Soft Comput
DOI 10.1007/s00500-012-0972-8

METHODOLOGIES AND APPLICATION

# Rough subspace-based clustering ensemble for categorical data

Can Gao · Witold Pedrycz · Duoqian Miao

**Abstract** Clustering categorical data arising as an important problem of data mining has recently attracted much attention. In this paper, the problem of unsupervised dimensionality reduction for categorical data is first studied. Based on the theory of rough sets, the attributes of categorical data are decomposed into a number of rough subspaces. A novel clustering ensemble algorithm based on rough subspaces is then proposed to deal with categorical data. The algorithm employs some of rough subspaces with high quality to cluster the data and yields a robust and stable solution by exploiting the resulting partitions. We also introduce a cluster index to evaluate the solution of clustering algorithm for categorical data. Experimental results for selected UCI data sets show that the proposed method produces better results than those obtained by other methods when being evaluated in terms of cluster validity indexes.

**Keywords** Categorical data · Rough sets · Fuzzy *k*-modes · Clustering ensemble · Cluster cardinality index

Communicated by A. Di Nola.

C. Gao · D. Miao
Department of Computer Science and Technology,
Tongji University, Shanghai 201804, People's Republic of China

C. Gao (✉) · W. Pedrycz
Department of Electrical and Computer Engineering,
University of Alberta, Edmonton, AB T6G 2G7, Canada
e-mail: 2008gaocan@gmail.com

W. Pedrycz
System Research Institute, Polish Academy of Sciences,
Warsaw, Poland

## 1 Introduction

Clustering is unsupervised learning when the data at hand are unlabeled. The essence of clustering is to partition a given set of unlabeled data into several clusters, in which the objects located within the same cluster are similar to each other, but quite dissimilar from those forming some other clusters. A large variety of clustering algorithms such as C-means (Ball and Hall 1967; Anderberg 1973; Jain 2010) and fuzzy C-means (FCM) (Bezdek 1981; Pedrycz 1996; Bargiela and Pedrycz 2005; Pedrycz et al. 2010) have been proposed and been widely used in real-world domains including data mining, information retrieval, machine learning and many others (Jain and Dubes 1988; Pedrycz 2005). Actually, clustering is a demanding combinatorial optimization task and no single clustering algorithm is capable of delivering sound solutions for all data sets.

Clustering ensemble (Ghaemi et al. 2009; Li et al. 2010; Vega-Pons and Ruiz-Shulcloper 2011), inspired by the idea of classifier ensemble encountered in supervised learning, has emerged as a technique for overcoming the problems associated with the individual clustering algorithms, such as robustness (Topchy et al. 2005), stability (Kuncheva and Vetrov 2006), parallelization (Tumer and Agogino 2008), and scalability (Hore et al. 2009), and has consequently found its applications in bioinformatics (Monti et al. 2003; Yu et al. 2007, 2011), image segmentation (Jiang and Zhou 2004; Yu et al. 2007; Zhang et al. 2008), feature selection (Hong et al. 2008a, b) and others (Yu et al. 2008). The method of clustering ensemble leverages the consensus across multiple clustering partitions and combines them into a single solution. Therefore, the process of clustering ensemble involves two stages. The first one is to produce a population of diverse multiple clustering partitions by a generative mechanism such as resampling (Fischer and

Published online: 10 January 2013

Buhmann 2003; Minaei-Bidgoli et al. 2004; Lange and Buhmann 2005; Yu and Wong 2009; Jia et al. 2011), attribute subspace or projection (Fern and Brodley 2003; Gionis et al. 2007; Al-Razgan et al. 2008), heterogeneous algorithm (Strehl and Ghosh 2002), homogenous algorithm with different initializations or parameters (Kuncheva and Vetrov 2006; Ayad and Kamel 2008), etc. The second stage is to combine multiple clustering results into a final solution, and some consensus functions have been proposed to deal with it. In Tumer and Agogino (2008), Fischer and Buhmann (2003), Ayad and Kamel (2008) and Ayad and Kamel (2010), a relabeling and voting method was introduced. The cluster labels of individual clustering were first aligned with respect to that of a reference clustering and then a voting scheme was employed to determine the final cluster label of objects. Strehl and Ghosh (2002), Topchy et al. (2005) and Luo et al. (2006) formalized clustering ensemble as a combinatorial optimization problem. Its objective was to find a mean partition that minimized its distance to all individual partitions in terms of normalized mutual information (NMI), generalized mutual information (GMI) or conditional entropy. The works in Yu et al. (2007), Strehl and Ghosh (2002), Fern and Brodley (2004) and Domeniconi and Al-Razgan (2009) represented clustering ensemble as a partitioning problem of a suitably defined graph or hypergraph whose vertices corresponded to the objects or clusters (or both the objects and clusters) and edges denoted their similarity. The final clustering solution was achieved using graph or hypergraph partition algorithm such as METIS, HMETIS, normalized cut or spectral clustering. In Fern and Brodley (2003), Fred and Jain (2005) and Iam-On et al. (2011, 2012), multiple clusterings were mapped into a matrix whose entries contained the pair-wise similarity of the objects over all clustering results and then a similarity-based clustering algorithm can be applied to yield a final solution. A different consensus function proposed by Topchy et al. (2005) and Lange and Buhmann (2005) defined the problem of clustering ensemble as maximum likelihood estimation and EM algorithm was used to find a consensus clustering solution. In addition, several papers study the general framework and extension of clustering ensemble. Yu et al. (2012) investigated the effect of data transformation on the performance of clustering ensemble, and a general form of transformation operator was proposed. Following that, they (Yu et al. 2012) extended clustering ensemble to structure ensemble. A novel structure ensemble approach based on graph partition was introduced to combine the structures obtained from different data sets generated by the bagging technique, and achieved better performance on practical cancer gene data sets.

Although many clustering ensemble algorithms have been proposed, they mainly deal with numeric data whose inherent geometric properties can be exploited to express distance between data points, which limits its use in the area of data mining where large categorical data are prevalent. In this paper, we mainly focus on the generative mechanism of diverse clustering partitions for categorical data, especially by the technique of attribute subspace, due to the identicalness between categorical data and numerical data in combining multiple clustering results. He et al. (2005) analyzed the relationships between clustering categorical data and clustering ensemble. Each attribute in categorical data was used to generate a partition of objects and then the resulting clustering partitions were combined by the consensus functions presented in Strehl and Ghosh (2002). Obviously, only one of all attributes is insufficient to generate a sound partition. Li and Chen (2010) extended the work reported in He et al. (2005) by weighting the attributes of categorical data. The redundant attribute was first removed by considering the implicit relationship between different attributes. Then, the similarity degree of objects was calculated by taking into account their relevance with respect to attribute weights. Nevertheless, the attribute weight is difficult to acquire without expert's involvement or domain knowledge. Al-Razgan et al. (2008) introduced a clustering ensemble method for categorical data based on random subspaces. The COOLCAT algorithm was used to produce the partition in each random subspace and the final clustering result was generated by the proposed categorical similarity partitioning algorithm (CSPA) and categorical bipartite partitioning algorithm (CBPA). Generally, the performance of clustering ensemble rests with not only the diversity of clustering partitions but also the quality of individual partition. Admittedly, random subspaces could yield diverse partitions. However, the quality of those resulting partitions could not be ensured because of the randomness in generating the attribute subspace.

Instead of single attribute subspace and random attribute subspace presented in Al-Razgan et al. (2008) and He et al. (2005), we propose a relevant subspace-based clustering ensemble algorithm for categorical data. Our motivation is from the following aspects: (1) it is quite likely that there are some irrelevant or redundant attributes in practical data, especially the data with high dimensions but small objects. (2) Several clusters may exist in different subspaces comprised of different combinations of attributes. In view of these characteristics of the data, we first exploit the theory of rough sets to eliminate the irrelevant and redundant attributes and then generate all relevant combinations of attributes (referred to as rough subspace hereafter). After ranking the generated rough subspaces by quality measure, our algorithm selects some rough subspaces with high quality to yield different clustering partitions. We anticipate that the selected rough subspaces not only preserve the granular structure of the original data but also bring high diversity of partitions. As a result, a sound solution could be obtained by combining the resulting partitions with a certain consensus function. In addition, we design a new internal index called cluster cardinality index (CCI) to evaluate the performance of clustering ensemble algorithm for categorical data. The new index is based on set operators, which could measure categorical data effectively. The experimental results demonstrate that the performance of the proposed algorithm is improved in comparison with some existing clustering ensemble algorithms and is substantially better than the result produced by an individual clustering algorithm.

The contributions of the paper are threefold. First, we introduce a concept of rough subspace into clustering ensemble, the aim of which is to remove the irrelevant and redundant attributes in categorical data and consequently improve the quality of individual clustering solution. Second, a novel clustering ensemble method based on ranked rough subspaces is presented to deal with categorical data, which could balance individual quality against diversity and provide a better final result. Third, a new internal cluster index called CCI is proposed to evaluate the performance of clustering ensemble algorithm.

The paper is organized as follows. Section 2 outlines some basic concepts related to our work. Section 3 proposes an unsupervised dimensionality reduction algorithm to find all rough subspaces of categorical data. Subsequently, a novel clustering ensemble method based on selected rough subspaces is introduced to deal with categorical data (Sect. 4). Section 5 reports on experimental results obtained from several UCI data sets (http://archive.ics.uci.edu/ml/). Finally, Sect. 6 concludes the paper and identifies several issues worth further investigation.

## 2 Related concepts and notations

In this section, we review some pertinent concepts, such as rough sets, fuzzy *k*-modes and clustering ensemble. Detailed investigations are reported in Ghaemi et al. (2009), Li et al. (2010), Vega-Pons and Ruiz-Shulcloper (2011), Pawlak (1982, 1991), Liu (2001), Wang (2001), Zhang et al. (2001), Miao and Li (2008) and Huang and Ng (1999).

### 2.1 Rough sets

In rough set theory, a bivariate table whose columns are labeled by attributes and rows are labeled by objects of interest is called information system. Formally, an information system is defined as $IS = (U, A, V, f)$, where $U$ is a nonempty and finite set of objects, called the universe; $A$ is a nonempty and finite set of attributes; $V$ is the union of attribute domains, i.e., $V = \cup V_a$, where $V_a$ denotes the domain for each attribute $a \in A$; and $f$ is an information function which associates a unique value of each attribute with every object belonging to $U$. If the attribute set $A$ can be divided into condition attribute set $C$ and decision attribute set $D$, the information system is also called decision information system or decision table. If not stated, the data sets in this study are all information systems with only categorical condition attributes (called categorical data hereafter).

An arbitrary attribute set $B$ of $A$ determines a binary relation $IND(B)$, called indiscernibility relation.

$$IND(B) = \{\langle x, y\rangle \in U \times U | \forall\, a \in B, f(x,a) = f(y,a)\} \quad (1)$$

Obviously, an indiscernibility relation is an equivalence relation which satisfies reflexivity, symmetry and transitivity. Arbitrary $\langle x,\ y\rangle$ belonging to $IND(B)$, it means that objects $x$ and $y$ are indiscernible with respect to $B$. The family of all equivalence classes of $IND(B)$, i.e., the partition of the universe $U$ determined by $B$, is denoted by $U/IND(B)$ or simply by $U/B$. While an equivalence class induced by $IND(B)$, i.e., the block of the partition $U/B$, is referred to as $B$-elementary set or $B$-elementary granule and is denoted by

$$[x]_B = \{y \in U | \langle x, y\rangle \in IND(B)\} \quad (2)$$

Rough set theory dwells upon two basic concepts, namely the lower and upper approximations of a set. Let $X$ be a subset of the universe $U$, the lower and upper approximations with respect to $B$ ($B \subseteq A$) are denoted as $\underline{B}(X)$ and $\overline{B}(X)$, respectively.

$$\underline{B}(X) = \{x \in U | [x]_B \subseteq X\} \quad (3)$$

$$\overline{B}(X) = \{x \in U | [x]_B \cap X \neq \emptyset\} \quad (4)$$

The $B$-lower approximation of the concept $X$ is the union of all $B$-elementary sets that are included in $X$, whereas the $B$-upper approximation of the concept $X$ is the union of all $B$-elementary sets that have a nonempty intersection with $X$. If $\underline{B}(X) = \overline{B}(X)$, $X$ is a crisp (definable) set with respect to $B$. Otherwise, $X$ is a rough (indefinable) set. $BND_B(X) = \overline{B}(X) - \underline{B}(X)$ is called the boundary of $X$ over $U$.

Let $IS = (U,\ A,\ V,\ f)$ be a categorical data, the discernible information among the objects can be represented by a symmetric matrix of dimensionality $|U| \times |U|$ (discernibility matrix). For any two objects $x_i$ and $x_j$, the element $r_{ij}$ of discernibility matrix $M$ is defined as $\{a \in A | a(x_i) \neq a(x_j)\}$.

Attribute reduction is a key problem in rough set theory. Given a categorical data $IS = (U,\ A,\ V,\ f)$ and its discernibility matrix $M$, for any subset $P \subseteq A$, if $P$ satisfies the conditions:

(I) for any element $r$ of $M$, $P$ has a nonempty intersection with $r$;

(II) no attribute can be eliminated from $P$ without affecting the requirement (I).

then, $P$ is a reduct of categorical data $IS$.

A reduct is a minimum subset of attributes that provide the same descriptive ability as the entire set of attributes. In other words, attributes in a reduct are jointly sufficient and individually necessary for clustering. Usually, there exist a number of reducts for a given categorical data. The intersection of all reducts is called attribute core in rough set theory.

### 2.2 Fuzzy *k*-modes

The method of fuzzy $k$-modes (Huang and Ng 1999) is one of the most popular individual clustering algorithms for categorical data. Specifically, it uses a simple matching dissimilarity measure as distance function, while the means presented in the fuzzy C-means are replaced by the modes. In the process of clustering, a frequency-based method is employed to update the modes in the FCM-like fashion to minimize the associated objective function. As the fuzzy $k$-modes algorithm uses the same clustering process as the FCM, it preserves the efficiency of the FCM algorithm.

Let $\boldsymbol{x}$, $\boldsymbol{y}$ be two categorical objects represented by vectors $[x_1, x_2, \ldots, x_m]$ and $[y_1, y_2, \ldots, y_m]$, respectively. Formally, the dissimilarity of the objects $\boldsymbol{x}$ and $\boldsymbol{y}$ is described as follows:

$$d(x, y) = \sum_{j=1}^{m} \delta(x_j, y_j) \tag{5}$$

where $\delta(x_j, y_j) = \begin{cases} 1 & x_j \neq y_j \\ 0 & x_j = y_j \end{cases}$.

The objective of fuzzy $k$-modes clustering is to split a set of $n$ categorical objects into $k$ clusters, i.e., to find $W$ and $Z$ that minimize the expression

$$J(X, W, Z) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{il}^{\alpha} d(X_i, Z_l) \tag{6}$$

Here $Z_l$ represents a set of modes for $k$ clusters and $w_{il}$ is an element of a fuzzy partition matrix. The fuzzy $k$-modes algorithm uses the FCM paradigm to cluster categorical data. However, the way to update the modes in each iteration is different from the one used in the FCM algorithm.

Let $\boldsymbol{X}$ be a set of objects described by categorical attributes $a_1, a_2, \ldots, a_m$, while the domain of attribute $a_j$ is denoted as $V(a_j) = \left\{a_j^{(1)}, a_j^{(2)}, \ldots, a_j^{(n_j)}\right\}$, where $n_j$ is the number of values of attribute $a_j$. Let $Z_l$ be a cluster center represented by $[z_{l1}, z_{l2}, \ldots, z_{lm}]$. Then, the quantity $J(X, W, Z)$ is minimized iff $z_{lj} = a_j^{(r)} \in V(a_j)$, where $\sum_{i, x_{ij} = a_j^{(r)}} w_{li}^{\alpha} \geq \sum_{i, x_{ij} = a_j^{(t)}} w_{li}^{\alpha}$, $1 \leq r \leq n_j$, $1 \leq t \leq n_j$, $t \neq r$, $1 \leq j \leq m$ and $1 \leq l \leq k$.

In each iteration, every attribute within cluster mode is given by the value that achieves the maximum of the summation of membership degree over all attribute values. If the value is not unique, the cluster mode will be arbitrarily assigned one of them.

### 2.3 Clustering ensemble

Cluster ensemble usually involves two stages. At the first one, some partitions of the data are produced by a given generative mechanism. Next, the cluster ensemble algorithm takes these partitions as input and uses a certain consensus function to form a single clustering partition to be treated as the final output.

Let $\boldsymbol{X}$ be a data of $n$ objects positioned in $m$-dimensional space, and $\boldsymbol{\Pi}$ be a set of $H$ partitions $\boldsymbol{\Pi} = \{\pi_1, \pi_2, \ldots, \pi_H\}$ of objects in $\boldsymbol{X}$. Each partition in $\boldsymbol{\Pi}$ is a set of disjoint and nonempty clusters $\pi_i = \left\{L_i^1, L_i^2, \ldots, L_i^{K(i)}\right\}$, $\boldsymbol{X} = L_i^1 \cup L_i^2 \cup \cdots \cup L_i^{K(i)}$, and for any $\pi_i$, $K(i)$ is the number of clusters in the $i$-th clustering partition. The problem of clustering ensemble is to find a new partition $\boldsymbol{\sigma} = \{C_1, C_2, \ldots, C_K\}$ of data $\boldsymbol{X}$ given the partitions in $\boldsymbol{\Pi}$, such that the final clustering solution is better than any individual clustering partition.

This statement of the problem is virtually the same as for the "conventional" clustering except that it uses the information contained in the already existing partitions. Other variants of this definition could be obtained by imposing some additional requirements on the target partition, such as fixing the number of clusters, or admitting membership values for data objects.

## 3 Rough subspaces of categorical data

In theory, a clustering algorithm with more attributes should provide better descriptive power, but in practice, with a limited number of data, excessive attributes not only slow down the learning process, but also result in poor performance as irrelevant and redundant attributes may confuse the learning algorithm. Attribute reduction (feature selection) is a research field, which has been shown effective in enhancing learning efficiency, improving learning performance, and reducing the complexity of learning algorithm. In rough set theory, attribute reduction is a key research problem and many useful algorithms have been proposed (Thangavel and Pethalakshmi 2009; Miao et al. 2009; Zhou et al. 2011). A reduct (rough subspace) is a subset of all attributes, which not only excludes irrelevant and redundant attributes, but also keeps the granular structure of the original data.

Generally, the entire set of attributes can be classified into three disjoint categories by exploiting the correlation measure, namely strongly relevant, weakly relevant and irrelevant attributes. Strong relevance of an attribute indicates that this attribute is always necessary for clustering; it cannot be removed without affecting the original granular structure. Weak relevance suggests that the attribute is not always important but may become necessary for clustering under certain conditions. Irrelevance indicates that the attribute is not necessary at all. An optimal rough subspace should include all strongly relevant attributes, none of irrelevant attributes, and a subset of weakly relevant ones.

In fact, with the concept of discernibility matrix in rough set theory, it is easy to categorize the attributes into three different sets. Each attribute in core set is closely relevant to clustering, namely strongly relevant attribute; the attribute that does not appear in discernibility matrix is completely irrelevant to clustering, namely irrelevant attribute; and the other attributes are weakly relevant. In order to elaborate on our algorithm, we first present some related concepts.

Formally, a categorical data is denoted as $IS = (U, A, V, f)$. From another viewpoint, the problem of clustering categorical data is to find a label for every object in the data and finally form a decision table. In order to ensure the validity of attribute reduction algorithm, we need to discuss the relationships of categorical data itself and its corresponding decision table, namely resulting partition of clustering algorithm. In what follows, the decision table induced by clustering algorithm is denoted by $DS = (U, A = C \cup D, V, f)$ and is referred to as underlying decision table.

**Proposition 1** *Let $IS = (U, A, V, f)$ be a categorical data. If $Core_1$ is the set of core attributes in categorical data IS and $Core_2$ is the set of core attributes in underlying decision table DS, then the formula $Core_2 \subseteq Core_1$ holds.*

*Proof* Reductio ad absurdum. Assume that there is a core attribute $a \in Core_2$ but $a \notin Core_1$. Then, it follows that there exist two objects $X_i$ and $X_j$ belonging to two different classes and only attribute $a$ can discern them. In other words, objects $X_i$ and $X_j$ are indiscernible without attribute $a$. From the definition of discernibility matrix, any discernible information between the objects will be produced. Consequently, the discernibility matrix of the categorical data $IS$ definitely contains a singleton set "$\{a\}$". Therefore, the formula $a \in Core_1$ holds. In other words, any attribute in the core set of the underlying decision table $DS$ is also a core attribute of the categorical data $IS$. The proposition has been proved.

**Proposition 2** *Let $IS = (U, A, V, f)$ be a categorical data. If $RED_1$ is a rough subspace of the categorical data IS, there must exist a rough subspace $RED_2$ in the underlying decision table DS and the formula $RED_2 \subseteq RED_1$ holds.*

*Proof* In order to maintain the descriptive ability of the categorical data, each discernible information between the objects is preserved in the discernibility matrix of the categorical data $IS$. However, only the discernible information related to the objects that have different class labels is presented in discernibility matrix of the underlying decision table $DS$. Therefore, the collection of elements in the discernibility matrix of $DS$ is a subset of that in the discernibility matrix of $IS$. Assume that $r_1$ is an element of the difference set of the collections of $IS$ and $DS$. Then, there exist three different relationships between $r_1$ and the elements of the collection $M_{DS}$ of $DS$.

1. $\exists r_2 \in M_{DS}$, it has $r_2 \subseteq r_1$. According to the definition of rough subspace, $RED_2$ should have a nonempty intersection with every element in $M_{DS}$. Therefore, the intersection of $RED_2$ and $r_1$ is definitely nonempty under the condition $r_2 \subseteq r_1$. In other words, $RED_2$ is sufficient to discern the objects which produce discernible information $r_1$. If this case holds for all elements in the difference set of the collections of $IS$ and $DS$, the rough subspace of $IS$ is the same as that of $DS$.
2. $\exists r_2 \in M_{DS}$, it has $r_2 \supset r_1$. Under these circumstances, some rough subspaces in $DS$ may be not enough to discern the objects which produce discernible information $r_1$. But the rough subspace that contains an element of the intersection set of $r_1$ and $r_2$ is sufficient for those objects. Therefore, for any rough subspace in $M_{DS}$ and $r_1$, there must exist a rough subspace in $DS$ and these two rough subspaces have same attributes.
3. $\forall r_2 \in M_{DS}$, it has $r_2 \not\subset r_1$ and $r_1 \not\subset r_2$. In this case, the rough subspace $RED_2$ in $DS$ may be not sufficient to discern all objects in $IS$, and some attributes will be added to be the rough subspace of $IS$. Therefore, the rough subspace of $DS$ will be included by that of $IS$.

In all cases, a rough subspace of $IS$ includes at least one of rough subspaces of $DS$. The proof of the proposition has been completed.

The propositions presented above show the implicit relationship of the categorical data and its underlying decision table. They guarantee that the rough subspace of categorical data can hold the descriptive ability for any underlying decision table. Therefore, the process of attribute reduction based on discernibility matrix is effective in dealing with categorical data. In the following sections, we first introduce two set operators defined on discernibility matrix, and an unsupervised algorithm based on the two set operators is then proposed to find all rough subspaces of categorical data.

**Definition 1** Let $M$ be discernibility matrix of a categorical data $IS = (U, A, V, f)$, for any subset $B$ of $A$, the relevant set of $B$ within $M$ is defined as $RS_M(B) = \{K \in M | K \cap B \neq \emptyset\}$.

**Definition 2** Let $M$ be discernibility matrix of a categorical data $IS = (U, A, V, f)$, for any subset $B$ of $A$, the complement set of $B$ within $M$ is defined as $CS_M(B) = \{K - B | K \in M\}$.

With the definitions stated above, a rough subspace could be produced through the following steps. Core attributes are first added to rough subspace, and then the algorithm iteratively puts a maximum frequency attribute in the rough subspace and at the same time the relevant set of this attribute is removed from discernibility matrix $M$ until $M$ is empty. As for all rough subspaces, we could employ the concept of complement set for the attribute to generate all possible combinations of attributes. The detailed procedure is shown in Algorithm 1.

Algorithm 1 involves two closely integrated stages: (1) categorizes all attributes into relevant, weakly relevant and irrelevant attribute sets with discernibility matrix, and (2) recursively explores the rough subspace from data stack. At the first stage (from Step 1 to Step 3), it first computes the discernibility matrix of the given categorical data, and some redundant elements are removed from the discernibility matrix using the law of absorption. Then, all attributes are partitioned into three different attribute sets, namely core, candidate, and irrelevant attribute sets. At the second stage (from Step 4 and Step 9), the algorithm first adds core attributes to the rough subspace. Then, it iteratively selects a weakly relevant attribute with maximum frequency within current candidate information set. Meanwhile, the relevant set of this attribute within current candidate information set is removed from the discernibility matrix. The algorithm produces a rough subspace when current information set is empty. In Step 8, the complement set of the selected attribute is saved by the algorithm if it differs from the current information set. Because the complement set of an attribute does not contain the attribute itself, it can be used to compute other rough subspaces without that attribute. As a result, the difference comes into existence between the produced rough subspaces. The algorithm terminates when the data stack is empty, which means that all possible combinations of attributes have been investigated.

In Algorithm 1, the major computing focuses on establishing the discernibility matrix. Assume that $|C| = m$, $|U/C| = n$, $C_m^{\lceil m/2 \rceil} = k$. The time complexity of forming the discernibility matrix is $O(mn^2)$. Because of the symmetry, there are only $n$ $(n-1)/2$ useful elements in the matrix. By virtue of the absorption law, the number of necessary elements in the discernibility matrix will decrease to the worse-case $k$ from $n$ $(n-1)/2$ (Wang and

**Algorithm 1 Search all rough subspaces of categorical data**

**Input:** A categorical data $IS=(U, A, V, f)$

**Output:** All rough subspaces of $IS$

```
Step 1 Set discernibility matrix $M=\emptyset$, core set $Core=\emptyset$ and all rough subspace set $REDS=\emptyset$;
Step 2 Compute the discernibility matrix $M$ and exclude unnecessary elements of discernibility
       matrix $M$ with the law of absorption;
Step 3 Add the singleton in $M$ to core set $Core$, $M=M-Core$; push $Core$ and $M$ to data stack $L$
       for rough subspace;
Step 4 If $L$ is null, then goto step 10;
Step 5 Pop the last attribute set $RED$ and candidate information $M'$;
Step 6 If $M'$ is null, $REDS=REDS\cup RED$, goto Step 4;
Step 7 Select maximum frequency attribute $a$, $M'=M'- RS_{M'}(\{a\})$, exclude unnecessary elements
       of $CS_{M'}(\{a\})$ with the law of absorption;
Step 8 If $M'\neq CS_{M'}(\{a\})$, push $RED$ and $CS_{M'}(\{a\})$ to $L$ // $CS_{M'}(\{a\})$ is used to compute the rough
       subspaces without attribute $a$;
Step 9 $RED=RED\cup\{a\}$; push $RED$ and $M'$ to $L$;
Step 10 Return $REDS$.
```

Gao 2009). In Step 7, if an attribute has been selected, this attribute and its supersets are removed from current candidate information set. In worst-case, the candidate information set will be null after $m = |C|$ times. Based on the discernibility matrix, the time complexity of computing a rough subspace is $O(mk)$, which is approximate to $O(mn^2)$. The space complexity is $O(k)$. When there are $m$ attributes in a given categorical data, the number of all rough subspaces is less than $O(k)$ (Wang and Gao 2009). Therefore, the total time complexity of Algorithm 1 is $O(kn^2m)$, and the space complexity is only $O(kn)$.

To fully illustrate our proposed algorithm, we give a categorical data whose objects are represented by six categorical attributes (Table 1). The detailed procedure is shown in Algorithm 1.

**Table 1** A toy categorical data

| | $a$ | $b$ | $c$ | $d$ | $e$ | $f$ |
|---|---|---|---|---|---|---|
| $x_1$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $x_2$ | 0 | 0 | 0 | 1 | 0 | 0 |
| $x_3$ | 0 | 0 | 1 | 0 | 1 | 0 |
| $x_4$ | 1 | 1 | 0 | 0 | 0 | 0 |
| $x_5$ | 1 | 1 | 0 | 1 | 0 | 1 |

**Table 2** The discernibility matrix for Table 1

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|
| $x_1$ | $\emptyset$ | | | | |
| $x_2$ | $\{d\}$ | $\emptyset$ | | | |
| $x_3$ | $\{c, e\}$ | $\{c, d, e\}$ | $\emptyset$ | | |
| $x_4$ | $\{a, b\}$ | $\{a, b, d\}$ | $\{a, b, c, e\}$ | $\emptyset$ | |
| $x_5$ | $\{a, b, d, f\}$ | $\{a, b, f\}$ | $\{a, b, c, d, e, f\}$ | $\{d, f\}$ | $\emptyset$ |

The algorithm first generates a discernibility matrix whose entries denote the discernible information between the objects (Table 2). With the law of absorption, the discernibility matrix is reduced to $\{\{d\}, \{a, b\}, \{c, e\}\}$. Then, core attribute $d$ is set to initial element of all rough subspaces and is pushed into data stack $L$ with other discernible information $\{\{a, b\}, \{c, e\}\}$. In the last pushed candidate information $\{\{a, b\}, \{c, e\}\}$, attribute $a$ is selected to be candidate for rough subspace, and its complement set $\{\{b\}, \{c, e\}\}$ and attribute set $\{d\}$ are pushed into $L$ for computing the rough subspaces without attribute $a$. While the attribute set $\{d, a\}$ and candidate information except the relevant set of attribute $a$ are also stored to generate the rough subspaces containing attribute $a$. In the second round of selection, attribute $c$ is chosen and its complement set $\{\{e\}\}$ with attribute set $\{d, a\}$ is deposited. Following that, a rough subspace $\{d, a, c\}$ is produced because current candidate information is null. Other rough subspaces could be deduced by analogy. The overall process of generating all rough subspaces can be depicted as Table 3 and be structured as a binary tree in Fig. 1.

## 4 Combining rough subspace-based partitions

### 4.1 Structure of rough subspace-based clustering ensemble

Based on Algorithm 1, the whole attributes can be grouped into a number of rough subspaces. Each rough subspace preserves the clustering power of the original attribute set. Therefore, it is sufficient to cluster the data. Moreover, different rough subspaces describe the data from different viewpoints, which suggests that we could choose some rough subspaces to cluster the data and ensemble them to obtain a good clustering solution. The structure of rough

**Table 3** The dynamic process of Algorithm 1 for Table 1

| Iteration | Data stack $L$ | Attribute set | $M'$ | Candidate | $RED$ |
|---|---|---|---|---|---|
| Initialization | $d$:$\{\{a, b\},\{c, e\}\}$ | $\emptyset$ | $\emptyset$ | $\emptyset$ | – |
| 1 | $\emptyset$ | $\{d\}$ | $\{\{a, b\}, \{c, e\}\}$ | $a$ | – |
| 2 | $d$:$\{\{b\}, \{c, e\}\}$ | $\{d, a\}$ | $\{\{c, e\}\}$ | $c$ | – |
| 3 | $d, a$:$\{\{e\}\}$ | | | | |
| | $d$:$\{\{b\}, \{c, e\}\}$ | $\{d, a, c\}$ | $\emptyset$ | – | $\{d, a, c\}$ |
| 4 | $d$:$\{\{b\}, \{c, e\}\}$ | $\{d, a\}$ | $\{\{e\}\}$ | $e$ | – |
| 5 | $d$:$\{\{b\}, \{c, e\}\}$ | $\{d, a, e\}$ | $\emptyset$ | – | $\{d, a, e\}$ |
| 6 | $\emptyset$ | $\{d\}$ | $\{\{b\}, \{c, e\}\}$ | $b$ | – |
| 7 | $\emptyset$ | $\{d, b\}$ | $\{\{c, e\}\}$ | $c$ | – |
| 8 | $d, b$:$\{\{e\}\}$ | $\{d, b, c\}$ | $\emptyset$ | – | $\{d, b, c\}$ |
| 9 | $\emptyset$ | $\{d, b\}$ | $\{\{e\}\}$ | $e$ | – |
| 10 | $\emptyset$ | $\{d, b, e\}$ | $\emptyset$ | – | $\{d, b, e\}$ |
| End | $\emptyset$ | $\emptyset$ | $\emptyset$ | – | – |

subspace-based clustering ensemble for categorical data is shown in Fig. 2.

### 4.2 Algorithm

Our aim here is to arrive at a robust and stable solution via clustering ensemble technique. As Fig. 2 depicts, the clustering ensemble model will employ some rough subspaces to cluster the data. When there are many rough subspaces in a given categorical data, how to choose a set of good and diversified rough subspaces is a key problem for the success of clustering ensemble. In classifier ensemble, some scholars argued that ensemble with some selective classifiers was better than that of all available classifiers (Zhou et al. 2002; Rokach 2010). The study in Hadjitodorov et al. (2006) also indicated that a moderate number of clustering results were better for clustering ensemble. In light of those works, we try to rank rough subspaces by their significance and performance, and only combine some high quality rough subspace-based clustering partitions. More specifically, the quality of rough subspace is first expressed by its attributes and cardinality, namely the significance of each attribute and the number of attributes in the rough subspace. The rough subspace with a fewer number of high frequency attributes will be superior to clustering ensemble. The rationale behind our approach is that the rough subspace with higher significant attribute could result in better clustering solution, and the one with smaller cardinality is more likely to avoid redundant and irrelative attributes which will deteriorate the performance of clustering algorithm. Formally, the quality measure of rough subspace can be expressed by formula (7). In addition, the rough subspace is evaluated by its performance in terms of cluster validity indexes. Therefore, the strategy for choosing rough subspace relies on achieving a tradeoff between the rough subspace itself and its performance. The higher the significance and performance of rough subspace, the higher its suitability for clustering ensemble.

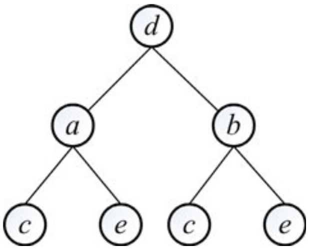


**Fig. 1** The tree structure of all rough subspaces

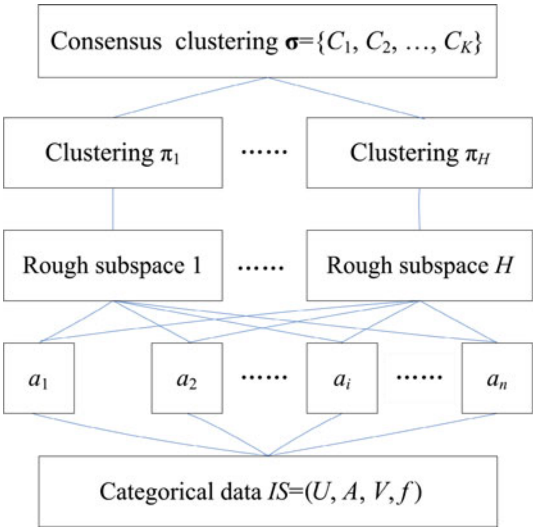


**Fig. 2** Rough subspace-based clustering ensemble model

$$Q(R_i) = |R_i|^{-1}\sum_{j=1}^{|R_i|} f(a_j) \tag{7}$$

Based on the model and quality measure of rough subspace mentioned previously, the algorithm to implement rough subspace-based clustering ensemble for categorical data is presented in Algorithm 2.

## 5 Experiments

In the experiments, we consider three clustering ensemble algorithms, namely the algorithm introduced in this study

```
Algorithm 2 Rough subspace-based clustering ensemble for categorical data
Input: A categorical data $IS=(U, A, V, f)$
Output: An optimal clustering solution $\sigma$
Step 1 Compute all rough subspaces $REDS$ of the categorical data $IS$ by Algorithm 1;
Step 2 Rank the rough subspace $R_i$ in $REDS$ with quality measure;
Step 3 Generate a set of clustering partitions $\Pi=\{\pi_1, \pi_2, \ldots, \pi_H\}$ with $H$ selected rough
       subspaces;
Step 4 Combine a set of clustering partitions $\Pi$ with consensus function;
Step 5 Return clustering ensemble solution $\sigma$.
```

(RSCE), ccdByEnsemble algorithm (CCDE) presented in He et al. (2005) and a random subspace-based clustering ensemble algorithm proposed in Al-Razgan et al. (2008) (RDCE). The experiments are designed to demonstrate: (1) the advantage of ensemble clustering over individual clustering; (2) the performance improvement of the proposed ensemble method compared with the other two ensemble algorithms; and (3) the effectiveness of a new cluster validity index for categorical data.

### 5.1 Data sets and parameter settings

Several UCI data sets are used in the experiments. The detailed information of these data sets is shown in Table 4, where the last column is the number of rough subspaces in each data set. Data sets "Dermatology" and "German Credit" contain few continuous attributes. We use the principle of equal frequency (Øhrn and Komorowski 1997) to discretize the continuous data, while the missing values in the last attribute of data set "Dermatology" are filled by conditioned mean (or mode) (Øhrn and Komorowski 1997).

In the experiments, we use fuzzy $k$-modes as individual clustering algorithm and hypergraph as a consensus function (Strehl and Ghosh 2002), where the fuzzification coefficient $m$ of fuzzy $k$-modes varies from 1.1 to 2.0. For each data set, Algorithm 1 is first used to generate all rough subspaces. Each rough subspace is then assessed by the measure presented in Sect. 4 and only some higher quality rough subspaces are selected for clustering ensemble. Fuzzy $k$-modes repeats ten times on each selected rough subspace with randomly initial modes. The clustering partition exhibiting the lowest objective function is chosen as the individual result for clustering ensemble. As for the number of clustering partitions for ensemble, it will vary with different data sets. Because of the randomness in the initial modes of individual fuzzy $k$-modes algorithm, we complete the ensemble algorithm ten times on each data set and adopt the average to describe their performance.

### 5.2 Evaluation criteria

Evaluating the quality of clustering is, in general, a difficult task. Although many cluster validation indexes (Wang and Zhang 2007) have been proposed, they mainly deal with numeric data. Since class labels are available for the data sets used here, we first evaluate the result of clustering algorithm by computing the accuracy. Then, a new cluster validity index without class information is introduced for categorical data, called CCI.

The accuracy is computed based on the confusion matrix produced by the clustering results. Given the number of clusters $k$, the clustering accuracy (ACC) is expressed as $\sum_{i=1}^{k} a_i/n$, where $n$ is the number of objects in the data set, $a_i$ is the number of objects occurring in both cluster $i$ and its corresponding class.

Cluster evaluation for categorical data is more difficult than numerical ones because of unordered attribute values. Inspired by set operations, which are often used to describe the property and structure of categorical data, we define a cluster index for categorical data as follows:

$$CCI = \frac{1}{k}\sum_{i=1}^{k} \max_{j:i\neq j}\left\{\frac{CI(i)+CI(j)}{CI(i,j)}\right\} \tag{8}$$

where $CI(i)=\frac{1}{m}\sum_{d=1}^{m}\frac{|A_{id}|}{|A_i|}$, $CI(i,j)=\frac{1}{m}\sum_{d=1}^{m}\frac{|A_{id}\cup A_{jd}|-|A_{id}\cap A_{jd}|+1}{|A_{id}\cup A_{jd}|+1}$, $A_{id}$ and $A_{jd}$ are the sets of categorical values of $d$th attribute within clusters $i$ and $j$, and $A_i$ is the set of objects within cluster $i$. The symbol "| |" denotes the cardinality of a set. In this formula, $CI(i)$ is the average number of categorical values of cluster $i$ over all attributes and $CI(i, j)$ is the average number of different categorical values between clusters $i$ and $j$ over all attributes. The CCI tries to minimize the average dissimilarity of objects within the same cluster with a smaller number of categorical values in each attribute and maximize the dissimilarity of different clusters with a larger number of different categorical values in all attributes. Hence, the CCI is small if the clusters are compact and far from each other.

### 5.3 Comparative results and discussion

The first data set, referred to as "Soybean", contains 47 objects. Each object is represented by 35 categorical attributes and is classified into one of the four classes. Three clustering ensemble algorithms and one single

**Table 4** Benchmark data sets

| Data sets | Attributes | Instances | Num. of Classes | Num. of rough subspaces |
|---|---|---|---|---|
| Lung-cancer (Lung) | 56 | 32 | 3 | >1,000 |
| Marine Sponges (Sponge) | 45 | 76 | 12 | >1,000 |
| Soybean-small (Soybean) | 35 | 47 | 4 | 310 |
| Dermatology (Dermat) | 34 | 366 | 6 | >1,000 |
| German Credit (German) | 24 | 1,000 | 2 | 64 |
| Lymphography (Lymph) | 18 | 142 | 2 | 72 |
| Tic-Tac-Toe (TTT) | 9 | 958 | 2 | 9 |

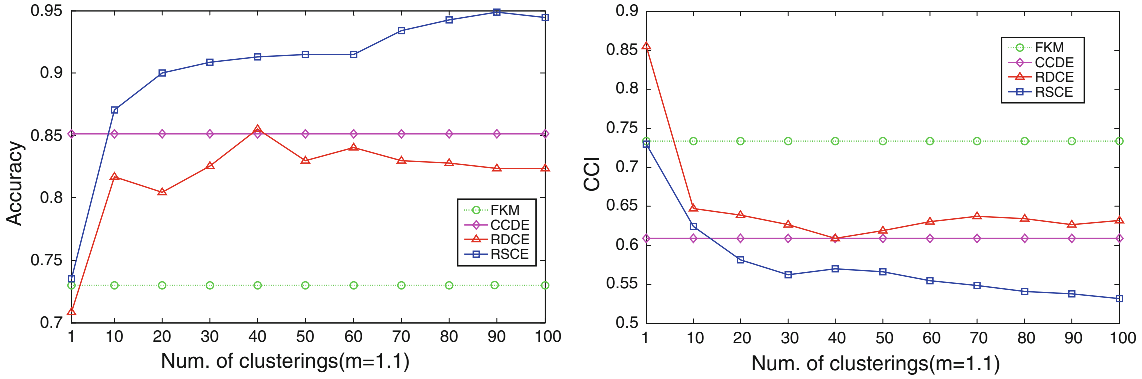


**Fig. 3** The performance of the selected algorithms for different numbers of clustering partitions ($m = 1.1$)

clustering algorithm (Fuzzy *k*-modes) are used to partition this data set into four clusters ($k = 4$). The number of clustering partitions for ensemble algorithm in this data varies from 1 to 100. Each algorithm is run ten times and the results are averaged. Huang and Ng (1999) indicated that fuzzy *k*-modes provided best performance when the fuzzification coefficient is set to be $m = 1.1$. Therefore, we first run the algorithms at $m = 1.1$. The results with respect to two different evaluation criteria are shown in Fig. 3.

We note that the performance of individual fuzzy *k*-modes algorithm is much worse than that of clustering ensemble algorithms. The reason behind this phenomenon could be ascribed to two points. One is that fuzzy *k*-modes is a single clustering algorithm without any other useful information to improve the performance, and the other one is that the attribute subspace used in fuzzy *k*-modes may contain irrelevant and redundant attributes, resulting in worse performance. However, clustering ensemble algorithms could benefit from the consensus strategy of multiple clustering partitions and be able to filter out spurious structures identified by individual clustering algorithm. Among the three clustering ensemble algorithms, the proposed method performs quite well, yielding 8.6 % average improvement compared with that of clustering algorithm based on random subspaces and almost 21.9 % maximum improvement over the results produced by the individual fuzzy *k*-modes.

In Fig. 3, the proposed algorithm produces the highest performance when the number of clustering partitions is set as 90. Under this number of clustering partitions, the experiments are carried out for other values of the fuzzification coefficient ($1.2 \leq m \leq 2.0$) and the results are shown in Fig. 4. The experiments with other numbers of clustering partitions are also evaluated. Under a given number of clustering partitions, the selected algorithms perform on the value of fuzzification coefficient from 1.1 to 2.0, and ten performance results are consequently obtained. To summarize these results, the area under the curve of performance results (like the performance curve in Fig. 4) is quantified to describe the overall performance of the selected algorithms for a given number of clustering partitions. The measure of “the area under the curve (AUC)” not only reflects the mean of performance results but also the trend in their variance. More specifically, under the condition of the same performance mean, the AUC will favor the set of results that include the highest performance. Under different numbers of clustering partitions, the overall performance of the selected algorithms is shown

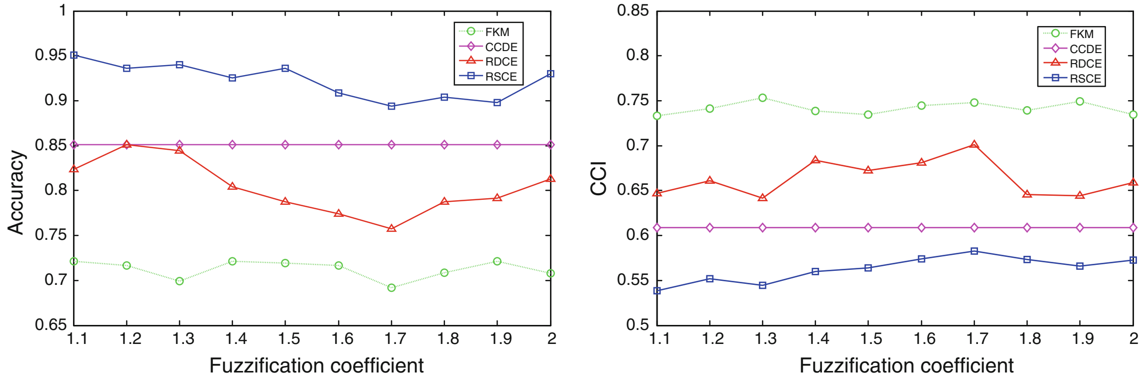


**Fig. 4** The performance of the selected algorithms for different fuzzification coefficients ($H = 90$)

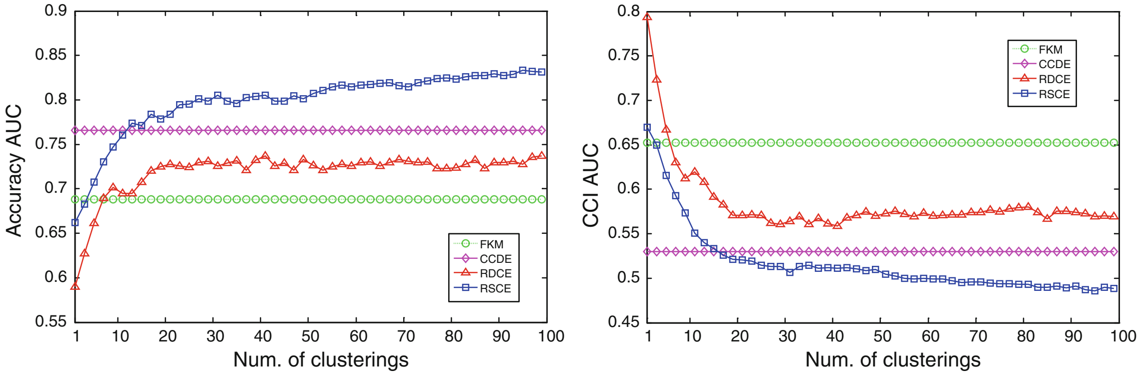


**Fig. 5** The overall performance of the selected algorithms for the Soybean data

in Fig. 5. Note that each point in the subfigure corresponds to the value of AUC with respect to the performance of the algorithm completed for different values of fuzzification coefficient.

In addition, the experiments on other data sets are performed. For each data set under a specific fuzzification coefficient, we complete the selected algorithms ten times and the best result is used to describe their performance. The overall experimental results are shown in Tables 5 and 6. In the columns of the selected algorithms, the evaluation indexes "avg." and "max" (or "min") denote the average and maximum (or minimum) performance over different fuzzification coefficients ($1.1 \leq m \leq 2.0$), respectively. The maximum performance improvement of the clustering ensemble algorithms is denoted by "improv.", which can be computed by the reduction of the best performance of the clustering ensemble algorithm over its initial performance. The fuzzification coefficient under

**Table 5** The accuracy of the selected algorithms for UCI data sets

| Data sets | FKM | | | CCDE | | RDCE | | | RSCE | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg. | Max | *m* | Max | Avg. | Max | Improv. | *m* | Avg. | Max | Improv. | *m* |
| Lung | 0.5963 | 0.6281 | 1.1 | 0.6250 | 0.7063 | 0.7583 | 0.2188 | 1.2 | **0.7219** | **0.7813** | **0.3750** | 1.1 |
| Sponge | 0.5066 | 0.5526 | 1.3 | 0.5526 | 0.6132 | 0.6247 | 0.2895 | 1.3 | **0.6289** | **0.6316** | **0.3026** | 1.3 |
| Soybean | 0.7213 | 0.8062 | 1.1 | 0.8511 | 0.9362 | 0.9575 | 0.4894 | 1.3 | **0.9575** | **0.9575** | **0.5532** | 1.1 |
| Dermat | 0.4862 | 0.5262 | 1.1 | 0.5828 | 0.6320 | 0.6511 | **0.3033** | 1.1 | **0.6545** | **0.6891** | 0.2158 | 1.1 |
| German | 0.5057 | 0.5410 | 1.8 | 0.5030 | 0.5499 | **0.5690** | **0.1710** | 1.3 | **0.5572** | 0.5610 | 0.1690 | 1.3 |
| Lymph | 0.6880 | 0.7254 | 1.3 | 0.6549 | 0.7445 | 0.7606 | **0.2394** | 1.4 | **0.7746** | **0.7887** | 0.0845 | 1.3 |
| TTT | 0.6785 | 0.7213 | 1.6 | 0.5198 | 0.7002 | 0.7286 | 0.1253 | 1.6 | **0.7598** | **0.7693** | **0.2578** | 1.6 |
| Avg. | 0.5975 | 0.6430 | – | 0.6127 | 0.6975 | 0.7214 | 0.2624 | – | **0.7221** | **0.7398** | **0.2797** | – |

Bold values indicate the best performance for each evaluation index

**Table 6** The CCI of the selected algorithms for UCI data sets

| Data sets | FKM | | | CCDE | RDCE | | | | RSCE | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg. | Min | *m* | Min | Avg. | Min | Improv. | *m* | Avg. | Min | Improv. | *m* |
| Lung | 0.9741 | 0.9315 | 1.1 | 0.9367 | 0.8908 | 0.8663 | 0.3041 | 1.2 | **0.8782** | **0.8516** | **0.3583** | 1.1 |
| Sponge | 1.0698 | 1.0360 | 1.3 | 1.0360 | 0.9756 | 0.9380 | **0.4462** | 1.3 | **0.9356** | **0.9211** | 0.4087 | 1.3 |
| Soybean | 0.7648 | 0.6243 | 1.1 | 0.5889 | 0.5284 | 0.5152 | **0.6379** | 1.3 | **0.4503** | **0.4503** | 0.3766 | 1.1 |
| Dermat | 0.3527 | 0.3248 | 1.1 | 0.3068 | 0.2828 | 0.2745 | 0.1239 | 1.1 | **0.2726** | **0.2382** | **0.1829** | 1.1 |
| German | 0.0482 | 0.0435 | 1.8 | 0.0484 | **0.0420** | **0.0409** | **0.0131** | 1.3 | 0.0425 | 0.0412 | 0.0047 | 1.3 |
| Lymph | 0.2401 | 0.2398 | 1.3 | 0.2571 | 0.2253 | 0.2139 | **0.0578** | 1.4 | **0.2068** | **0.1956** | 0.0346 | 1.3 |
| TTT | 0.0469 | 0.0379 | 1.6 | 0.0518 | 0.0413 | 0.0385 | 0.0001 | 1.6 | **0.0294** | **0.0213** | **0.0090** | 1.6 |
| Avg. | 0.4995 | 0.4625 | – | 0.4608 | 0.4266 | 0.4125 | **0.2262** | – | **0.4022** | **0.3885** | 0.1964 | – |

Bold values indicate the best performance for each evaluation index which the selected algorithms achieve the best performance is also listed in the table. The row “avg.” in each table shows the average results over all experimental data sets, and the highest values for selected evaluation indexes on each data set have been boldfaced.

Tables 5 and 6 show that the average and maximum performance of the clustering ensemble algorithms are generally better than the individual algorithm. The ccdByEnsemble algorithm is deterministic; therefore, its average performance is not listed in the tables. Although the ccdByEnsemble algorithm uses the technique of clustering ensemble, its mechanism is a little different from the other two clustering ensemble algorithms in the component. The individual result of the ccdByEnsemble algorithm is directly from the values of the categorical attribute rather than the single clustering algorithm. Therefore, it is worse than fuzzy $k$-modes on such data sets as “Lung”, “German”, “Lymph” and “TTT”. The average and maximum performance of the clustering ensemble algorithms (RDCE and RSCE) are always better than their individual algorithm (FKM). Although the maximum performance improvement is obtained by the RDCE on some data sets, its average and maximum performance are not better than that of our proposed one. By averaging the performance over all data sets, the average accuracy and CCI values of the RDCE increase by 10 and 7.3 %, respectively, whereas our proposed one achieves an overall 12.5 and 9.7 % improvement over its individual clustering algorithm. As for the fuzzification coefficient, the best value for the selected clustering algorithms varies from different data sets, but our proposed algorithm is almost consistent with its individual fuzzy $k$-modes algorithm.

Under different fuzzification coefficients and numbers of clustering partitions, the overall performance of the selected algorithms on each data set is provided in Fig. 6, in which we report on the AUC values of accuracy and CCI for the ensemble algorithms, as well as the performance of fuzzy $k$-modes obtained in case when all attributes are used.

Rough subspace-based clustering ensemble algorithm (RSCE) achieves significant improvement over most of the data sets as shown in Figs. 5 and 6. RDCE and ccdByEnsemble algorithm also attain an improvement over fuzzy $k$-modes with all attributes (FKM), but their performance improvement is not better than that of our algorithm for all selected data sets. In the ccdByEnsemble algorithm, each clustering result is generated by only one of all attributes. It is quite obvious that single attribute is not enough to form good clustering result. Therefore, the performance of the ccdByEnsemble algorithm may be worse than that of the individual algorithm with all attributes (FKM). This claim is confirmed by the results reported in Fig. 6k–l. RDCE uses some of the attributes to partition the data. However, these attribute sets may not include essential attribute but irrelevant or redundant one because of the randomness in generating the attribute set. This phenomenon is exhibited by the performance of the RDCE on data sets “Soybean” and “TTT”. Instead of random subspaces, the proposed algorithm in this study uses some high quality rough subspaces to partition the data. Each rough subspace could preserve the clustering power of the original attribute set. Furthermore, different rough subspaces describe the internal granule structure of the data in different ways, which could be utilized by clustering ensemble algorithm to achieve a sound consensus solution.

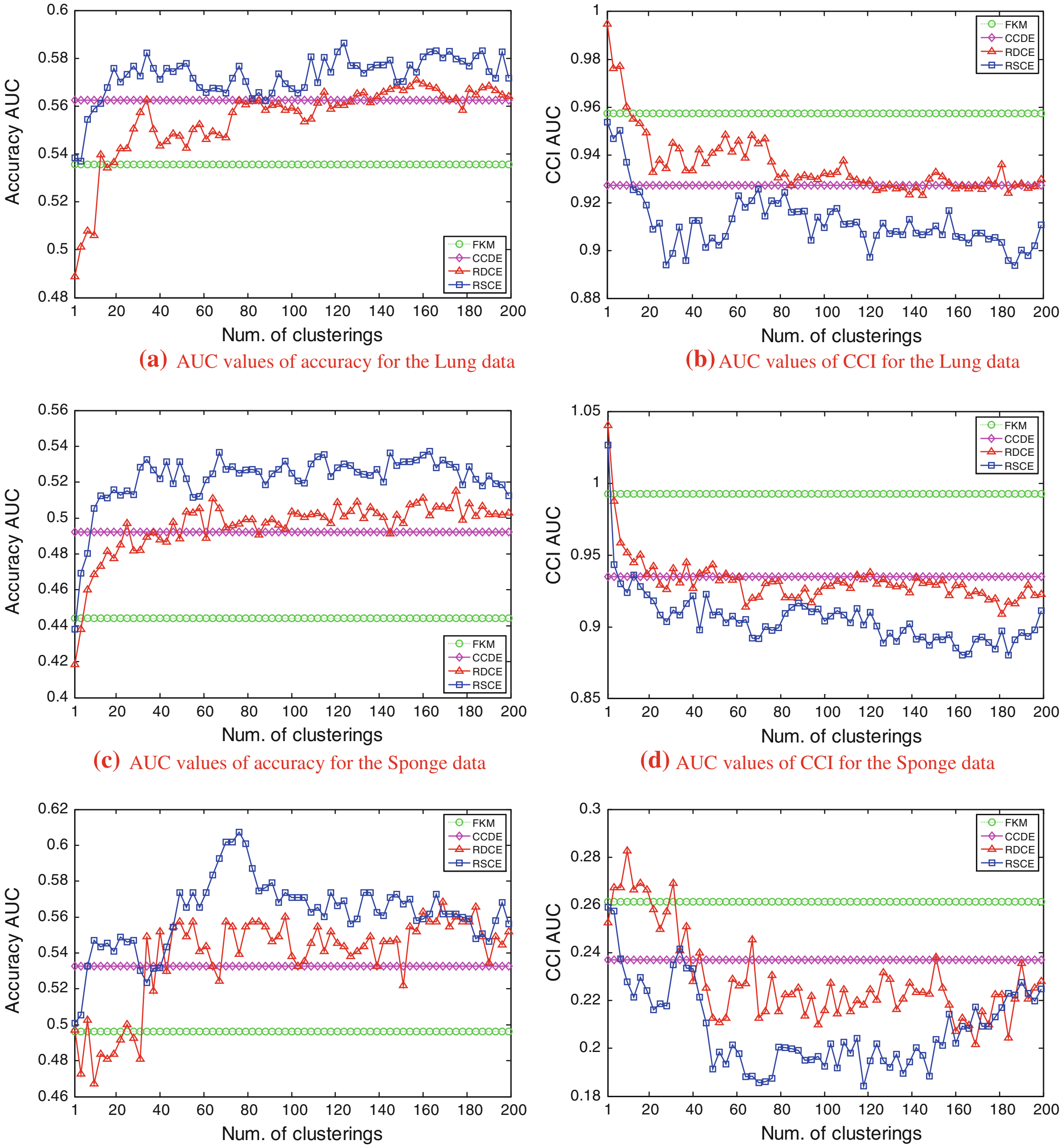
Since the CCI is an internal measure not resorting itself to class labels, its trend is more unstable than the one reported in terms of the accuracy measure. Nevertheless, it approximately reflects the accuracy of the selected algorithms. In Figs. 5 and 6, the accuracy of the proposed algorithm is better than that of the RDCE. The CCI curves in the figures also visualize this tendency. Note that good clustering comes with small CCI value.

In addition to Figs. 5 and 6, we also quantitatively examine the significance level of performance difference between the proposed algorithm and other compared algorithms. Given two algorithms $A$ and $B$, when the maximum number of clustering partitions and fuzzification coefficient are fixed, ten independent runs are performed for each algorithm. Therefore, two-tailed pairwise $t$ test is employed to evaluate the significance level of performance gap between the two algorithms. More specifically, the $p$ value returned by the two-tailed pairwise $t$ test is used as a measure for how much difference between the two algorithms’ performance. The smaller the $p$ value is, the higher the level of performance difference is. Generally speaking, a significant difference is deemed to occur if the returned $p$ value is less than 0.05 (i.e., $5.0e{-}2$).

Table 7 reports the win/tie/loss counts based on statistical tests. Under a specific fuzzification coefficient (i.e., $m$), a win (or loss) is counted for the data set (i.e., $p < 0.05$) when our proposed algorithm is significantly better (or worse) than the compared algorithm out of ten runs. Otherwise, a tie is recorded. In addition, the maximum, minimum, and average $p$ values across different fuzzification coefficients are also listed for reference purpose along with the win/tie/loss counts.

As shown in Table 7, it is clear that our proposed algorithm is superior or at least comparable to FKM, CCDE and RDCE in most cases. Furthermore, either FKM or CCDE seldom outperforms our proposed algorithm. To sum up, our proposed algorithm is statistically superior to FKM, CCDE and RDCE in around 94.3, 97.1 and 74.3 % cases, and is only inferior to RDCE in around 2.9 % cases.

In summary, the ensemble technique is beneficial to data clustering. The existing clustering ensemble methods are



**(a)** AUC values of accuracy for the Lung data

**(b)** AUC values of CCI for the Lung data

**(c)** AUC values of accuracy for the Sponge data

**(d)** AUC values of CCI for the Sponge data

**(e)** AUC values of accuracy for the Dermat data

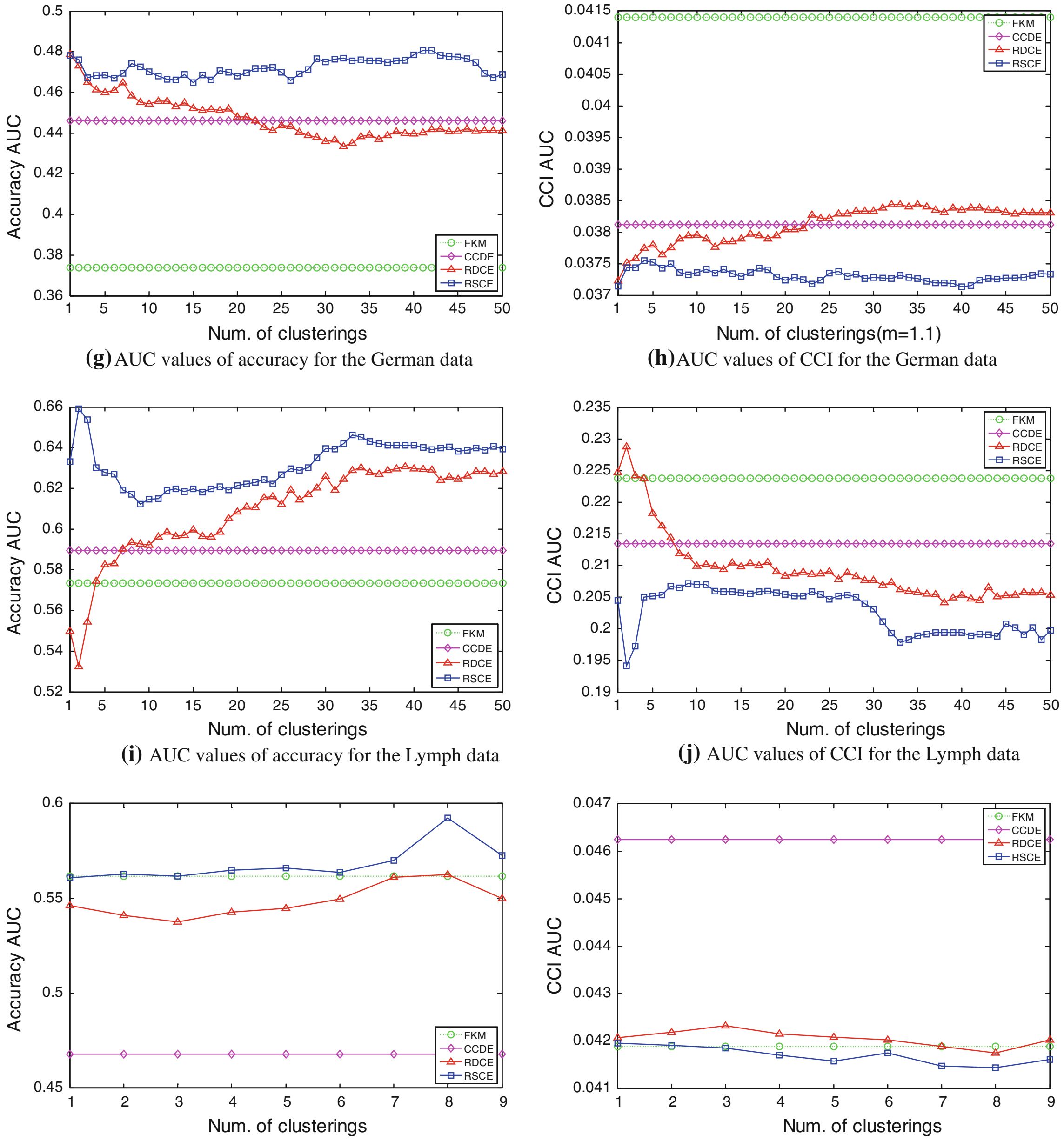
**(f)** AUC values of CCI for the Dermat data

**Fig. 6** The performance of the selected algorithms for UCI data sets. **a** AUC values of accuracy for the Lung data, **b** AUC values of CCI for the Lung data, **c** AUC values of accuracy for the Sponge data, **d** AUC values of CCI for the Sponge data, **e** AUC values of accuracy for the Dermat data, **f** AUC values of CCI for the Dermat data, **g** AUC values of accuracy for the German data, **h** AUC values of CCI for the German data, **i** AUC values of accuracy for the Lymph data, **j** AUC values of CCI for the Lymph data, **k** AUC values of accuracy for the TTT data, **l** AUC values of CCI for the TTT data

ineffective in dealing with categorical data, while the proposed clustering ensemble algorithm is shown to be able to boost the performance of fuzzy *k*-modes.

## 6 Conclusions and future work

Ensemble learning is an effective technique that uses multiple models to obtain better performance than which could have been obtained by running any of its constituent models. However, clustering ensemble has received less attention than individual clustering, especially for categorical data. In this paper, we have introduced a novel clustering ensemble algorithm for categorical data. The attribute spaces used in the proposed algorithm are rough subspaces of the data, which preserve the clustering power of the entire collection of the attributes. The algorithm introduced here only combines some high quality rough



**(g)** AUC values of accuracy for the German data

**(h)** AUC values of CCI for the German data

**(i)** AUC values of accuracy for the Lymph data

**(j)** AUC values of CCI for the Lymph data

**(k)** AUC values of accuracy for the TTT data

**(l)** AUC values of CCI for the TTT data

**Fig. 6** continued

**Table 7** Statistical test for RSCE against FKM, CCDE and RDCE under different fuzzification coefficients

| Data sets | FKM | | CCDE | | RDCE | |
|---|---|---|---|---|---|---|
| | Win/tie/loss | *p* value [min, max, avg.] | Win/tie/loss | *p* value [min, max, avg.] | Win/tie/loss | *p* value [min, max, avg.] |
| Lung | 10/0/0 | [1.0*e*−3, 4.1*e*−2, 1.6*e*−2] | 8/2/0 | [3.9*e*−3, 1.5*e*−1, 4.5*e*−2] | 4/6/0 | [7.8*e*−2, 8.6*e*−1, 5.6*e*−1] |
| Sponge | 10/0/0 | [1.5*e*−4, 2.3*e*−3, 7.1*e*−4] | 10/0/0 | [2.1*e*−3, 3.0*e*−2, 8.3*e*−3] | 9/1/0 | [6.1*e*−3, 7.3*e*−2, 3.6*e*−2] |
| Soybean | 10/0/0 | [3.8*e*−23, 2.3*e*−4, 4.1*e*−5] | 10/0/0 | [9.2*e*−22, 2.9*e*−3, 5.2*e*−4] | 10/0/0 | [3.2*e*−3, 4.7*e*−2, 2.9*e*−2] |
| Dermat | 10/0/0 | [4.7*e*−9, 1.2*e*−4, 7.3*e*−5] | 10/0/0 | [6.1*e*−5, 4.0*e*−3, 1.7*e*−4] | 10/0/0 | [2.5*e*−4, 3.7*e*−2, 8.3*e*−3] |
| German | 10/0/0 | [2.3*e*−6, 1.4*e*−4, 3.1*e*−5] | 10/0/0 | [7.5*e*−4, 1.4*e*−2, 5.0*e*−3] | 4/4/2 | [3.0*e*−2, 9.3*e*−1, 4.4*e*−1] |
| Lymph | 10/0/0 | [9.1*e*−6, 4.8*e*−3, 1.2*e*−3] | 10/0/0 | [5.3*e*−5, 1.1*e*−2, 2.2*e*−3] | 6/4/0 | [3.7*e*−2, 7.0*e*−1, 3.2*e*−1] |
| TTT | 6/4/0 | [3.4*e*−4, 3.1*e*−1, 8.1*e*−2] | 10/0/0 | [1.3*e*−5, 4.0*e*−2, 8.4*e*−3] | 9/1/0 | [6.3*e*−4, 1.4*e*−1, 3.2*e*−2] |

subspace-based partitions but the diversity of individual clustering results is retained as well. In this sense, the RSCE could benefit from diverse clustering partitions and generate a robust and stable solution. Furthermore, we have proposed a new cluster index for categorical data, which could approximately reflect the performance of the clustering algorithms. Empirical evidence shows that this ensemble method is promising in practice. Future work will focus on speeding up the process of searching rough subspaces and improving the strategy for choosing rough subspaces with high quality and diversity.

**Acknowledgments** The authors would like to thank the Editors for their kindly help and the anonymous referees for their valuable comments and helpful suggestions. Special thanks go to Ms. Ting Zhu for her assistance in revising the paper. The work is partially supported by the National Natural Science Foundation of China (Serial No. 60970061, 61075056, 61103067, 61202170, 61203247, 61273304), China Postdoctoral Science Foundation (Serial No. 2011M500626, 2011M500815) and Fundamental Research Funds for the Central Universities.

## References

Al-Razgan M, Domeniconi C, Barbara D (2008) Random subspace ensembles for clustering categorical data. SCI 126:31–48

Anderberg MR (1973) Cluster analysis for applications. Academic Press, New York

Ayad HG, Kamel MS (2008) Cumulative voting consensus method for partitions with variable number of clusters. IEEE Trans Pattern Anal Mach Intell 30(1):160–173

Ayad HG, Kamel MS (2010) On voting-based consensus of cluster ensembles. Pattern Recogn 43(5):1943–1953

Ball GH, Hall DJ (1967) A clustering technique for summarizing multivariate data. Behav Sci 12(2):153–155

Bargiela A, Pedrycz W (2005) A model of granular data: a design problem with the Tchebyschev FCM. Soft Comput 9(3):155–163

Bezdek JC (1981) Pattern recognition with fuzzy objective function algorithms. Kluwer Academic Publishers, Norwell

Domeniconi C, Al-Razgan M (2009) Weighted cluster ensembles: methods and analysis. ACM Trans Knowl Discov Data 2(4):1–40

Fern XZ, Brodley CE (2003) Random projection for high dimensional data clustering: A cluster ensemble approach. In: Proceedings of the 20th international conference on machine learning. pp 186–193

Fern XZ, Brodley CE (2004) Solving cluster ensemble problems by bipartite graph partitioning. In: Proceedings of the 21th international conference on machine learning. Banff, Alberta, Canada

Fischer B, Buhmann JM (2003) Bagging for path-based clustering. IEEE Trans Pattern Anal Mach Intell 25(11):1411–1415

Fred A, Jain AK (2005) Combining multiple clusterings using evidence accumulation. IEEE Trans Pattern Anal Mach Intell 27(6):835–850

Ghaemi R, Sulaiman MN, Ibrahim H et al (2009) A survey: clustering ensembles techniques. World Acad Sci Eng Technol 50:636–645

Gionis A, Mannila H, Tsaparas P (2007) Clustering aggregation. ACM Trans Knowl Discov Data 1(1):1–30

Hadjitodorov ST, Kuncheva LI, Todorova LP (2006) Moderate diversity for better cluster ensembles. Inf Fusion 7(3):264–275

He ZY, Xu XF, Deng SC (2005) A cluster ensemble method for clustering categorical data. Inf Fusion 6(2):143–151

Hong Y, Kwong S, Chang YC et al (2008a) Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm. Pattern Recogn 41(9):2742–2756

Hong Y, Kwong S, Chang YC et al (2008b) Consensus unsupervised feature ranking from multiple views. Pattern Recogn Lett 29(5):595–602

Hore P, Hall LO, Goldgof DB (2009) A scalable framework for cluster ensembles. Pattern Recogn 42(5):676–688

Huang ZX, Ng MK (1999) A fuzzy k-modes algorithm for clustering categorical data. IEEE Trans Fuzzy Syst 7(4):446–452

Iam-On N, Boongoen T, Garrett S et al (2011) A link-based approach to the cluster ensemble problem. IEEE Trans Pattern Anal Mach Intell 33(12):2396–2409

Iam-On N, Boongeon T, Garrett S et al (2012) A link-based cluster ensemble approach for categorical data clustering. IEEE Trans Knowl Data Eng 24(3):413–425

Jain AK (2010) Data clustering: 50 years beyond K-means. Pattern Recogn Lett 31(8):651–666

Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice-Hall, Inc., Upper Saddle River

Jia JH, Xiao X, Liu BX et al (2011) Bagging-based spectral clustering ensemble selection. Pattern Recogn Lett 32(10):1456–1467

Jiang Y, Zhou Z-H (2004) SOM ensemble-based image segmentation. Neural Process Lett 20(3):171–178

Kuncheva LI, Vetrov DP (2006) Evaluation of stability of k-means cluster ensembles with respect to random initialization. IEEE Trans Pattern Anal Mach Intell 28(11):1798–1808

Lange T, Buhmann JM (2005) Combining partitions by probabilistic label aggregation. In: Proceedings of the 11th ACM SIGKDD international conference on knowledge discovery in data mining. pp 147–156

Li TY, Chen Y (2010) Fuzzy clustering ensemble with selection of number of clusters. J Comput 5(7):1112–1118

Li T, Ogihara M, Ma S (2010) On combining multiple clusterings: an overview and a new perspective. Appl Intell 33(2):207–219

Liu Q (2001) Rough sets and rough reasoning. Science Press, Beijing (in Chinese)

Luo HL, Jing FR, Xie XB (2006) Combining multiple clusterings using information theory based genetic algorithm. In: Proceedings of the 2006 international conference on computational intelligence and security. pp 84–89

Miao DQ, Li DG (2008) Rough sets theory, algorithms and applications. Tsinghua University Press, Beijing (in Chinese)

Miao DQ, Zhao Y, Yao YY et al (2009) Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model. Inf Sci 179(24):4140–4150

Minaei-Bidgoli B, Topchy A, Punch W (2004) A comparison of resampling methods for clustering ensembles. In: Proceedings of the international conference on artificial intelligence (IC-AI'04). pp 939–945

Monti S, Tamayo P, Mesirov J, Golub T (2003) Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Mach Learn 52(1–2):91–118

Øhrn A, Komorowski J (1997) ROSETTA: a rough set toolkit for analysis of data. In: Proceedings of the 3rd international joint conference on information sciences and 5th international workshop on rough sets and soft computing (RSSC'97), Durham, NC, USA, March. pp 403–407

Pawlak Z (1982) Rough sets. Int J Comput Inf Sci 11(5):341–356

Pawlak Z (1991) Rough sets: theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht

Pedrycz W (1996) Conditional fuzzy C-means. Pattern Recogn Lett 17(6):625–632

Pedrycz W (2005) Knowledge based clustering: From data to information granules. Wiley, Hoboken

Pedrycz W, Loia V, Senatore S (2010) Fuzzy clustering with viewpoints. IEEE Trans Fuzzy Syst 18(2):274–284

Rokach L (2010) Ensemble-based classifiers. Artif Intell Rev 33(1–2):1–39

Strehl A, Ghosh J (2002) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J Mach Learn Res 3:583–617

Thangavel K, Pethalakshmi A (2009) Dimensionality reduction based on rough set theory: a review. Appl Soft Comput 9(1):1–12

Topchy A, Jain AK, Punch W (2005) Clustering ensembles: models of consensus and weak partitions. IEEE Trans Pattern Anal Mach Intell 27(12):1866–1881

Tumer K, Agogino AK (2008) Ensemble clustering with voting active clusters. Pattern Recogn Lett 29(14):1947–1953

Vega-Pons S, Ruiz-Shulcloper J (2011) A survey of clustering ensemble algorithms. Int J Pattern Recognit Artif Intell 25(3): 337–372

Wang GY (2001) Rough sets theory and knowledge acquisition. Xi'an Jiaotong University Press, Xi'an (in Chinese)

Wang JY, Gao C (2009) An improved algorithm for attribute reduction based on discernibility matrix. Comput Eng 35(3): 66–68 (in Chinese)

Wang WN, Zhang YJ (2007) On fuzzy cluster validity indices. Fuzzy Sets Syst 158(19):2095–2117

Yu ZW, Wong H-S (2009) Class discovery from gene expression data based on perturbation and cluster ensemble. IEEE Trans Nanobiosci 8(2):147–160

Yu ZW, Wong H-S, Wang HQ (2007a) Graph-based consensus clustering for class discovery from gene expression data. Bioinformatics 23(21):2888–2896

Yu ZW, Zhang SH, Wong H-S, et al (2007) Image segmentation based on cluster ensemble. In: Proceedings of the 4th international symposium on neural networks: advances in neural networks, part III. Springer, Berlin, pp 894–903

Yu ZW, Deng ZK, Wong H-S, et al (2008) Fuzzy cluster ensemble and its application on 3D head model classification. In: Proceedings of the IEEE international joint conference on neural networks (IJCNN 2008). pp 569–576

Yu ZW, Wong H-S, You J et al (2011) Knowledge based cluster ensemble for cancer discovery from biomolecular data. IEEE Trans Nanobiosci 10(2):76–85

Yu ZW, Wong H-S, You J et al (2012a) Hybrid cluster ensemble framework based on the random combination of data transformation operators. Pattern Recogn 45(5):1826–1837

Yu ZW, You J, Wong H-S et al (2012b) From cluster ensemble to structure ensemble. Inf Sci 198:81–99

Zhang WX, Wu WZ, Liang JY et al (2001) Rough sets theory and methods. Science Press, Beijing (in Chinese)

Zhang XR, Jiao LC, Liu F et al (2008) Spectral clustering ensemble applied to SAR image segmentation. IEEE Trans Geosci Remote Sens 46(7):2126–2136

Zhou ZH, Wu JX, Tang W (2002) Ensembling neural networks: many could be better than all. Artif Intell 137(1–2):239–263

Zhou J, Miao DQ, Pedrycz W et al (2011) Analysis of alternative objective functions for attribute reduction in complete decision tables. Soft Comput 15(8):1601–1616