Information Sciences 332 (2016) 167–183

Contents lists available at ScienceDirect

Information Sciences

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

Initialization of *K*-modes clustering using outlier detection techniques

Feng Jiang[a,*], Guozhu Liu[a], Junwei Du[a], Yuefei Sui[b]

[a] *College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, P.R. China*
[b] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, P.R. China*

ARTICLE INFO

Article history:
Received 22 April 2015
Revised 4 September 2015
Accepted 5 November 2015
Available online 6 November 2015

Keywords:
K-modes clustering
Outlier detection
Initial cluster centers
Distance
Partition entropy

ABSTRACT

The *K*-modes clustering has received much attention, since it works well for categorical data sets. However, the performance of *K*-modes clustering is especially sensitive to the selection of initial cluster centers. Therefore, choosing the proper initial cluster centers is a key step for *K*-modes clustering. In this paper, we consider the initialization of *K*-modes clustering from the view of outlier detection. We present two different initialization algorithms for *K*-modes clustering, where the first is based on the traditional distance-based outlier detection technique, and the second is based on the partition entropy-based outlier detection technique. By using the above two outlier detection techniques to calculate the degree of outlierness of each object, our algorithms can guarantee that the chosen initial cluster centers are not outliers. Moreover, during the process of initialization, we adopt a new distance metric — weighted matching distance metric, to calculate the distance between two objects described by categorical attributes. Experimental results on several UCI data sets demonstrate the effectiveness of our initialization algorithms for *K*-modes clustering.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

As one of the most important tasks in data mining, clustering aims to group a set of objects into clusters such that the objects within the same cluster are more similar to each other than to objects in other clusters [23,55]. The current clustering algorithms can be generally classified into five types: hierarchical clustering [23,26], partitional clustering [1,47], density-based clustering [21], grid-based clustering and model-based clustering. As a well-known partitional clustering algorithm, the *K*-means algorithm is very efficient for processing large data sets [1,9,47]. However, the *K*-means algorithm works only on the data sets with numeric attributes, which limits the use of it in solving categorical data clustering problems. To remove the numeric-only limitation of the *K*-means algorithm, [28,29] proposed the *K*-modes algorithm, which extends the *K*-means algorithm by using: (1) a simple matching dissimilarity measure for categorical objects; (2) modes instead of means for clusters; and (3) a frequency-based method to update modes in the clustering process to minimize the clustering cost function [58].

In general, the *K*-modes algorithm is faster than the *K*-means algorithm since the former needs less iterations to converge [29]. It should be noted, however, that the *K*-modes algorithm uses the same clustering process as the *K*-means algorithm except for the clustering cost function, and hence it also suffers from the same problems as the *K*-means algorithm. For instance, the performance of *K*-means algorithm is sensitive to the choice of initial cluster centers, and an improper choice may result in

[*] Corresponding author. Tel.: +86 532 88959036; fax: +86 532 88959036.
E-mail address: jiangkong@163.net (F. Jiang).

http://dx.doi.org/10.1016/j.ins.2015.11.005
0020-0255/© 2015 Elsevier Inc. All rights reserved.

undesirable cluster structures [14]. Similarly, the performance of *K*-modes algorithm is also sensitive to the choice of initial cluster centers [40], hence, it is important to provide the *K*-modes clustering with good initial cluster centers. However, there are still no generally accepted initialization methods for *K*-modes clustering.

In this paper, we consider the initialization of *K*-modes clustering from the view of outlier detection [42]. Based on the idea that outliers should not be selected as initial cluster centers, we combine the selection of initial cluster centers in *K*-modes clustering with the detection of outliers. To solve the initialization problem of *K*-modes clustering, we first propose an initialization algorithm (called *Ini_Distance*) for *K*-modes clustering via the traditional distance-based outlier detection technique [42]. Second, to avoid the problems of distance-based technique, we further present a partition entropy-based outlier detection technique within the framework of rough sets [53], and design an initialization algorithm (called *Ini_Entropy*) via the partition entropy-based technique.

A brief description of algorithms *Ini_Distance* and *Ini_Entropy* is given as follows. Given a data set *T*, for any candidate center *x* in *T*, to decide whether *x* should be selected as an initial cluster center, our algorithms first calculate the degree of outlierness of *x* via a given outlier detection technique. Second, the distance between *x* and each currently existing initial center is calculated. Finally, the degree of outlierness of *x* and the distance between *x* and each existing initial center are used together to decide whether *x* should be selected as an initial cluster center. That is, if the degree of outlierness of *x* is low, and the distance between *x* and each existing initial center is always large, then the possibility of *x* being an initial center will be high. By using the degree of outlierness of each candidate center to select initial centers, our algorithms can avoid that outliers are selected as initial centers. In addition, by calculating the distances between candidate centers and all currently existing initial centers, our algorithms can avoid that various initial centers come from the same cluster.

During the process of initialization, we need to calculate the distance between two objects described by categorical attributes, and the simple matching distance metric [37] is usually used to do that. In this paper, we adopt a new distance metric — weighted matching distance metric, which can obtain better results than the simple metric. Moreover, to reduce the time complexities of algorithms *Ini_Distance* and *Ini_Entropy*, we use the counting sort-based method to compute the partition of universe *U* induced by a given indiscernibility relation [65].

The remainder of this paper is organized as follows. In Section 2, we give a brief overview of the related work. In Section 3, we introduce some preliminaries that are relevant to this paper. In Section 4, we propose a new distance metric for categorical data. In Section 5, we propose an initialization algorithm for *K*-modes clustering via the distance-based outlier detection technology. In Section 5, we present a partition entropy-based outlier detection technique within the framework of rough sets. Moreover, we design an initialization algorithm for *K*-modes clustering via the partition entropy-based outlier detection technique. Experimental results are given in Section 7. Finally, Section 8 concludes the paper.

2. Related work

In recent years, various methods have been proposed to initialize cluster centers for *K*-modes clustering [6,8,10,15,25,29,38–40,61,64]. The random initialization method has been widely used in *K*-modes clustering for its simplicity. However, the random method does not guarantee a unique clustering result, and very poor clustering results may occur in some cases [40]. To obtain desirable clustering results, the *K*-modes algorithm must be executed several times. Hence, it is necessary to design a non-random initialization algorithm for *K*-modes clustering.

Huang [29] proposed two non-random initialization methods for *K*-modes clustering, where the first method selects the first *K* objects from the data set as initial cluster centers, and the second method assigns the most frequent categories equally to the *K* initial cluster centers. However, both of the two methods have certain problems. The first method is efficient only if the first *K* objects come from *K* disjoint clusters. Although the second method aims to select diverse initial centers, a uniform criterion is not yet provided for selecting initial centers [15].

Sun et al. [61] proposed an initialization method for *K*-modes clustering, by using the iterative initial points refinement algorithm given by Bradley and Fayyad [12]. Barbara et al. [10] found the *K* most 'dissimilar' objects from the data set by maximizing the minimum pairwise entropy of the chosen points, and used the *K* objects as initial seeds. Khan and Ahmad [38] used the density-based multiscale data condensation approach [49] to select initial centers for *K*-modes clustering. He [25] proposed two initialization methods for *K*-modes clustering based on the farthest-point heuristic. Khan and Kant [39] proposed an initialization method for *K*-modes clustering, which is based on the idea of evidence accumulation for combining the results of multiple clusterings [22].

Wu et al. [64] proposed a density based initialization method for *K*-modes clustering. Cao et al. [15] presented a method to select initial cluster centers by considering the distance between objects and the density of each object. Bai et al. [6] proposed an initialization method for *K*-modes clustering, which can not only select initial cluster centers, but also determine the number of clusters. Bai et al. [8] proposed a global *K*-modes algorithm. The algorithm randomly selects K_{ini} initial centers ($K_{ini} \gg K$, where *K* is the predefined number of clusters), and then eliminates the redundant centers by using an iterative optimization process. Khan and Ahmad [40] presented an initialization algorithm for *K*-modes clustering by performing multiple clustering of data based on the attribute values present in different attributes.

As mentioned above, in this paper we consider the initialization of *K*-modes clustering from the view of outlier detection. In recent years, interest in the detection of outliers has grown considerably [2–4,13,19,24,34,35,42]. As an important task of data mining, outlier detection concentrates on the behavior of a small amount of objects that are exceptional when compared with the rest of the data objects. Outlier detection was first discussed in statistics. To solve the problems of statistics-based outlier detection technique, Knorr and Ng [42] proposed the distance-based outlier detection technique, which calculates each object's distance to its neighbors for finding outliers. The distance-based technique is now widely used for outlier detection, since it does not need to make any assumptions about the data distribution, and can be applied to any feature space for which we can define a distance metric.

As two different tasks in data mining, clustering and outlier detection have a close relationship. Clustering can be used for outlier detection, and outliers may emerge as small clusters far from other clusters. In many cases, the two problems are tightly coupled, for example, outliers may have a disproportionate impact on the shape of clusters, which in turn mask obvious outliers [18]. By now, many clustering techniques have been employed to solve the problems of outlier detection [24,32], but few researchers have attempted to solve the problems of clustering by virtue of outlier detection techniques. It is meaningful to address the issue of clustering from the view of outlier detection.

3. Preliminaries

In this section, we introduce some basic concepts in rough set theory [53]. In rough set terminology, a data table is also called an *information table*, which is formally defined as follows [53,54].

Definition 3.1. An information table is a quadruple $IS = (U, A, V, f)$, where

(1) U is a non-empty and finite set of objects;
(2) A is a non-empty and finite set of attributes;
(3) V is the union of attribute domains, i.e., $V = \bigcup_{a \in A} V_a$, where V_a denotes the domain of attribute a;
(4) f: $U \times A \rightarrow V$ is an information function such that for any $a \in A$ and $x \in U$, $f(x, a) \in V_a$.

Given an information table $IS = (U, A, V, f)$, if each attribute $a \in A$ is a categorical attribute, then we call IS a categorical information table. In this paper, we may assume that all information tables are categorical.

Given an information table $IS = (U, A, V, f)$, for any subset $B \subseteq A$, we call binary relation $IND(B)$ an indiscernibility relation on U, which is defined as follows:

$$IND(B) = \{(x, y) \in U \times U : \forall a \in B(f(x, a) = f(y, a))\}. \tag{1}$$

$IND(B)$ can be viewed as a given knowledge in IS, which partitions U into disjoint equivalence classes. Let $U/IND(B)$ denote the family of all equivalence classes induced by $IND(B)$, and for any $x \in U$, let $[x]_B \in U/IND(B)$ denote the equivalence class that contains x. $U/IND(B)$ is also called the partition of U induced by $IND(B)$ [53,54].

As one of the most important metrics in information theory, Shannon entropy [59] and its variants have been widely used for characterizing uncertainty in rough set theory. Different models of entropy have been proposed within the framework of rough sets [11,20,27,36,43–45,48,52,57,60,62,63], where the model of partition entropy is widely used.

Definition 3.2. Given an information table $IS = (U, A, V, f)$, for any $B \subseteq A$, let $U/IND(B) = \{X_1, \ldots, X_p\}$ be the partition of U induced by the indiscernibility relation $IND(B)$. The partition entropy $PE(B)$ of $U/IND(B)$ in IS is defined as [20]:

$$PE(B) = -\sum_{i=1}^{p} \frac{|X_i|}{|U|} \log_2 \frac{|X_i|}{|U|}, \tag{2}$$

where $|X_i|/|U|$ denotes the probability of any object $x \in U$ being in equivalence class X_i, $1 \le i \le p$.

The partition entropy $PE(B)$ reaches the maximum value $\log_2|U|$ if each element in $U/IND(B)$ is a singleton subset of U, and $PE(B)$ reaches the minimum value 0 if $U/IND(B) = \{U\}$.

Definition 3.3. Given an information table $IS = (U, A, V, f)$, for any $a \in A$, the partition entropy-based significance of attribute a in IS is defined as

$$Sig(a) = \frac{PE(A) - PE(A - \{a\})}{PE(A) + PE(A - \{a\})}, \tag{3}$$

where for any $B \subseteq A$, $PE(B)$ denotes the partition entropy of $U/IND(B)$.

Given an information table $IS = (U, A, V, f)$, for any $B_1, B_2 \subseteq A$, if $B_1 \subseteq B_2$, then from the monotonicity of partition entropy, we can obtain that $PE(B_1) \le PE(B_2)$. Therefore, from Definition 3.3, we can further obtain that for any attribute $a \in A$, $0 \le Sig(a) \le 1$.

4. The weighted matching distance metric

During the process of initialization, we need to calculate the distance between two objects described by categorical attributes. The current initialization methods for K-modes clustering usually use the simple matching distance metric [37] to measure the distance between two objects described by categorical attributes.

Definition 4.1 (Simple matching distance metric)**.** Given an information table $IS = (U, A, V, f)$, for any two objects $x, y \in U$, the simple matching distance between x and y is defined as follows:

$$d(x, y) = \sum_{a \in A} \delta_a(x, y), \tag{4}$$

where $\delta_a(x, y)$ denotes the simple matching distance of x and y on attribute a, which is defined as:

$$\delta_a(x, y) = \begin{cases} 1, & \text{if } f(x, a) \neq f(y, a); \\ 0, & \text{otherwise.} \end{cases}$$

The simple matching distance metric has its limitations [7,16,50,58]. In Definition 4.1, all attributes are used with the same significance (or importance) level, say 1. That is, the weight of all attributes is considered to be equal. However, in a real data set the effect of one attribute may be different from other attributes [31]. Therefore, it is more appropriate to assign different weights such as 1.5, 0.8, 1.2 and 1.1 to various attributes. In the following, we propose a weighted matching distance metric, where the weight of each attribute is calculated via the partition entropy-based attribute significance.

Definition 4.2 (Weight of attribute)**.** Given an information table $IS = (U, A, V, f)$, for any $a \in A$, let $Sig(a)$ denote the partition entropy-based significance of attribute a in IS, where $0 \leq Sig(a) \leq 1$. The weight of attribute a in IS is defined as:

$$weight(a) = \begin{cases} \frac{1}{2} \times \left(1 + \frac{count_{zero}}{|A| + \sqrt{|A| - count_{zero}}}\right), & \text{if } Sig(a) = 0; \\ 1 + Sig(a), & \text{if } Sig(a) > 0. \end{cases}$$

In the above definition, $|A|$ denotes the cardinality of set A, and $count_{zero}$ denotes the number of attributes in A whose significances equal 0, that is, $count_{zero} = |\{a \in A : Sig(a) = 0\}|$.

From Definition 4.2, it can be seen that for any $a \in A$, $0 < weight(a) \leq 2$, and $weight(a)$ is calculated based on the significance $Sig(a)$ of a and $count_{zero}$. If $Sig(a) > 0$ then $weight(a)$ is equal to $1 + Sig(a)$, else $weight(a) = 0.5 \times (1 + count_{zero}/(|A| + \sqrt{|A| - count_{zero}}))$. It is obvious that if $Sig(a) > 0$ then $weight(a)$ is proportional to $Sig(a)$, and if $Sig(a) = 0$ then $weight(a)$ is proportional to $count_{zero}$. Especially, if $Sig(a) = 0$ and $count_{zero} = |A|$, then $weight(a) = 1$.

In Definition 4.2, if we regard the significance of a as the weight of a (i.e., let $weight(a) = Sig(a)$), then only those attributes whose significances are greater than 0 can be used to calculate the distance between two objects. In many cases, there exists a number of attributes in A whose significances equal 0, and these attributes should not be neglected when calculating the distance between two objects. Therefore, we adopt a compromise proposal, that is, assign a small weight (the weight is less than or equal to 1) to those attributes whose significances equal 0, and assign bigger weights (the weights are greater than 1) to those attributes whose significances are greater than 0.

Definition 4.3 (Weighted matching distance metric)**.** Given an information table $IS = (U, A, V, f)$, for any $a \in A$, let $weight(a)$ denote the weight of attribute a in IS. For any two objects $x, y \in U$, the weighted matching distance between x and y is defined as follows:

$$wd(x, y) = \sum_{a \in A} weight(a) \times \delta_a(x, y), \tag{5}$$

where $\delta_a(x, y)$ is given in Definition 4.1.

In this paper, during the process of initialization, we use the weighted matching distance metric to calculate the distance between two objects. However, during the process of clustering, we still use the simple matching distance metric to calculate the distance between two objects, since this paper only concentrates on the issue of initialization for K-modes clustering. In another paper, we will consider the issue of K-modes clustering based on the weighted matching distance metric.

5. The initialization algorithm *Ini_Distance* for *K*-modes clustering

In this section, we present the initialization algorithm *Ini_Distance* for K-modes clustering. In algorithm *Ini_Distance*, we use the distance-based outlier detection technique to calculate the degree of outlierness of each object, in order to avoid that outliers are selected as initial centers. However, for any object $x \in U$, the traditional distance-based outlier detection method can only give a binary classification of x, i.e., x is or is not an outlier [42]. In many cases, it is more meaningful to assign to x a degree of being an outlier. Therefore, in the following, we introduce a concept called ‘ distance outlier factor (DOF)’, which can quantify the degree of outlierness of a given object [34,35].

Definition 5.1 (Distance outlier factor)**.** Given an information table $IS = (U, A, V, f)$, for any $x \in U$, the *distance outlier factor of* object x in IS is defined as:

$$DOF(x) = \frac{|\{y \in U : wd(x, y) > dis\}|}{|U|}, \tag{6}$$

where $wd(x, y)$ denotes the weighted matching distance between objects x and y.

In Definition 5.1, *dis* is a given parameter. It is obvious that for any $x, y \in U$, $0 \leq wd(x, y) \leq \sum_{a \in A} weight(a)$, hence *dis* should be within the range of 0 to $\sum_{a \in A} weight(a)$.

In algorithm *Ini_Distance*, we not only calculate the degree of outlierness of each object, but also consider the distance between any two initial centers. Since clusters are separated groups in a feature space, it is desirable to select initial centers which are well separated. For any two initial centers c_1 and c_2, if the distance between c_1 and c_2 is too small, then they may come from the same cluster [15]. To avoid that different initial centers come from the same cluster, in algorithm *Ini_Distance*, we calculate the distances between candidate initial centers and all currently existing initial centers. For any candidate initial center *x*, if *x* is far away from every existing initial center, and the degree of outlierness of *x* is low, then the possibility of *x* being an initial center will be high.

In summary, algorithm *Ini_Distance* determines initial cluster centers based on the following two factors:

(1) the degree of outlierness of each candidate initial center;
(2) the distances between candidate initial centers and all currently existing initial centers.

In the following definition, the above two factors are used together to calculate the possibility of each candidate center being an initial center.

Definition 5.2. Given an information table $IS = (U, A, V, f)$, for any object $x \in U$, let $DOF(x)$ be the distance outlier factor of *x*. Let $C = \{c_1, c_2, \ldots, c_q\}$ denote the set of all currently existing initial centers, for any candidate center $y \in U - C$, the *possibility of y* being a distance-based initial center is defined as:

$$Pos\_DIC(y) = \frac{\sum_{j=1}^{q} wd(y, c_j)}{q} - \sqrt{DOF(y)} + \frac{\sum_{j=1}^{q} wd(y, c_j)}{q \times (1 + \sqrt{DOF(y)})}, \tag{7}$$

where $wd(y, c_j)$ denotes the weighted matching distance between *y* and c_j(as defined in Definition 4.3), $1 \leq j \leq q$, and $\sum_{j=1}^{q} wd(y, c_j)/q$ denotes the average distance between *y* and all existing initial centers in *C*.

In Definition 5.2, we call the initial centers selected by algorithm *Ini_Distance* the "distance-based initial centers". From Definition 5.2, it can be seen that if $DOF(y)$ is high, then $Pos\_DIC(y)$ will be low, and if $\sum_{j=1}^{q} wd(y, c_j)/q$ is large, then $Pos\_DIC(y)$ will be high.

Formula (7) in Definition 5.2 was obtained by the following three steps. Since $Pos\_DIC(y)$ is dependent on the two factors: $DOF(y)$ and $\sum_{j=1}^{q} wd(y, c_j)/q$, we first used the simplest case to define $Pos\_DIC(y)$, that is, $Pos\_DIC(y) = \sum_{j=1}^{q} wd(y, c_j)/q - DOF(y)$. By using the above definition, we tested the performance of algorithm *Ini_Distance* on various data sets. Second, we repeatedly adjusted the definition of Formula (7) to obtain better performance for *Ini_Distance*. For instance, let $Pos\_DIC(y) = \sum_{j=1}^{q} wd(y, c_j)/q - \sqrt{DOF(y)}$ or $Pos\_DIC(y) = \sum_{j=1}^{q} wd(y, c_j)/(q \times (1 + DOF(y)))$, etc. Finally, we obtained the final definition of Formula (7), from which *Ini_Distance* can have a good performance.

The pseudo code for algorithm *Ini_Distance* is given as follows.

Given any $B \subseteq A$, if we use the traditional method to calculate the partition $U/IND(B)$, the time complexity is $O(|U|^2)$. To reduce the time complexity for calculating $U/IND(B)$, Nguyen and Nguyen [51] proposed an algorithm to calculate $U/IND(B)$ by sorting objects from *U*, and the time complexity is $O(|B| \times |U| \times \log_2|U|)$.

In Algorithm 1, we adopt the counting sort-based method to calculate the partition $U/IND(B)$ [65], and the time complexity is $O(|B| \times |U|)$. In the worst case, the time complexity of Algorithm 1 is $O(|U|^2 \times |A|)$, and its space complexity is $O(|U| + |A|)$.

6. The initialization algorithm *Ini_Entropy* for *K*-modes clustering

The distance-based outlier detection is an effective non-parametric technique for detecting outliers, but it is not feasible for dealing with large data sets as its time complexity is usually high. For instance, the outlier detection algorithms based on nested loops typically require $O(n^2)$ distance computations [42], where *n* is the number of objects. In many application domains - such as banking, government censuses, and geographic information systems - *n* is usually too large for an algorithm with $O(n^2)$ complexity to be practical. When using the distance-based technique to detect outliers, we must select an appropriate distance metric for calculating the distance-based outliers. However, it is difficult to do so for many practical tasks. Finding suitable distance metrics may involve too many trials. Moreover, the quality of distance-based technique is highly dependent on the parameter *dis* (as defined in Definition 5.1).

In this section, to avoid the problems of distance-based technique, we first propose a partition entropy-based outlier detection technique within the framework of rough sets. By using the partition entropy-based technique, we further present the initialization algorithm *Ini_Entropy* for *K*-modes clustering.

In recent years, many researchers have introduced Shannon entropy to rough set theory [11,20,27,45,52,63]. Düntsch and Gediga [20] discussed the measurement of uncertainty in predicting based on rough sets and proposed the notion of partition entropy. Beaubouef et al. [11] discussed the information theoretic measures of uncertainty in rough set theory and rough relational databases. Wierman [63] presented the concept of granularity to measure uncertainty in rough set theory, which is closely connected with Shannon entropy. Hu et al. [27] introduced a new measure of feature quality, called rank mutual information, which combines the advantage of robustness of Shannon entropy with the ability of dominance rough sets in extracting ordinal structures from monotonic data sets.

```
Input: information table $IS = (U, A, V, f)$, where $U = \{x_1, ..., x_n\}$, $A = \{a_1, ..., a_m\}$; parameters $K$ and $dis$, where $K$ is the predefined number of clusters, and $dis$ is the threshold value for calculating the distance-based outliers.
Output: a set $C$ of initial cluster centers.
Initialization: let set $C \leftarrow \emptyset$;
calculate the partition $U/IND(A)$ based on the counting sort;
calculate the partition entropy $PE(A)$ of $U/IND(A)$;
for $j = 1$ to $m$ do
    calculate the partition $U/IND(A - \{a_j\})$ based on the counting sort;
    calculate the partition entropy $PE(A - \{a_j\})$ of $U/IND(A - \{a_j\})$;
    calculate the significance $Sig(a_j)$ of attribute $a_j$;
end
for each $1 \leq j \leq m$, calculate the weight $weight(a_j)$ of attribute $a_j$;
for $i = 1$ to $n$ do
    $count \leftarrow 0$;
    for $j = 1$ to $n$ do
        calculate the weighted matching distance $wd(x_i, x_j)$ between objects $x_i$ and $x_j$;
        if $wd(x_i, x_j) > dis$ then $count \leftarrow count + 1$;
    end
    calculate the distance outlier factor of $x_i$, i.e., $DOF(x_i) \leftarrow \frac{count}{n}$;
end
sort domain $U$ in ascending order based on the set $\{DOF(x) : x \in U\}$;
select the first object $y$ from $U$ and let $C \leftarrow C \cup \{y\}$ (i.e., if $y$ has the minimum degree of outlierness, then select $y$ as the first initial center);
while $|C| < K$ do
    for $i = 1$ to $n$ do
        if $x_i \notin C$ then
            for $j = 1$ to $|C|$ do
                calculate the weighted matching distance $wd(x_i, c_j)$ between $x_i$ and $c_j$, where $c_j \in C$ is the $j$-th currently existing initial center;
            end
            calculate the possibility $Pos\_DIC(x_i)$ of object $x_i$ being a distance-based initial center;
        end
    end
    select $z \in U - C$ such that $Pos\_DIC(z) = Max(\{Pos\_DIC(y) : y \in U - C\})$, and let $C \leftarrow C \cup \{z\}$;
end
return $C$.
```

Algorithm 1. Ini_Distance.

Moreover, Shannon entropy has also been used for finding reducts in rough sets [33,36,43,44,48,57,60,62]. Miao and Hu [48] proposed a heuristic attribute reduction algorithm by using the mutual information-based attribute significance. Liang et al. [43,44] proposed a new definition for Shannon entropy in rough sets and used the corresponding conditional entropy to reduce redundant attributes in incomplete information systems. Qian et al. [57] presented the combination entropy to measure the uncertainty of incomplete information systems and used the corresponding conditional entropy to find reducts.

As mentioned in Definition 3.3, given an information table $IS = (U, A, V, f)$, for any $a \in A$, we can measure the significance of attribute a based on the partition entropy. That is, we calculate the partition entropy of $U/IND(A)$, and observe the change of partition entropy when removing a from A. If the partition entropy of $U/IND(A - \{a\})$ is much less than that of $U/IND(A)$, then the significance of a will be large.

It can be seen that the significance of attribute a is in fact a measurement for the effect of a on the partition entropy. If the partition entropy decreases markedly when removing a from A, then $Sig(a)$ will be large, since a has a significant effect on the partition entropy. Similarly, for any $x \in U$, we may quantify the effect of object x on the partition entropy. In the following, we introduce a new concept called 'significance of object', which can measure the effect of a given object on the partition entropy.

Definition 6.1 (Significance of object)**.** Given an information table $IS = (U, A, V, f)$, for any $B \subseteq A$, let $U/IND(B) = \{X_1, \ldots, X_p\}$ be the partition of U induced by the indiscernibility relation $IND(B)$. For any $x \in U$, a new information table $IS_x = (U - [x]_B, A, V_x, f_x)$ can be generated from IS by deleting some objects (*i.e.*, all elements in $[x]_B$) from U, where $[x]_B$ is the equivalence class in $U/IND(B)$

that contains x. The significance $Sig_B(x)$ of object x with respect to relation $IND(B)$ is defined as:

$$Sig_B(x) = \begin{cases} \dfrac{PE(B) - PE_x(B)}{PE(B) \times |[x]_B|}, & \text{if } PE(B) > PE_x(B); \\ 0, & \text{otherwise}, \end{cases}$$

where $PE(B)$ denotes the partition entropy of $U/IND(B)$ in IS, and $PE_x(B)$ denotes the partition entropy of $(U - [x]_B)/IND(B)$ in IS_x (Note: if $U - [x]_B = \emptyset$, then $PE_x(B) = 0$).

In Definition 6.1, for any $x \in U$ and $B \subseteq A$, we use $Sig_B(x)$ to denote the effect of x on the partition entropy of $U/IND(B)$. Various objects in U have different effect on the partition entropy. To measure the effect of x on the partition entropy, here we adopt a method similar to that used in Definition 3.3, i.e., removing x from U and observing the change of partition entropy. Moreover, since all elements in equivalence class $[x]_B$ are indiscernible with respect to the given knowledge $IND(B)$, we may treat them as a whole. Therefore, in Definition 6.1, we delete all elements in $[x]_B$ from U and observe the change of partition entropy. If the partition entropy of $U/IND(B)$ decreases markedly, then $Sig_B(x)$ will be large, since x has a significant effect on the partition entropy. On the other hand, if the partition entropy varies slightly, then $Sig_B(x)$ will be small, since x has little effect on the partition entropy.

In summary, $Sig_B(x)$ gives a measure for the effect of x on the partition entropy of $U/IND(B)$. The higher the significance of x, the bigger the effect of x on the partition entropy of $U/IND(B)$.

Theorem 6.1. *Given an information table $IS = (U, A, V, f)$, for any $B \subseteq A$, let $U/IND(B) = \{X_1, \ldots, X_p\}$ be the partition of U induced by the indiscernibility relation $IND(B)$ in IS. For any $x \in U$, construct a new information table $IS_x = (U - [x]_B, A, V_x, f_x)$ from IS by removing all elements in $[x]_B$ from U, and let $(U - [x]_B)/IND(B) = \{X'_1, \ldots, X'_{p-1}\}$ be the partition of $U - [x]_B$ induced by $IND(B)$ in IS_x. We have the following results*:

(1) $0 \le Sig_B(x) \le 1$, *where $Sig_B(x)$ denotes the significance of object x with respect to relation $IND(B)$*;

(2) $PE_x(B) = \log_2(|U| - |[x]_B|) - \frac{\sum_{i=1}^{p} |X_i| \log_2 |X_i| - |[x]_B| \log_2 |[x]_B|}{|U| - |[x]_B|}$, *where $PE_x(B)$ denotes the partition entropy of $(U - [x]_B)/IND(B)$ in IS_x.*

Proof.

(1) From Definitions 3.2 and 6.1, it is easy to prove that $0 \le Sig_B(x) \le 1$, and hence we omit the proof of (1);

(2) Since $(U - [x]_B)/IND(B) = \{X'_1, \ldots, X'_{p-1}\}$, from Definition 3.2, we can obtain that

$$\begin{aligned} PE_x(B) &= -\sum_{i=1}^{p-1} \frac{|X'_i|}{|U| - |[x]_B|} \log_2 \frac{|X'_i|}{|U| - |[x]_B|} \\ &= \log_2(|U| - |[x]_B|) - \sum_{i=1}^{p-1} \frac{|X'_i|}{|U| - |[x]_B|} \log_2 |X'_i| \\ &= \log_2(|U| - |[x]_B|) - \frac{\sum_{i=1}^{p-1} |X'_i| \log_2 |X'_i|}{|U| - |[x]_B|}. \end{aligned}$$

Moreover, since $(U - [x]_B)/IND(B) = U/IND(B) - \{[x]_B\} = \{X_1, \ldots, X_p\} - \{[x]_B\}$, we have that $\sum_{i=1}^{p-1} |X'_i| \log_2 |X'_i| = \sum_{i=1}^{p} |X_i| \log_2 |X_i| - |[x]_B| \log_2 |[x]_B|$. Hence, $PE_x(B) = \log_2(|U| - |[x]_B|) - \frac{\sum_{i=1}^{p} |X_i| \log_2 |X_i| - |[x]_B| \log_2 |[x]_B|}{|U| - |[x]_B|}$.
This completes the proof of (2). □

Based on (2) of Theorem 6.1, we can obtain an efficient method for calculating the partition entropy $PE_x(B)$ of $(U - [x]_B)/IND(B)$ in IS_x. To calculate $PE_x(B)$, the method does not use Eq. (2) in Definition 3.2. That is, we do not need to calculate $\frac{|X'|}{|U| - |[x]_B|} \log_2 \frac{|X'|}{|U| - |[x]_B|}$ for each $X' \in (U - [x]_B)/IND(B)$, which can effectively reduce the time complexity of algorithm *Ini_Entropy*. Instead, the method first calculates $\sum_{i=1}^{p} |X_i| \log_2 |X_i|$ based on the partition $U/IND(B)$ of U in IS. Then, for any $x \in U$, the partition entropy $PE_x(B)$ can be obtained from (2) of Theorem 6.1.

In the following, for a given information table IS, we define two kinds of sequences in IS: the weight-based sequence of attributes and the weight-based sequence of attribute subsets.

Definition 6.2 (Weight-based sequence of attributes)**.** Given an information table $IS = (U, A, V, f)$, where $A = \{a_1, \ldots, a_m\}$, for each $1 \le j \le m$, let $weight(a_j)$ denote the weight of attribute a_j (as defined in Definition 4.2). Based on the set $\{weight(a_j) : 1 \le j \le m\}$, we can obtain a *weight-based sequence* $S = \langle a'_1, \ldots, a'_m \rangle$ *of attributes* in IS, such that for any $1 \le j \le m$, $a'_j \in A$, and for any $1 \le j < m$, $weight(a'_j) \le weight(a'_{j+1})$.

Definition 6.3 (Weight-based sequence of attribute subsets)**.** Given an information table $IS = (U, A, V, f)$, where $A = \{a_1, \ldots, a_m\}$, let $S = \langle a'_1, \ldots, a'_m \rangle$ be the weight-based sequence of attributes in IS. Assume that $AS = \langle A_1, \ldots, A_m \rangle$ is a sequence of attribute subsets in IS, where for each $1 \le j \le m$, $A_j \subseteq A$. If $A_1 = A$, $A_m = \{a'_m\}$ and $A_{j+1} = A_j - \{a'_j\}$ for each $1 \le j < m$, then we call AS a *weight-based sequence of attribute subsets* in IS.

From Definition 6.3, it can be seen that given a weight-based sequence $AS = \langle A_1, \ldots, A_m \rangle$ of attribute subsets, for each $1 \leq j < m$, A_{j+1} is obtained from A_j by deleting the attribute a'_j from A_j, where a'_j is the jth attribute in the sequence S. Therefore, for a given information table IS, both the weight-based sequence of attributes in IS and the weight-based sequence of attribute subsets in IS are unique.

To quantify the degree of outlierness of a given object, we introduce a concept called 'partition entropy-based outlier factor', which is similar to the distance outlier factor given in Section 5.

Definition 6.4 (Partition entropy-based outlier factor)**.** Given an information table $IS = (U, A, V, f)$, where $A = \{a_1, \ldots, a_m\}$. Let $AS = \langle A_1, \ldots, A_m \rangle$ be the weight-based sequence of attribute subsets in IS, for any object $x \in U$, the *partition entropy-based outlier factor PEOF*(x) of x in IS is defined as follows:

$$PEOF(x) = \frac{\sum_{j=1}^{m} weight(a_j) \sum_{y \in [x]_{\{a_j\}}} Sig_{\{a_j\}}(y) + \sum_{j=1}^{m} W(A_j) \sum_{y \in [x]_{A_j}} Sig_{A_j}(y)}{2 \times m}, \tag{8}$$

where for each $1 \leq j \leq m$, $Sig_{\{a_j\}}(y)$ and $Sig_{A_j}(y)$ respectively denote the significances of object y with respect to relations $IND(\{a_j\})$ and $IND(A_j)$. Moreover, for any $1 \leq j \leq m$, $weight(a_j)$ denotes the weight of attribute a_j, and $W(A_j) = \frac{\sum_{a \in A_j} weight(a)}{|A_j|}$ denotes the weight of attribute subset A_j (*i.e.*, the average weight of all attributes in A_j).

Given any $x \in U$ and $B \subseteq A$, from Definition 6.1, we can obtain that for any $y \in [x]_B$, $Sig_B(y) = Sig_B(x)$. Hence, we can rewrite Eq. (8) as follows:

$$PEOF(x) = \frac{\sum_{j=1}^{m} (weight(a_j) \times |[x]_{\{a_j\}}| \times Sig_{\{a_j\}}(x) + W(A_j) \times |[x]_{A_j}| \times Sig_{A_j}(x))}{2 \times m}. \tag{9}$$

In Definition 6.4, for any $x \in U$, the significances of object x with respect to different indiscernibility relations are used to calculate $PEOF(x)$. From Eq. (9), it can be seen that $PEOF(x)$ is proportional to the significances of x, that is, those objects with larger significances are more likely to be outliers.

The reason why we use the significance of x to calculate $PEOF(x)$ is as follows. In nearly all its meanings, entropy can be viewed as a measure of disorder or disorganization in a system. Therefore, given an information table $IS = (U, A, V, f)$ and any $B \subseteq A$, we can use the partition entropy of $U/IND(B)$ to quantify the disorder of domain U with respect to the given knowledge $IND(B)$. Compared with a normal object (denoted by x_1) in U, an outlier (denoted by x_2) in U usually makes a greater contribution to the disorder of U. In other words, if we delete x_2 from U, then the disorder of U may markedly decrease. On the contrary, if we delete x_1 from U, then the disorder of U may vary slightly.

Based on the above observation, it can be seen that for any $x \in U$, the degree of outlierness of object x is related to the effect of x on the partition entropy (i.e., the disorder of U). That is, if x has a significant effect on the partition entropy, then the degree of outlierness of x will be high. In Definition 6.1, the significance of x is used to measure the effect of x on the partition entropy. Therefore, the significance of x can be further used to calculate the degree of outlierness of x. Given a set $\{IND(B_i) \colon B_i \subseteq A, 1 \leq i \leq s\}$ of indiscernibility relations on U, if for each $1 \leq i \leq s$, the significance of x with respect to relation $IND(B_i)$ (i.e., the effect of x on the partition entropy of $U/IND(B_i)$) is always disproportionately large compared with other objects in U, then we may consider x as behaving abnormally and $PEOF(x)$ will be high.

Given an information table $IS = (U, A, V, f)$, each subset B of A determines an indiscernibility relation $IND(B)$ on U, and so we would have $2^{|A|}$ relations on U. For any $x \in U$, to obtain $PEOF(x)$, it is impracticable to calculate the significances of x with respect to all these relations, because the time complexity will be exponential in $|A|$. Therefore, to reduce the time complexity of outlier detection, a compromise solution is proposed in Definition 6.4. That is, we just calculate the significances of x with respect to the indiscernibility relations induced by each singleton subset $\{a_j\}$ of A and the relations induced by each subset A_j in sequence AS, where $1 \leq j \leq |A|$. Finally, we combine the results from the two sources to calculate $PEOF(x)$.

By using the partition entropy-based outlier detection technique, we can design another initialization algorithm for K-modes clustering, i.e., algorithm *Ini_Entropy*. In *Ini_Entropy*, for any candidate center y, to calculate the possibility of y being an initial center, we also employ the two factors used in algorithm *Ini_Distance*, i.e., (1) the degree of outlierness of y; (2) the distance between y and each currently existing initial center.

Definition 6.5. Given an information table $IS = (U, A, V, f)$, for any $x \in U$, let $PEOF(x)$ be the partition entropy-based outlier factor of object x. Let $C = \{c_1, c_2, \ldots, c_q\}$ denote the set of all currently existing initial centers, for any candidate center $y \in U - C$, the *possibility of* y being a partition entropy-based initial center is defined as follows.

$$Pos\_PEIC(y) = \frac{\sum_{j=1}^{q} wd(y, c_j)}{q} - \sqrt{PEOF(y)} + \frac{\sum_{j=1}^{q} wd(y, c_j)}{q \times (1 + \sqrt{PEOF(y)})}, \tag{10}$$

where $wd(y, c_j)$ denotes the weighted matching distance between y and c_j (as defined in Definition 4.3), $1 \leq j \leq q$.

In Definition 6.5, for clarity, we call the initial centers selected by algorithm *Ini_Entropy* the 'partition entropy-based initial centers'.

The pseudo code for algorithm *Ini_Entropy* is given as follows.

```
Input: information table $IS = (U, A, V, f)$, where $U = \{x_1, ..., x_n\}$, $A =$
       $\{a_1, ..., a_m\}$; parameters $K$, where $K$ is the predefined number of
       clusters.
Output: a set $C$ of initial cluster centers.
Initialization: let set $C \leftarrow \emptyset$;
by using Steps 1-8 of Algorithm 1, calculate the weight of attribute $a_i$,
$1 \leq i \leq m$;
construct the weight-based sequence $S = \langle a'_1, ..., a'_m \rangle$ of attributes and the
weight-based sequence $AS = \langle A_1, ..., A_m \rangle$ of attribute subsets in $IS$;
for $j = 1$ to $m$ do
    calculate the weight $W(A_j)$ of attribute subset $A_j$;
    calculate the partitions $U/IND(\{a_j\})$ and $U/IND(A_j)$;
    calculate the partition entropy $PE(\{a_j\})$ of $U/IND(\{a_j\})$ and the
    partition entropy $PE(A_j)$ of $U/IND(A_j)$;
    for $i = 1$ to $n$ do
        by using Theorem 6.1, calculate the partition entropy $PE_{x_i}(\{a_j\})$ of
        partition $(U - [x_i]_{\{a_j\}})/IND(\{a_j\})$ and the partition entropy
        $PE_{x_i}(A_j)$ of partition $(U - [x_i]_{A_j})/IND(A_j)$;
        calculate the significances $Sig_{\{a_j\}}(x_i)$ and $Sig_{A_j}(x_i)$ of object $x_i$;
    end
end
for any $x \in U$, calculate the outlier factor $PEOF(x)$ of object $x$;
sort domain $U$ in ascending order based on the set $\{PEOF(x) : x \in U\}$;
select the first object $y$ from $U$ and let $C \leftarrow C \cup \{y\}$ (i.e., if $y$ has the
minimum degree of outlierness, then select $y$ as the first initial center);
while $|C| < K$ do
    for $i = 1$ to $n$ do
        if $x_i \notin C$ then
            for $j = 1$ to $|C|$ do
                calculate the weighted matching distance $wd(x_i, c_j)$ between
                $x_i$ and $c_j$, where $c_j \in C$ is the $j$-th currently existing initial
                center;
            end
            calculate the possibility $Pos\_PEIC(x_i)$ of object $x_i$ being a
            partition entropy-based initial center;
        end
    end
    select $z \in U - C$ such that $Pos\_PEIC(z) = Max(\{Pos\_PEIC(y) :$
    $y \in U - C\})$, and let $C \leftarrow C \cup \{z\}$;
end
return $C$.
```

Algorithm 2. Ini_Entropy.

In Algorithm 2 we also use the counting sort-based method to calculate the partition $U/IND(B)$ of U induced by relation $IND(B)$ [65]. In the worst case, the time complexity of Algorithm 2 is $O(|A| \times |U| \times (K + |A|))$, and its space complexity is $O(|U| \times (K + |A|))$. Since the time complexity of Algorithm 2 is linear in the number of objects, the algorithm is able to achieve good scalability for large data sets.

7. Experimental analysis

To evaluate the performances of algorithms *Ini_Distance* and *Ini_Entropy*, we compared the results of the two algorithms with those of four other initialization methods on six categorical data sets.

7.1. The experimental setup

To test *Ini_Distance* and *Ini_Entropy*, the following six categorical data sets were used

(1) Soybean (Small) data set (denoted by Soybean),
(2) Zoo data set (denoted by Zoo),
(3) Breast Cancer Wisconsin (Original) data set (denoted by Breast),
(4) Mushroom data set (denoted by Mushroom),
(5) Lung Cancer data set (denoted by Lung), and
(6) Congressional Voting Records data set (denoted by Voting).

Table 1
Properties of the six UCI data sets.

Properties	Soybean	Zoo	Breast	Mushroom	Lung	Voting
Number of classes	4	7	2	2	3	2
Number of instances	47	101	699	8124	32	435
Number of attributes	35	16	9	22	56	16
Has missing values	no	no	yes	yes	yes	yes

The above data sets were all obtained from the UCI Machine Learning Repository [5]. A detailed description of the six data sets is given in Table 1.

It should be noted that there are 14 special attributes in the Soybean data set. For each of the 14 attributes, the number of distinct attribute values equals 1. Therefore, in the experimentation we removed the 14 attributes from the Soybean data set and the final data set contains 21 attributes. Moreover, for the missing attribute values in the Breast, Mushroom, Lung and Voting data sets, we treated them as special values [5].

In the experimentation, algorithms *Ini_Distance* and *Ini_Entropy* were compared with the following four initialization methods for *K*-modes clustering:

(1) Khan’s initialization method [40],
(2) Wu’s initialization method [64],
(3) Cao’s initialization method [15], and
(4) the random initialization method.

In the experimentation, for each of the six data sets given in Table 1, we first used *Ini_Distance* and *Ini_Entropy* to select *K* initial cluster centers. Second, the four initialization methods listed above were also used to select *K* initial cluster centers. Third, for each set of initial centers generated by a specific initialization method, we obtained the corresponding clustering result by employing the standard *K*-modes algorithm proposed by Huang [29]. Finally, the *K*-modes clustering results derived from the six initialization methods were compared with each other. It should be noted that the weighted matching distance metric (given in Definition 4.3) was only used in the process of initialization. During the process of clustering, we still used the standard *K*-modes algorithm proposed by Huang, that is, we used the simple matching distance metric to calculate the distance between two objects in the process of clustering. Since all initialization methods used the same clustering algorithm in the process of clustering, the differences between different clustering results are only determined by the differences between different initialization methods.

We implemented algorithms *Ini_Distance* and *Ini_Entropy* in Java. Experiments were conducted on a 2.13 GHz Intel Pentium P6200 machine with 2 GB RAM, running the Windows XP operating system.

For each initialization method *M*, we evaluated the effectiveness of *M* in terms of the quality of the *K*-modes clustering results derived from *M*. To evaluate the quality of clustering results, we adopted the performance metrics used by Wu et al. [64], which were derived from the information retrieval community.

For a given data set *DT* and clustering algorithm *A*, we may assume that *DT* contains *K* classes (denoted by $C_1, \ldots, C_K$), and the algorithm *A* partitions *DT* into *K* clusters (denoted by $C'_1, \ldots, C'_K$). For each $1 \leq i \leq K$, let p_i denote the number of objects that are correctly assigned to class C_i (i.e., $p_i = |C_i \cap C'_i|$); q_i denote the number of objects that are incorrectly assigned to class C_i (i.e., $q_i = |C'_i| - p_i$); and r_i denote the number of objects that are incorrectly rejected from class C_i (i.e., $r_i = |C_i| - p_i$). The quality of clustering results generated by algorithm *A* was evaluated by the following three metrics: precision(*PR*), recall(*RE*) and accuracy(*AC*) [64], where

$$PR = \frac{\sum_{i=1}^{K} \frac{p_i}{p_i+q_i}}{K}; \quad RE = \frac{\sum_{i=1}^{K} \frac{p_i}{p_i+r_i}}{K}; \quad AC = \frac{\sum_{i=1}^{K} p_i}{|DT|}. \tag{11}$$

When using algorithm *Ini_Distance* to initialize the cluster centers of *K*-modes clustering, the value of parameter *dis* must be determined. In the experimentation, the value of *dis* was obtained by the following steps. First, we assigned an original empirical value to *dis*, and tested the performance of *Ini_Distance* on each data set. Second, we repeatedly adjusted the value of *dis* to obtain better performance for *Ini_Distance*. Finally, we adopted the best value for parameter *dis*.

7.2. Experimental results on the six UCI data sets

Tables 2–7 respectively show the *K*-modes clustering results (i.e., the three performance metrics: *PR*, *RE* and *AC*, as defined in Eq. (11)) on the Soybean, Zoo, Breast, Mushroom, Lung and Voting data sets. Moreover, to further illustrate the clustering results derived from the initialization algorithms *Ini_Distance* and *Ini_Entropy*, we also list the confusion matrices of the clustering results derived from *Ini_Distance* and *Ini_Entropy*.

In Tables 2–7, ‘*AC*’, ‘*PR*’, and ‘*RE*’ respectively denote the three performance metrics: accuracy, precision, and recall. Moreover, ‘Random’, ‘ Khan’, ‘Wu’, ‘Cao’, ‘*Distance*’, and ‘*Entropy*’ respectively denote the six initialization methods: the random method, Khan’s method [40], Wu’s method [64], Cao’s method [15], *Ini_Distance*, and *Ini_Entropy*, where results for the random method are taken from [40].

Table 2
Clustering results on the Soybean data set.

K-modes clustering	Various initialization methods					
	Random	Khan	Wu	Cao	*Distance*	*Entropy*
AC	0.8644	0.9574	**1**	**1**	**1**	**1**
PR	0.8999	0.9583	**1**	**1**	**1**	**1**
RE	0.8342	0.9705	**1**	**1**	**1**	**1**
	Confusion matrix derived from *Ini_Distance*					
Class		*D*1	*D*2	*D*3	*D*4	
*D*1		10	0	0	0	
*D*2		0	10	0	0	
*D*3		0	0	10	0	
*D*4		0	0	0	17	
	Confusion matrix derived from *Ini_Entropy*					
Class		*D*1	*D*2	*D*3	*D*4	
*D*1		10	0	0	0	
*D*2		0	10	0	0	
*D*3		0	0	10	0	
*D*4		0	0	0	17	

Table 3
Clustering results on the Zoo data set.

K-modes clustering	Various initialization methods						
		Random	Khan	Wu	Cao	*Distance*	*Entropy*
AC		0.8356	0.8911	0.8812	0.8812	0.8911	**0.9010**
PR		0.8072	0.7224	0.8702	0.8702	0.7695	**0.8906**
RE		0.6012	0.7716	0.6714	0.6714	0.8146	**0.8432**
	Confusion matrix derived from *Ini_Distance*						
Class	*a*	*b*	*c*	*d*	*e*	*f*	*g*
a	37	0	4	0	0	0	0
b	0	20	0	0	0	0	0
c	1	0	0	3	1	0	0
d	0	0	0	13	0	0	0
e	0	0	0	0	4	0	0
f	0	0	0	0	0	8	0
g	0	0	0	0	0	2	8
	Confusion matrix derived from *Ini_Entropy*						
Class	*a*	*b*	*c*	*d*	*e*	*f*	*g*
a	37	2	0	2	0	0	0
b	0	20	0	0	0	0	0
c	0	1	1	2	1	0	0
d	0	0	0	13	0	0	0
e	0	0	0	0	4	0	0
f	0	0	0	0	0	8	0
g	0	0	0	0	0	2	8

From Tables 2–7, it can be seen that algorithms *Ini_Distance* and *Ini_Entropy* perform better than the random method. First, for each of the six data sets listed in Table 1, the *AC*, *PR* and *RE* values of *Ini_Entropy* are always higher than those of the random method. Second, for each of the following five data sets: Soybean, Breast, Mushroom, Lung, and Voting, the *AC*, *PR* and *RE* values of *Ini_Distance* are higher than those of the random method. For the Zoo data set, although the *PR* value of *Ini_Distance* is lower than that of the random method, the *AC* and *RE* values of *Ini_Distance* are obviously higher than those of the random method. Therefore, the overall performance of *Ini_Distance* is better than that of the random method.

Compared with the other three initialization methods (i.e., Khan's method, Wu's method and Cao's method), *Ini_Distance* and *Ini_Entropy* algorithms also have better performance. For each of the six data sets listed in Table 1, the *AC*, *PR* and *RE* values of our algorithms are higher than or equal to those of the three methods, except for the following two cases: (1) for the Zoo data set, the *PR* value of *Ini_Distance* is lower than those of Wu's method and Cao's method; (2) for the Breast data set, the *PR* value of *Ini_Distance* is slightly lower than that of Khan's method.

Table 4
Clustering results on the Breast data set.

K-modes clustering	Various initialization methods					
	Random	Khan	Wu	Cao	*Distance*	*Entropy*
AC	0.8364	0.9127	0.9113	0.9113	0.9242	**0.9328**
PR	0.8699	0.9318	0.9292	0.9292	0.9309	**0.9424**
RE	0.7743	0.8783	0.8773	0.8773	0.9009	**0.9094**
	Confusion matrix derived from *Ini_Distance*					
Class			*Benign*	*Malignant*		
Benign			447	11		
Malignant			42	199		
	Confusion matrix derived from *Ini_Entropy*					
Class			*Benign*	*Malignant*		
Benign			451	7		
Malignant			40	201		

Table 5
Clustering results on the Mushroom data set.

K-modes clustering	Various initialization methods					
	Random	Khan	Wu	Cao	*Distance*	*Entropy*
AC	0.7231	0.8815	0.8754	0.8754	**0.8941**	0.8876
PR	0.7614	0.8975	0.9019	0.9019	**0.9138**	0.9095
RE	0.7174	0.8780	0.8709	0.8709	**0.8903**	0.8835
	Confusion matrix derived from *Ini_Distance*					
Class			*Poisonous*	*Edible*		
Poisonous			3070	846		
Edible			14	4194		
	Confusion matrix derived from *Ini_Entropy*					
Class			*Poisonous*	*Edible*		
Poisonous			3016	900		
Edible			13	4195		

From Tables 2–7, we can also see that the *RE* values of *Ini_Distance* and *Ini_Entropy* are always higher than or equal to those of other methods, which indicates that our algorithms can tightly control the data objects from given classes to be not assigned to incorrect groups. Moreover, the *AC* values of our algorithms are also higher than or equal to those of other methods. However, in some cases the *PR* value of *Ini_Distance* is lower than those of other methods. For instance, the *PR* value of *Ini_Distance* on the Zoo data set is lower than those of the random method, Wu's method and Cao's method.

The cause of the above problem is given as follows. From Eq. (11), it can be seen that *PR* and *AC* are two different performance metrics, where *PR* denotes the average value of the correctness ratio of each cluster, and *AC* denotes the ratio of all data objects that are correctly assigned to the corresponding classes. As shown in Table 3, within the confusion matrix derived from *Ini_Distance*, there is a cluster (i.e., '*c*') whose correctness ratio equals 0, that is, there does not exist an object in Zoo correctly assigned to the cluster, thus the average correctness ratio (i.e., *PR* value) of *Ini_Distance* is lower than those of the random method, Wu's method and Cao's method. Although the correctness ratio of that cluster equals 0, the number of objects in Zoo belonging to that cluster is very small, and hence the *AC* value of *Ini_Distance* is still higher than those of the three methods.

When comparing the two initialization algorithms proposed in this paper, we may find that the performance of *Ini_Entropy* algorithm is relatively better. For each of the following five data sets: Soybean, Zoo, Breast, Lung, and Voting, the *AC*, *PR* and *RE* values of *Ini_Entropy* are higher than or equal to those of *Ini_Distance*. Moreover, for the Mushroom data set, the *AC*, *PR* and *RE* values of *Ini_Entropy* are slightly lower than those of *Ini_Distance*. Therefore, the overall performance of *Ini_Entropy* is better than that of *Ini_Distance*.

7.3. Scalability analysis

So far, we have illustrated the performance of *Ini_Distance* and *Ini_Entropy* algorithms, it is still interesting to know the scalability of the two algorithms. We used the following experiments to test the scalability of *Ini_Distance* and *Ini_Entropy* with increasing numbers of objects and dimensions.

Table 6
Clustering results on the Lung data set.

K-modes clustering	Various initialization methods					
	Random	Khan	Wu	Cao	*Distance*	*Entropy*
AC	0.5210	0.5000	0.5000	0.5000	0.5313	**0.6250**
PR	0.5766	0.6444	0.5584	0.5584	0.6569	**0.6833**
RE	0.5123	0.5168	0.5014	0.5014	0.5274	**0.5932**
	Confusion matrix derived from *Ini_Distance*					
Class		*a*	*b*	*c*		
a		6	3	0		
b		5	8	0		
c		1	6	3		
	Confusion matrix derived from *Ini_Entropy*					
Class		*a*	*b*	*c*		
a		3	6	0		
b		2	11	0		
c		1	3	6		

Table 7
Clustering results on the Voting data set.

K-modes clustering	Various initialization methods					
	Random	Khan	Wu	Cao	*Distance*	*Entropy*
AC	0.4972	0.8506	0.8621	0.8621	**0.8690**	**0.8690**
PR	0.5030	0.8484	0.8571	0.8571	**0.8630**	**0.8630**
RE	0.5031	0.8672	0.8755	0.8755	**0.8811**	**0.8811**
	Confusion matrix derived from *Ini_Distance*					
Class		*Republican*			*Democrat*	
Republican		157			11	
Democrat		46			221	
	Confusion matrix derived from *Ini_Entropy*					
Class		*Republican*			*Democrat*	
Republican		157			11	
Democrat		46			221	

Table 8
Data distributions of 10%KDD and Final_10%KDD.

Data set	Probe	DoS	U2R	R2L	Normal	Total
10%KDD	4107	391458	52	1126	97278	494021
Final_10%KDD	2129	54572	52	999	87832	145584

Since the six data sets listed in Table 1 are very small, here we used a new data set — the KDD-99 data set, which is a common benchmark for evaluation of intrusion detection techniques [5]. Each record in the data set is described by a set of 41 attributes and labeled as either normal, or as an attack. All attacks fall into four main categories: Probe, DoS, U2R and R2L [5].

In the experimentation, we adopted a subset of the KDD-99 data set to evaluate the scalability of algorithms *Ini_Distance* and *Ini_Entropy*. The subset is usually known as '10%KDD', which contains 494021 records. Moreover, since there exist many duplicate records (that is, more than one record has the same value on each attribute) in 10%KDD, we removed duplicate records from it. The final 10%KDD data set (denoted by Final_10%KDD) contains 145584 records. The distributions of normal records and four attack categories in 10%KDD and Final_10%KDD are summarized in Table 8 [5].

It should be noted that there are a number of continuous attributes in Final_ 10%KDD. To discretize these attributes, we used the 'equal width intervals' discretization algorithm [17], where the number of intervals was set to 5.

Finally, we evaluated the scalability of *Ini_Distance* and *Ini_Entropy* on the discretized Final_10%KDD data set. When measuring scalability with increasing number of objects, the number of attributes was set to 41 and the number of objects is in the range of 10000 to 145584. When measuring scalability with increasing number of dimensions, the number of objects was set to 145584 and the number of attributes is in the range of 5 to 41.

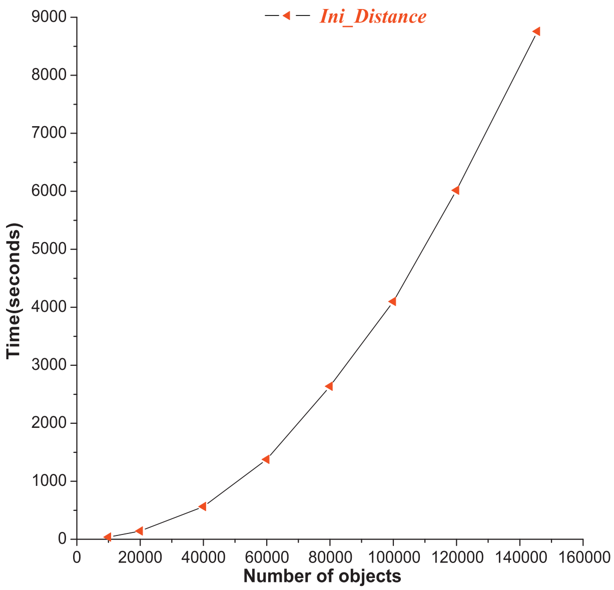

Fig. 1. Scalability of *Ini_Distance* with the number of objects (41 attributes).

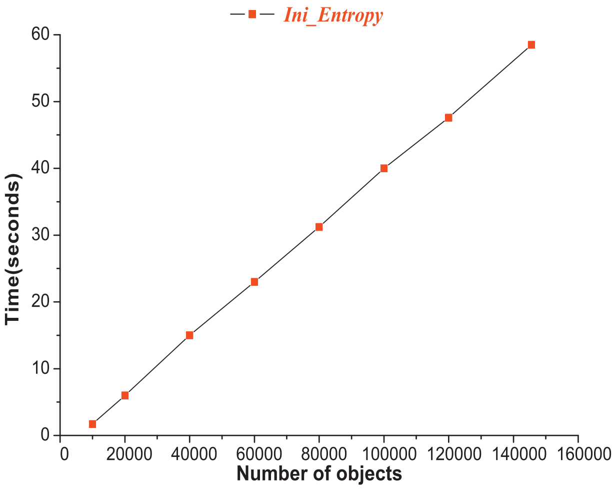

Fig. 2. Scalability of *Ini_Entropy* with the number of objects (41 attributes).

Figs. 1 and 2 respectively show the execution time of *Ini_Distance* and *Ini_Entropy* with increasing number of objects, where each algorithm was run 10 times and the mean time was calculated.

Moreover, Figs. 3 and 4 respectively show the execution time of *Ini_Distance* and *Ini_Entropy* with increasing number of dimensions.

From Figs. 1–4, it can be seen that *Ini_Entropy* algorithm takes much less running time than *Ini_Distance* algorithm. For the same numbers of objects and dimensions, the execution time of *Ini_Entropy* is much lower than that of *Ini_Distance*. Therefore, compared with *Ini_Distance*, *Ini_Entropy* can achieve better scalability. In particular, the results illustrated in Fig. 2 are consistent with our observations in Section 6 about the scalability of *Ini_Entropy* with the number of objects. That is, *Ini_Entropy* is able to achieve good scalability for large data sets, since the time cost of it grows linearly with the increase in the number of objects.

Table 9 compares the time complexities of *Ini_Distance* and *Ini_Entropy* with the other three methods, that is, Khan's method, Wu's method and Cao's method.

In Table 9, m and n respectively denote the number of attributes and the number of objects; K and r respectively denote the number of clusters and the number of iterations needed for convergence. In most cases, $m \ll n$ and $K \ll n$, hence, it can be seen that the time complexity of *Ini_Entropy* is similar to Cao's method, and lower than Khan's method and Wu's method. Moreover, the time complexity of *Ini_Distance* is the highest.

Table 10 details the running time of various initialization methods on the whole Final_10%KDD data set, which contains 145584 records and 41 attributes.

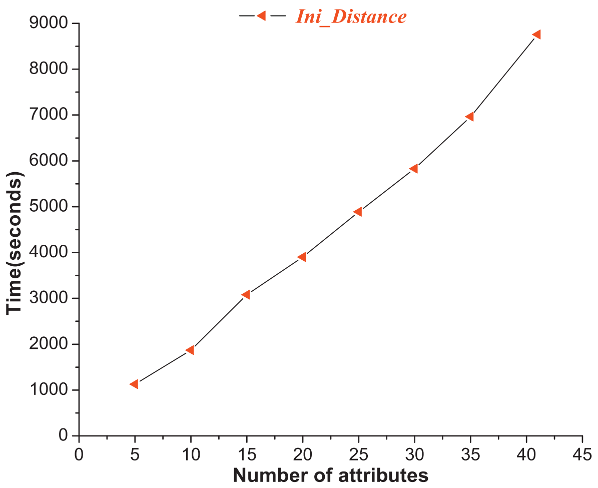

Fig. 3. Scalability of *Ini_Distance* with the number of attributes (145584 objects).

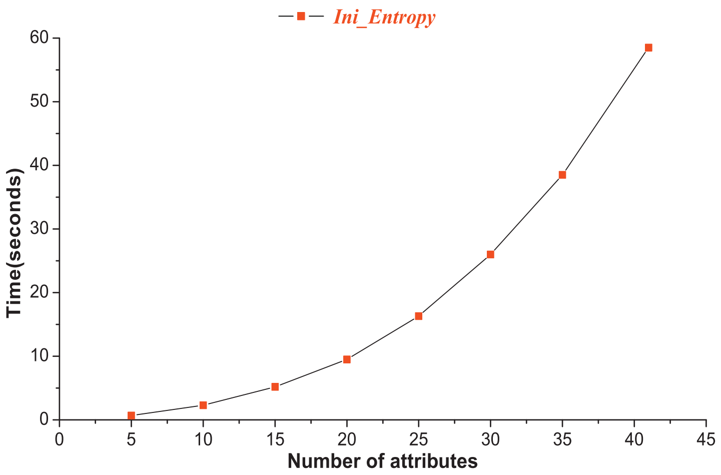

Fig. 4. Scalability of *Ini_Entropy* with the number of attributes (145584 objects).

Table 9
Time complexities of various initialization methods.

Method	Time complexity
Khan's Method	$O(nm + rKm^2n + nlogn)$, for all/prominent attributes
Wu's Method	$O(c \times n)$, where c can be between 2 to $n^{0.5}$
Cao's Method	$O(n \times m \times K^2)$
Ini_Distance	$O(m \times n^2)$
Ini_Entropy	$O(Kmn + m^2n)$

Table 10
The running time of various methods on Final_10%KDD.

Initialization methods	Khan's method	Wu's method	Cao's method	*Ini_Distance*	*Ini_Entropy*
Time (second)	1750.7	107.3	63.8	8755.1	58.5

8. Conclusions

The selection of initial cluster centers is very important for K-modes clustering, since improper initial centers may result in undesirable cluster structures. In this paper, we combined the selection of initial cluster centers in K-modes clustering with the detection of outliers. By virtue of the distance-based and partition entropy-based outlier detection techniques, we presented two initialization algorithms for K-modes clustering. For a given object x, our algorithms calculate the possibility of x being an initial center based on the following two factors: (1) the degree of outlierness of *x*; (2) the distance between *x* and each existing initial center, where the first factor can avoid that outliers are selected as initial centers, and the second factor can avoid that various initial centers come from the same cluster. We tested the proposed algorithms by using several categorical data sets, and the experimental results demonstrated the effectiveness of our algorithms.

Besides the distance-based and partition entropy-based outlier detection techniques, there are many other outlier detection techniques reported in the literature [2–4,13,19,24,34,35], for instance, the density-based method [13], the deviation-based method [4], the granular computing-based method [19,56], etc. In the future work, we plan to use other outlier detection techniques to select initial centers for *K*-modes clustering. Moreover, we may apply *Ini_Distance* and *Ini_Entropy* to the *K*-modes-variant algorithms, which include fuzzy *K*-mode [30], fuzzy *K*-mode with fuzzy centroid [41], and *K*-prototype algorithms [29,46].

Acknowledgments

This work is supported by the National Natural Science Foundation of China (grant nos. 61273180, 60802042), the Natural Science Foundation of Shandong Province, China (grant nos. ZR2011FQ005, ZR2014FM015), and the Project of Shandong Province Higher Educational Science and Technology Program (grant no. J11LG05).

References

[1] M.R. Anderberg, Cluster Analysis for Applications, Academic Press, New York, 1973.

[2] F. Angiulli, R. Ben-Eliyahu-Zohary, L. Palopoli, Outlier Detection for Simple Default Theories, Artificial Intelligence 174 (15) (2010) 1247–1253.

[3] F. Angiulli, S. Basta, S. Lodi, C. Sartori, Distributed Strategies for Mining Outliers in Large Data Sets, IEEE Transactions on Knowledge and Data Engineering 25 (7) (2013) 1520–1532.

[4] A. Arning, R. Agrawal, P. Raghavan, A linear method for deviation detection in large databases, in: Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96), Oregon, USA, 1996, pp. 164–169.

[5] K. Bache, M. Lichman, UCI machine learning repository, 2013, http://archive.ics.uci.edu/ml.

[6] L. Bai, J.Y. Liang, C.Y. Dang, An initialization method to simultaneously find initial cluster centers and the number of clusters for clustering categorical data, Knowledge-Based Systems 24 (6) (2011) 785–795.

[7] L. Bai, J.Y. Liang, C.Y. Dang, F.Y. Cao, A novel attribute weighting algorithm for clustering high-dimensional categorical data, Pattern Recognition 44 (12) (2011) 2843–2861.

[8] T. Bai, C.A. Kulikowski, L.G. Gong, B. Yang, L. Huang, C.G. Zhou, A Global *K*-modes Algorithm for Clustering Categorical Data, Chinese Journal of Electronics 21 (3) (2012) 460–465.

[9] L. Bai, J.Y. Liang, C. Sui, C.Y. Dang, Fast global *K*-means clustering based on local geometrical information, Information Sciences 245 (2013) 168–180.

[10] D. Barbara, J. Couto, Y. Li, COOLCAT: An entropy-based algorithm for categorical clustering, in: Proc. of the Eleventh Int. Conf. on Information and Knowledge Management, 2002, pp. 582–589.

[11] T. Beaubouef, F.E. Petry, G. Arora, Information-theoretic measures of uncertainty for rough sets and rough relational databases, Information Sciences 109 (1998) 535–563.

[12] P.S. Bradley, U.M. Fayyad, Refining initial points for *K*-means clustering, in: Proc. of the 15th Int. Conf. on Machine Learning, Morgan Kaufmann, 1998, pp. 91–99.

[13] M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, LOF: Identifying density-based local outliers, in: Proc. of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, USA, 2000, pp. 93–104.

[14] F.Y. Cao, J.Y. Liang, G. Jiang, An initialization method for the *K*-Means algorithm using neighborhood model, Computers and Mathematics with Applications 58 (3) (2009) 474–483.

[15] F.Y. Cao, J.Y. Liang, L. Bai, A new initialization method for categorical data clustering, Expert Systems and Applications 36 (7) (2009) 10223–10228.

[16] F.Y. Cao, J.Y. Liang, D.Y. Li, L. Bai, C.Y. Dang, A dissimilarity measure for the *K*-Modes clustering algorithm, Knowledge-Based Systems 26 (2012) 120–127.

[17] J. Catlett, Megainduction: machine learning on very large database, University of Sydney, 1991 (PhD thesis).

[18] S. Chawla, A. Gionis, *K*-means-: A Unified Approach to Clustering and Outlier Detection, in: Proc. of the 13th SIAM Int. Conf. on Data Mining, Texas, USA, 2013, pp. 189–197.

[19] Y.M. Chen, D.Q. Miao, R.Z. Wang, Outlier detection based on granular computing, in: Proc. of the 6th International Conference on Rough Sets and Current Trends in Computing, Akron, USA, 2008, pp. 283–292.

[20] I. Düntsch, G. Gediga, Uncertainty measures of rough set prediction, Artificial Intelligence 106 (1998) 109–137.

[21] M. Ester, H.-P. Kriegel, J. Sander, X.W. Xu, Density-based algorithm for discovering clusters in large spatial databases with noise, in: Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining. Portland, 1996, pp. 226–231.

[22] A. Fred, A.K. Jain, Combining Multiple Clustering Using Evidence Accumulation, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (6) (2005) 835–850.

[23] S. Guha, R. Rastogi, K. Shim, CURE: An Efficient Clustering Algorithm for Large Databases, Proc. of ACM SIGMOD Int. Conf. on Management of Data, New York (1998) 73–84.

[24] Z.Y. He, X.F. Xu, S.C. Deng, Discovering cluster based local outliers, Pattern Recognition Letters 24 (9-10) (2003) 1641–1650.2003

[25] Z.Y. He, Farthest-point heuristic based initialization methods for *K*-modes clustering, 2006, CoRR, abs/cs/0610043.

[26] C.C. Hsu, C.L. Chen, Y.W. Su, Hierarchical clustering of mixed data based on distance hierarchy, Information Sciences 177 (20) (2007) 474–4492.

[27] Q.H. Hu, X.J. Che, L. Zhang, D. Zhang, M.Z. Guo, D.R. Yu, Rank entropy based decision trees for monotonic classification, IEEE Transactions on Knowledge and Data Engineering 24 (11) (2012) 2052–2064.

[28] Z.X. Huang, A fast clustering algorithm to cluster very large categorical data sets in data mining, in: Proc. of the SIGMOD Workshop Research Issues on Data Mining and Knowledge Discovery, Canada, 1997, pp. 1–8.

[29] Z.X. Huang, Extensions to the *k*-means algorithm for clustering large data sets with categorical values, Data Mining and Knowledge Discovery 2 (3) (1998) 283–304.

[30] Z.X. Huang, M.K. Ng, A Fuzzy *K*-modes Algorithm for Clustering Categorical Data, IEEE Transactions on Fuzzy Systems 7 (4) (1999) 446–452.

[31] Z.X. Huang, M.K. Ng, H.Q. Rong, Z.C. Li, Automated Variable Weighting in *K*-means Type Clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (5) (2005) 657–668.

[32] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.

[33] R. Jensen, Q. Shen, Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches, IEEE Press and Wiley & Sons, 2008.

[34] F. Jiang, Y.F. Sui, C.G. Cao, A rough set approach to outlier detection, International Journal of General Systems 37 (5) (2008) 519–536.

[35] F. Jiang, Y.F. Sui, C.G. Cao, A hybrid approach to outlier detection based on boundary region, Pattern Recognition Letters 32 (14) (2011) 1860–1870.

[36] F. Jiang, Y.F. Sui, L. Zhou, A relative decision entropy-based feature selection approach, Pattern Recognition 48 (7) (2015) 2151–2163.2015

[37] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data - An Introduction to Cluster Analysis, Wiley, 1990.

[38] S.S. Khan, A. Ahmad, Computing initial points using density based multiscale data condensation for clustering categorical data, in: Proc. of the 2nd International Conference on Applied Artificial Intelligence(ICAAI 03), Kolhapur, India, 2003.

[39] S.S. Khan, S. Kant, Computation of initial modes for *K*-modes clustering algorithm using evidence accumulation, in: Proc. of the 20th international joint conference on artificial intelligence (IJCAI 2007), Hyderabad, India, 2007, pp. 2784–2789.

[40] S.S. Khan, A. Ahmad, Cluster center initialization algorithm for *K*-modes clustering, Expert Systems with Applications 40 (18) (2013) 7444–7456.

[41] D.W. Kim, K.H. Lee, D. Lee, Fuzzy clustering of categorical data using fuzzy centroids, Pattern Recognition Letters 25 (11) (2004) 1263–1271.

[42] E.M. Knorr, R.T. Ng, Algorithms for mining distance-based outliers in large datasets, in: Proc. of the 24th VLDB Conference, New York, 1998, pp. 392–403.

[43] J.Y. Liang, Z.Z. Shi, The information entropy, rough entropy and knowledge granulation in rough set theory, Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 12 (1) (2004) 37–46.

[44] J.Y. Liang, Z.Z. Shi, D.Y. Li, M.J. Wierman, Information entropy, rough entropy and knowledge granularity in incomplete information systems, Int. Journal of General Systems 35 (6) (2006) 641–654.

[45] J.Y. Liang, J.H. Wang, Y.H. Qian, A new measure of uncertainty based on knowledge granulation for rough sets, Information Sciences 179 (4) (2009) 458–470.

[46] J.Y. Liang, X.W. Zhao, D.Y. Li, F.Y. Cao, C.Y. Dang, Determining the number of clusters using information entropy for mixed data, Pattern Recognition 45 (6) (2012) 2251–2265.

[47] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proc. of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[48] D.Q. Miao, G.R. Hu, A Heuristic Algorithm for Reduction of Knowledge, Computer Research and Development 36 (6) (1999) 681–684.

[49] P. Mitra, C.A. Murthy, S.K. Pal, Density-based multiscale data condensation, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (6) (2002) 734–747.

[50] M.K. Ng, M.J. Li, Z.X. Huang, Z.Y. He, On the impact of dissimilarity measure in *K*-Modes clustering algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (3) (2007) 503–507.

[51] S.H. Nguyen, H.S. Nguyen, Some efficient algorithms for rough set methods, in: Proc. of the 6th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU' 96), Granada, Spain, 1996, pp. 1451–1456.

[52] S.K. Pal, B.U. Shankar, P. Mitra, Granular computing, rough entropy and object extraction, Pattern Recognition Letters 26 (16) (2005) 2509–2517.

[53] Z. Pawlak, Rough Sets, International Journal of Computer and Information Sciences 11 (5) (1982) 341–356.

[54] Z. Pawlak, Rough sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, 1991.

[55] W. Pedrycz, Knowledge-Based Clustering: From Data to Information Granules, Wiley Interscience, New York, 2005.

[56] W. Pedrycz, A. Skowron, V. Kreinovich, Handbook of Granular Computing, Wiley Interscience, New York, 2008.

[57] Y.H. Qian, J.Y. Liang, F. Wang, A New Method for Measuring the Uncertainty in Incomplete Information Systems, Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 17 (6) (2009) 855–880.

[58] Y.H. Qian, F.J. Li, J.Y. Liang, B. Liu, C.Y. Dang, Space structure and clustering of categorical data, IEEE Transactions on Neural Networks and Learning Systems (2015). Accepted.

[59] C.E. Shannon, The mathematical theory of communication, Bell System Technical Journal 27 (3-4) (1948) 373–423.

[60] D. Slezak, Approximate Entropy Reducts, Fundamenta Informaticae 53 (2002) 365–390.

[61] Y. Sun, Q.M. Zhu, Z.X. Chen, An iterative initial-points refinement algorithm for categorical data clustering, Pattern Recognition Letters 23 (7) (2002) 875–884.

[62] G.Y. Wang, H. Yu, D.C. Yang, Decision table reduction based on conditional information entropy, Chinese Journal of Computers 25 (7) (2002) 759–766.

[63] M.J. Wierman, Measuring uncertainty in rough set theory, International Journal of General Systems 28 (4) (1999) 283–297.

[64] S. Wu, Q.S. Jiang, Z.X. Huang, A new initialization method for clustering categorical data, in: Proc. of the 11th Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining, in: Springer LNAI, vol. 4426, 2007, pp. 972–980.

[65] Z.Y. Xu, Z.P. Liu, B.R. Yang, W. Song, A Quick Attribute Reduction Algorithm with Complexity of $\max(O(|C||U|), O(|C|^2|U/C|))$, Chinese Journal of Computers 29 (3) (2006) 391–399.