes and its variations need to choose the parameter $\gamma$ to avoid favoring either type of attribute. However, experimental results show that the parameter has great influence on these algorithms. The proposed similarity measure does not require the weight $\gamma$. Indeed, SBAC also does not need to adjust the parameter between categorical and numerical values. However, a computational efficient similarity measure remains to be developed. In addition, OCIL only can measure the similarity between an object and a cluster. Thus this point may be a roadblock for the broad application of the similarity measure. We present a unified similarity metric for measuring numerical and categorical values. It does not need feature transformation and parameter adjustment between categorical and numerical values. The strong points of such similarity measure are the ease of application and the broad coverage of the method, that is to say for the clustering purpose of mixed type data.

Secondly, we discuss clustering algorithm. Currently most algorithms have shortcomings including low clustering quality and poor robustness. The main reason for this phenomenon may be that most methods, e.g. OCIL, KP and its variations, use the K-Means paradigm to cluster mixed type data. Thus they are sensitive to initialization and are generally unsuitable for non-spherical distribution data. By contrast, we use peak density clustering algorithm which is a density-based data clustering. DP is able to detect non-spherical distribution data and does not need to pre-assign the number of clusters. To further improve the robustness of DP, we use fuzzy neighborhood relation to redefine the local density, which integrates the speed of DP clustering algorithm with the robustness of FJP algorithm. To avoid manually selecting the cluster centers, we develop an automatic cluster center selection method. We integrate this entropy-based strategy with the improved DP clustering method so that they play their own strengths in order to achieve higher clustering quality and better robustness.

On the basis of the new criterion, the new local density and the cluster center selection method, we develop an entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood (DP-MD-FN). We design 12 synthetic data sets and compare our algorithm with some traditional clustering for mixed type data on these data sets. Experimental results reveal that, of the five methods, DP-MD-FN is the most robust and does an excellent job when the degradation levels or the number of the irrelevant attributes increases. Also, it is less sensitive to the choice

of $d_c$. In addition, we design 17 synthetic data sets only containing categorical attributes and test the scalability of the proposed algorithm. Finally, 11 UCI data sets are used to test the performance of the proposed algorithm. In most cases, the results obtained by DP-MD-FN are superior to those obtained by other methods.

Future works will develop an adaptive DP-MD-FN with non-parametric method. DP-MD-FN costs much time in the calculation of the similarity matrix, thus we will try to introduce the idea of the grid into our method. The cost is only associated with the number of cells. And the number of cells $K$ is far less than the number of objects $N$.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. 61672522 and 61379101), the China Postdoctoral Science Foundation under Grant No.2016M601910, the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), and the Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET).

## References

[1] N. Iam-On, T. Boongoen, N. Kongkotchawan, A new link-based method to ensemble clustering and cancer microarray data analysis, Int. J. Collaborative Intell. 1 (1) (2014) 45–67.
[2] S.A. Ludwig, MapReduce-based fuzzy c-means clustering algorithm: implementation and scalability, Int. J. Mach. Learn. Cybern. 6 (6) (2015) 923–934.
[3] X. Li, Y. Liang, Y. Cai, CC-K-means: a candidate centres-based K-means algorithm for text data, Int. J. Collaborative Intell. 1 (3) (2016) 189–204.
[4] C. Bouras, V. Tsogkas, Assisting cluster coherency via n-grams and clustering as a tool to deal with the new user problem, Int. J. Mach. Learn. Cybern. 7 (2) (2016) 171–184.
[5] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1(14), 1967, pp. 281–297.
[6] C. Li, G. Biswas, Unsupervised learning with mixed numeric and nominal data, IEEE Trans. Knowl. Data Eng. 14 (4) (2002) 673–690.
[7] Z. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, Data Min. Knowl. Discovery 2 (3) (1998) 283–304.
[8] Y.M. Cheung, H. Jia, A unified metric for categorical and numerical attributes in data clustering, in: Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2013, pp. 135–146.
[9] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[10] E.N. Nasibov, G. Ulutagay, A new unsupervised approach for fuzzy clustering, Fuzzy Sets Syst. 158 (19) (2007) 2118–2133.
[11] H. Ralambondrainy, A conceptual version of the K-means algorithm, Pattern Recognit. Lett. 16 (11) (1995) 1147–1157.
[12] C.C. Hsu, Generalizing self-organizing map for categorical data, IEEE Trans. Neural Netw. 17 (2) (2006) 294–304.
[13] C.C. Hsu, C.L. Chen, Y.W. Su, Hierarchical clustering of mixed data based on distance hierarchy, Inf. Sci. 177 (20) (2007) 4474–4492.
[14] Z. Huang, A fast clustering algorithm to cluster very large categorical data sets in data mining, in: Proceedings of the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, 1997, pp. 1–8.
[15] S.P. Chatzis, A fuzzy c-means-type algorithm for clustering of data with mixed numeric and categorical attributes employing a probabilistic dissimilarity functional, Expert Syst. Appl. 38 (7) (2011) 8684–8689.
[16] J. Ji, W. Pang, C. Zhou, et al., A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data, Knowl.-Based Syst. 30 (2012) 129–135.
[17] Z. Zheng, M. Gong, J. Ma, et al., Unsupervised evolutionary clustering algorithm for mixed type data, in: Proceedings of 2010 IEEE Congress on Evolutionary Computation, 2010, pp. 1–8.
[18] L. Huang, G. Wang, Y. Wang, et al., A link density clustering algorithm based on automatically selecting density peaks for overlapping community detection, Int. J. Modern Phys. B 30 (24) (2016) 1650167.
[19] Y.W. Chen, D.H. Lai, H. Qi, et al., A new method to estimate ages of facial image for large database, Multimedia Tools. Appl. 75 (5) (2016) 2877–2895.
[20] B. Wang, J. Zhang, Y. Liu, et al., Density peaks clustering based integrate framework for multi-document summarization, CAAI Trans. Intell. Technol. 2 (1) (2017) 26–30.
[21] T. Ma, Y. Wang, M. Tang, et al., LED: A fast overlapping communities detection algorithm based on structural clustering, Neurocomputing 207 (2016) 488–500.
[22] Z. Liang, P. Chen, Delta-density based clustering with a divide-and-conquer strategy: 3DC clustering, Pattern Recognit. Lett. 73 (2016) 52–59.
[23] 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.
[24] X. Xu, S. Ding, M. Du, et al., DPCG: an efficient density peaks clustering algorithm based on grid, Int. J. Mach. Learn. Cybern. (2016), doi:10.1007/s13042-016-0603-2.
[25] E.N. Nasibov, G. Ulutagay, Robustness of density-based clustering methods with various neighborhood relations, Fuzzy Sets Syst. 160 (24) (2009) 3601–3615.
[26] G. Ulutagay, E.N. Nasibov, Influence of transitive closure complexity in FJP-based clustering algorithms, Turkish J. Fuzzy Syst. 1 (1) (2010) 3–20.
[27] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (3) (1948) 379–423.
[28] J. Yao, M. Dash, S.T. Tan, et al., Entropy-based fuzzy clustering and fuzzy modeling, Fuzzy Sets Syst. 113 (3) (2000) 381–388.
[29] D. Barbará, Y. Li, J. Couto, COOLCAT: an entropy-based algorithm for categorical clustering, in: Proceedings of the 11th International Conference on Information and Knowledge Management, 2002, pp. 582–589.
[30] C.H. Cheng, A.W. Fu, Y. Zhang, Entropy-based subspace clustering for mining numerical data, in: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999, pp. 84–93.
[31] F.G. Ashby, N.A. Perrin, Toward a unified theory of similarity and recognition, Psychol. Rev. 95 (1) (1988) 124–150.
[32] S. Santini, R. Jain, Similarity measures, IEEE Trans. Pattern Anal. Mach. Intell. 21 (9) (1999) 871–883.
[33] R.N. Shepard, Toward a universal law of generalization for psychological science, Science 237 (4820) (1987) 1317–1323.
[34] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, Wiley, New York, USA, 1990.
[35] J. Basak, R. Krishnapuram, Interpretable hierarchical clustering by constructing an unsupervised decision tree, IEEE Trans. Knowl. Data Eng. 17 (1) (2005) 121–132.
[36] Z. Pan, J. Lei, Y. Zhang, et al., Fast motion estimation based on content property for low-complexity H.265/HEVC encoder, IEEE Trans. Broadcast. 62 (3) (2016) 675–684.
[37] J. Jiang, X. Yan, Z. Yu, et al., A Chinese expert disambiguation method based on semi-supervised graph clustering, Int. J. Mach. Learn. Cybern. 6 (2) (2015) 197–204.
[38] W. Chen, Y. Song, H. Bai, et al., Parallel spectral clustering in distributed systems, IEEE Trans. Pattern Anal. Mach. Intell. 33 (3) (2011) 568–586.
[39] A. Strehl, J. Ghosh, Cluster ensembles- knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2003) 583–617.
[40] H. Jia, S. Ding, M. Du, et al., Approximate normalized cuts without Eigen-decomposition, Inf. Sci. 374 (2016) 135–150.
[41] S. Wagner, D. Wagner, Comparing Clusterings—An Overview, Faculty of Informatics, Universität Karlsruhe (TH), 2007 Technical Report.
[42] Y. Xu, J. Dong, B. Zhang, et al., Background modeling methods in video analysis: a review and comparative evaluation, CAAI Trans. Intell. Technol. 1 (1) (2016) 43–60.
[43] D.M.W. Powers, Evaluation: from precision, recall and F-Factor to ROC, informedness, markedness & correlation, J. Mach. Learn. Technol. 2 (2011) 2229–3981.
[44] J. Derrac, S. García, D. Molina, et al., A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[45] M. Gong, S. Wang, W. Liu, et al., Evolutionary computation in China: a literature survey, CAAI Trans. Intell. Technol. 1 (4) (2016) 334–354.
[46] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.
[47] S. Ding, Y. An, X. Zhang, et al., Wavelet twin support vector machines based on glowworm swarm optimization, Neurocomputing 225 (2017) 157–163.
[48] A. Hatamlou, Black hole: a new heuristic optimization approach for data clustering, Inf. Sci. 222 (2013) 175–184.