Pattern Recognition 44 (2011) 2843–2861

Contents lists available at ScienceDirect

Pattern Recognition

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

A novel attribute weighting algorithm for clustering high-dimensional categorical data

Liang Bai [a,b], Jiye Liang [a,*], Chuangyin Dang [b], Fuyuan Cao [a]

[a] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan, 030006 Shanxi, China

[b] Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:
Received 5 October 2010
Received in revised form
23 April 2011
Accepted 26 April 2011
Available online 10 May 2011

Keywords:
Cluster analysis
Optimization algorithm
High-dimensional categorical data
Subspace clustering
Attribute weighting

ABSTRACT

Due to data sparseness and attribute redundancy in high-dimensional data, clusters of objects often exist in subspaces rather than in the entire space. To effectively address this issue, this paper presents a new optimization algorithm for clustering high-dimensional categorical data, which is an extension of the *k*-modes clustering algorithm. In the proposed algorithm, a novel weighting technique for categorical data is developed to calculate two weights for each attribute (or dimension) in each cluster and use the weight values to identify the subsets of important attributes that categorize different clusters. The convergence of the algorithm under an optimization framework is proved. The performance and scalability of the algorithm is evaluated experimentally on both synthetic and real data sets. The experimental studies show that the proposed algorithm is effective in clustering categorical data sets and also scalable to large data sets owning to its linear time complexity with respect to the number of data objects, attributes or clusters.

1. Introduction

Cluster analysis is a branch in statistical multivariate analysis and unsupervised machine learning, which has extensive applications in various domains, including financial fraud, medical diagnosis, image processing, information retrieval, bioinformatics. Clustering is a process of grouping a set of objects into clusters so that the objects in the same cluster have high similarity but are very dissimilar with objects in other clusters. Various types of clustering methods have been developed in the literature (e.g., [1]). While clustering of numeric data is extensively studied, recently increasing attention has been paid to clustering categorical data [2–10], where records are made up of non-numerical data, since this task is of great practical relevance in several fields ranging from statistics to psychology.

There are a number of challenges in clustering categorical data. First, the lack of an inherent order on the domains of the individual attributes prevents the definition of a notion of similarity, which catches resemblance between categorical data objects. Clearly, this imposes difficulties at devising a suitable clustering quality that are not encountered in the case of numeric attributes, where, instead, object similarity naturally follows from the geometric properties of the data. This means that the techniques used in clustering numerical data are not applicable to categorical data. Currently, several dissimilarity measures for categorical data have been proposed in the literature, see, for instance, [11]. Among them, the simple matching dissimilarity measure [6] is widely used for its low computational burden. Furthermore, categorical data are often high-dimensional such as market-basket and Web usage data and so on. Records in such data sets include a large number of attributes, typically with Boolean values. Several emerging application settings require clustering techniques that provide an effective and efficient treatment of this kind of data, such as text analysis, bioinformatics, e-commerce, astronomy, and the insurance industry [12,13].

Unfortunately, conventional clustering techniques fall short when clustering is performed in high-dimensional spaces [14]. For example, due to data sparseness or skewness, as well as attribute irrelevancy or redundancy in high-dimensional data, as the increase of the dimension cardinality, the dissimilarity between a given object x and its nearest object will be close to the dissimilarity between x and its farthest object. While the loss of the dissimilarity discrimination in high dimensions, discovering meaningful, separable clusters will be very challenging, if not impossible. Moreover, an interesting cluster usually occurs in a subspace defined by a subset of the initially selected attributes. To find the cluster, it is important to identify the subset of attributes. However, conventional clustering algorithms cannot select

* Corresponding author.
E-mail addresses: sxbailiang@126.com (L. Bai), ljy@sxu.edu.cn (J. Liang), mecdang@cityu.edu.hk (C. Dang), cfy@sxu.edu.cn (F. Cao).

0031-3203/$ - see front matter Crown Copyright © 2011 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.04.024

attributes automatically because they treat all attributes equally in the clustering process. A common approach to cope with the curse of dimensionality for mining tasks is to reduce the data dimensionality by using techniques of feature transformation and feature selection [13]. The feature transformation techniques, such as principal component analysis (PCA) and singular value decomposition (SVD), summarize the data in a fewer set of dimensions derived from the combinations of the original data attributes. However, the transformed features/dimensions have no intuitive meaning any more and thus the resulting clusters are hard to interpret and analyze [15]. On the other hand, the feature selection methods [16–19] reduce the data dimensionality by trying to select the most relevant attributes from the original data attributes. In such a way, only a particular subspace is selected to discover clusters. However, in many real data sets, clusters may be embedded in varying subspaces, and thus in the feature selection approaches the information of data objects clustered differently in varying subspaces is lost [15].

To tackle the above problem of clustering high-dimensional data sets, several clustering techniques have been developed in the literature, such as bi-clustering and subspace clustering. Bi-clustering [20,21] is a methodology where rows and columns are clustered simultaneously, as an alternative to dimensionality reduction techniques. The bi-clustering algorithms measure not only the dissimilarity between rows but also the dissimilarity between columns. The applied dissimilarity measures are similar to the simple matching measure. However, the bi-clustering algorithms are mainly applied to cluster binary data. When these algorithms are used to tackle categorical data [22], multiple category attributes need be converted into binary attributes (using 0 and 1 to represent either a category absent or present). If it is used in data mining, these algorithms need to handle a large number of binary attributes because data sets in data mining often have categorical attributes with hundreds or thousands of categories. This will inevitably increase both computational and space costs. This is also an important reason for designing algorithms to directly cluster categorical data.

Subspace clustering is a very effective method for clustering high-dimensional numerical or categorical data. Its goal is to locate clusters in different subspaces of the same data set. In general, a subspace cluster represents not only the cluster itself, but also the subspace where the cluster is situated. The two main categories of subspace clustering algorithms are hard subspace clustering and soft subspace clustering. Hard subspace clustering methods are for the clustering of high-dimensional data. This kind of subspace clustering algorithm identifies exact subspaces for different clusters. For numerical data, such methods include CLIQUE [12], ENCLUS [23], MAFIA [24], PROCLUS [25], ORCLUS [26], FINDIT [27], DOC [28], d-Clusters [29], HARP [30] and LDR [31] and so on. A detailed review of hard subspace clustering algorithms for numerical data can be found in [13]. Although most of these approaches were defined for numerical data, some recent studies [32–34] also consider hard subspace clustering for categorical data. For example, Gan [32] proposed the SUBCAD algorithm for subspace clustering high-dimensional categorical data set which iteratively finds an approximation of the optimal partition. However, the algorithm cannot guarantee that the value of the objective function decreases strictly in the iterative process. Zaki [33] presented the CLICKS algorithm which encodes a data set into a weighted graph structure. This algorithm starts from the observation that clusters correspond to dense maximal k-partite cliques and proceeds by enumerating all maximal k-partite cliques and checking their frequency. A crucial step is the computation of strongly connected components, that is, pairs of attribute values whose co-occurrence is above a specified threshold. For large values of m (or, more generally, when the number of attributes or the cardinality of each attribute is high), this is an expensive task, which makes the approach inefficient. In addition, the technique depends upon a set of parameters, whose tuning can be problematic in practical applications.

While the exact subspaces are identified in hard subspace clustering, a weight is assigned to each attribute in the clustering process of soft subspace clustering to measure the contribution of each attribute to the formation of a particular cluster. In the clustering procedure, each attribute contributes differently to every cluster. The subspaces of different clusters can be identified by the values of weights after clustering. Soft subspace clustering can be considered as an extension of the conventional attribute weighting clustering [35–37] which employs a common weight vector for the whole data set in the clustering procedure. However, it is also distinct in that different weight vectors are assigned to different clusters. From this perspective, soft subspace clustering may thus be referred to as multiple attribute weighting clustering. Soft subspace clustering has recently emerged as a hot research topic, and many algorithms have been reported [38–45]. Among them, the k-means-type attribute weighting methods are well known for their efficiency in clustering large data sets. Their computational complexity is linear with respect to either the number of data objects, attributes or clusters. The representative methods include Chan's weighting method [38] and Jing's weighting method [42] and so on. Chan used a similar optimization problem for the fuzzy clustering proposed by Bezdek [46] to automatically compute weights for all attributes in a cluster. Jing computed weights for all attributes in a cluster by using an optimization method based on maximal entropy which was proposed by Miyamoto [47]. However, as regard to the convergence of the maximal entropy optimization method, there are some disputes in the academic circles [48,49]. A detailed review of these soft subspace clustering algorithms for numerical data can be found in [50].

Although these soft subspace clustering algorithms have been developed and applied to different areas, the use of these algorithms is only limited to numeric data. Huang [6,7] proposed the k-modes algorithm which extends the k-means algorithm by using a simple matching dissimilarity measure for categorical objects instead of Euclidean distance measure, modes instead of means for clusters, and a frequency-based method to update modes in the clustering process to minimize the clustering objective function. These extensions have removed the numeric-only limitation of the k-means algorithm and enable the k-means clustering process to be used to efficiently cluster large categorical data sets from real world databases. Chan [38] presented a soft subspace clustering algorithm which uses the k-modes paradigm to deal with categorical data. However, the method encounters some problems in handling categorical data. The simple matching dissimilarity measure computes the distance by comparing their categorical values in each attribute. If the two categorical values are identical, then the difference is 0 and otherwise 1. Due to the fact that 0 multiplied by any weight value is still 0, these weighting methods will not work when the comparative result is 0. Moreover, for different comparative results of the two categorical values, the weight values should be different. While the comparative result is 1, the corresponding weight value should be inversely proportional to the dispersion in the attribute of the cluster. In this case, the larger the weighting value is, the larger the dissimilarity between the object and the cluster center in the attribute is. Conversely, while the comparative result is 0, the corresponding weight value should be proportional to the dispersion in the attribute of the cluster. In this case, the smaller the weighting value is, the larger the similarity between the object and the cluster center in the attribute is. Therefore, motivated by this idea, we extend the k-modes algorithm to propose an optimal attribute weighting

algorithm for high-dimensional categorical data. The major contributions of this paper are as follows:

- A new weighted dissimilarity measure is proposed, which is applied to the k-modes algorithm. The updating formulas of the k-modes clustering algorithm with the new weighted dissimilarity measure are derived. The convergence of the algorithm under an optimization framework is proved.
- The new dissimilarity measure is integrated with Chan's weighted dissimilarity measure to form a mixed weighted dissimilarity measure. Based on the mixed dissimilarity measure, a mixed attribute weighting algorithm is proposed to cluster high-dimensional categorical data. The convergence of the proposed algorithm under an optimization framework is proved.
- The performance and scalability of the mixed attribute weighting algorithm is investigated by using both synthetic and real data sets.

The rest of this paper is organized as follows. A detailed review of the k-modes algorithm and the weighting k-modes algorithm is presented in Sections 2 and 3, respectively. In Section 4, a new weighted dissimilarity measure is presented and analyzed. In Section 5, a mixed attribute weighting algorithm is proposed. Section 6 illustrates the performance and scalability of the proposed algorithm. Finally, a concluding remark is given in Section 7.

2. The k-modes algorithm

As we know, the structural data are stored in a table, where each row(tuple) represents facts about an object. A data table is also called an information system in rough set theory [51–53]. Data in the real world usually contain categorical attributes [54]. More formally, a categorical data table is defined as a quadruple $IS=(U,A,V,f)$, where:

(1) $U=\{x_1,x_2,\ldots,x_n\}$ is a nonempty set of n data points, called a universe.
(2) $A=\{a_1,a_2,\ldots,a_m\}$ is a nonempty set of m categorical attributes.
(3) V is the union of attribute domains, i.e., $V=\bigcup_{j=1}^{m} V_{a_j}$, where $V_{a_j}=\{a_j^{(1)},a_j^{(2)},\ldots,a_j^{(n_j)}\}$ is the value domain of categorical attribute a_j and is finite and unordered, e.g., for any $1\le p\le q\le n_j$, either $a_j^{(p)}=a_j^{(q)}$ or $a_j^{(p)}\neq a_j^{(q)}$. Here, n_j is the number of categories of attribute a_j for $1\le j\le m$.
(4) $f: R\times A\to V$ is an information function such that $f(x_i,a_j)\in V_{a_j}$ for $1\le i\le n$ and $1\le j\le m$, where $R=V_{a_1}\times V_{a_2}\times\cdots\times V_{a_m}$ and $U\subseteq R$.

The k-modes algorithm uses the k-means paradigm to cluster categorical data. The objective of clustering a set of n categorical objects into k clusters is to find W and Z that minimize [7]

$$F(W,Z)=\sum_{l=1}^{k}\sum_{i=1}^{n} w_{li}d(z_l,x_i) \tag{1}$$

subject to

$$\begin{cases} w_{li}\in\{0,1\}, & 1\le l\le k,\ 1\le i\le n,\\ \sum_{l=1}^{k} w_{li}=1, & 1\le i\le n,\\ 0<\sum_{i=1}^{n} w_{li}<n, & 1\le l\le k, \end{cases} \tag{2}$$

where

- n is the number of objects in U, $k(\le n)$ is a known number of clusters;
- $W=[w_{li}]$ is a k-by-n $\{0,1\}$ matrix, w_{li} is a binary variable, and indicates whether object x_i belongs to the lth cluster, $w_{li}=1$ if x_i belongs to the lth cluster and 0 otherwise;
- $Z=[z_1,z_2,\ldots,z_k]$ and $z_l=[f(z_l,a_1),f(z_l,a_2),\ldots,f(z_l,a_m)]$ is the lth cluster center with categorical attributes $a_1,a_2,\ldots,a_m$;
- $d(z_l,x_i)$ is a distance or dissimilarity measure between object x_i and the center z_l of the lth cluster which is defined as

$$d(z_l,x_i)=\sum_{j=1}^{m}\delta_{a_j}(z_l,x_i), \tag{3}$$

where

$$\delta_{a_j}(z_l,x_i)=\begin{cases}1, & f(z_l,a_j)\neq f(x_i,a_j),\\ 0, & f(z_l,a_j)=f(x_i,a_j).\end{cases} \tag{4}$$

The minimization of F in (1) with the constraints in (2) forms a class of constrained nonlinear optimization problems whose solutions are unknown. The usual method towards optimization of F in (1) is to use partial optimization for Z and W. In this method, we first fix Z and find necessary conditions on W to minimize F. Then, we fix W and minimize F with respect to Z. The above optimization problem can be solved by iteratively solving the following two minimization problems:

1. Problem P_1: Fix $Z=\hat{Z}$, solve the reduced problem $F(W,\hat{Z})$;
2. Problem P_2: Fix $W=\hat{W}$, solve the reduced problem $F(\hat{W},Z)$.

Problem P_1 is solved by

$$\hat{w}_{li}=\begin{cases}1 & \text{if } d(\hat{z}_l,x_i)\le d(\hat{z}_h,x_i),\ 1\le h\le k,\\ 0 & \text{otherwise}\end{cases} \tag{5}$$

for $1\le i\le n$, $1\le l\le k$.

Problem P_2 is solved by

$$f(z_l,a_j)=a_j^{(r)}\in V_{a_j} \tag{6}$$

where

$$|\{w_{li}|f(x_i,a_j)=a_j^{(r)},w_{li}=1\}|\ge|\{w_{li}|f(x_i,a_j)=a_j^{(t)},w_{li}=1\}|,\quad 1\le t\le n_j \tag{7}$$

for $1\le j\le m$. Here, $|X|$ denotes the number of elements in the set X, $V_{a_j}=\{a_j^{(1)},a_j^{(2)},\ldots,a_j^{(n_j)}\}$, n_j is the number of categories of attribute a_j for $1\le j\le m$.

This process is formalized in the k-modes algorithm as follows [7]:

Step 1. Choose an initial point set $Z^{(1)}\subseteq R$. Determine $W^{(1)}$ such that $F(W,Z^{(1)})$ is minimized. Set $t=1$.

Step 2. Determine $Z^{(t+1)}$ such that $F(W^{(t)},Z^{(t+1)})$ is minimized. If $F(W^{(t)},Z^{(t+1)})=F(W^{(t)},Z^{(t)})$, then stop; otherwise goto Step 3.

Step 3. Determine $W^{(t+1)}$ such that $F(W^{(t+1)},Z^{(t+1)})$ is minimized. If $F(W^{(t+1)},Z^{(t+1)})=F(W^{(t)},Z^{(t+1)})$, then stop; otherwise set $t=t+1$ and goto Step 2.

One of the drawback of the k-modes algorithm is that they treat all attributes equally in deciding the cluster memberships of objects in Theorem 1. This is undesirable in many applications such as data mining where data often contain a large number of sparse or redundancy attributes. A cluster in a given data set is often confined to a subset of attributes rather than the entire attribute set. Inclusion of other attributes can only obscure the discovery of the cluster by a clustering algorithm.

3. The weighting *k*-modes algorithm

Chan [38] proposed a soft subspace clustering algorithm for numerical and categorical data which extends the *k*-means clustering process to automatically calculate a weight for each attribute in each cluster. When the algorithm is used to deal with categorical data, the algorithm is viewed as a weighting *k*-modes algorithm (WKM). The objective function of WKM is defined as

$$F_1(W,Z,\Lambda)=\sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} w_{li}\lambda_{lj}^{\beta}\delta_{a_j}(z_l,x_i) \tag{8}$$

subject to the constraints in (2) and

$$\begin{cases} \lambda_{lj}\in[0,1], & 1\le l\le k,\ 1\le j\le m,\\ \sum_{j=1}^{m}\lambda_{lj}=1, & 1\le l\le k, \end{cases} \tag{9}$$

where $\Lambda=[\lambda_{lj}]$ is a *k*-by-*n* [0, 1] matrix, λ_{lj} is the weight for the *j*th attribute in the *l*th cluster and $\beta\in(1,+\infty)$ is a parameter for controlling attribute weight λ_{lj}.

Similar to solving (1), the objective function (8) can be locally minimized by iteratively solving the following three minimization problems:

1. Problem P_1: Fix $Z=\hat{Z}$ and $\Lambda=\hat{\Lambda}$, solve the reduced problem $F_1(W,\hat{Z},\hat{\Lambda})$;
2. Problem P_2: Fix $W=\hat{W}$ and $\Lambda=\hat{\Lambda}$, solve the reduced problem $F_1(\hat{W},Z,\hat{\Lambda})$;
3. Problem P_3: Fix $W=\hat{W}$ and $Z=\hat{Z}$, solve the reduced problem $F_1(\hat{W},\hat{Z},\Lambda)$.

Problem P_1 is solved by

$$\hat{w}_{li}=\begin{cases} 1 & \text{if } \sum_{j=1}^{m}\lambda_{lj}^{\beta}\delta_{a_j}(\hat{z}_l,x_i)\le\sum_{j=1}^{m}\lambda_{lj}^{\beta}\delta_{a_j}(\hat{z}_h,x_i),\ 1\le h\le k,\\ 0 & \text{otherwise}\end{cases} \tag{10}$$

for $1\le i\le n$, $1\le l\le k$, and problem P_2 is solved in (6) and (7). The solution to problem P_3 is given by

$$\hat{\lambda}_{lj}=\begin{cases} \dfrac{1}{m_i} & \text{if } D_{lj}=0 \text{ and } m_i=|\{t: D_{lt}(\hat{z}_l,x_i)=0\}|,\\ 0 & \text{if } D_{lj}\neq 0 \text{ but } D_{lt}=0 \text{ for some } t,\\ \dfrac{1}{\sum_{h=1}^{k}\left[\frac{D_{lj}}{D_{lh}}\right]^{1/(\beta-1)}} & \text{if } D_{lj}\neq 0 \text{ and } D_{lh}\neq 0,\ 1\le h\le m\end{cases} \tag{11}$$

for $1\le j\le m$, $1\le l\le k$, where

$$D_{lj}=\sum_{i=1}^{n} w_{li}\delta_{a_j}(\hat{z}_l,x_i) \tag{12}$$

for $1\le j\le m$, $1\le l\le k$.

The above procedure is formalized in the algorithm as follows [38]:

Step 1: Choose an initial point set $Z^{(1)}\subseteq R$ and set $\Lambda^{(1)}$ be a *k*-by-*m* matrix with all the entries being equal to $1/m$. Determine $W^{(1)}$ such that $F_1(W,Z^{(1)},\Lambda^{(1)})$ is minimized. Set $t=1$.

Step 2: Determine $Z^{(t+1)}$ such that $F_1(W^{(t)},Z^{(t+1)},\Lambda^{(t)})$ is minimized. If $F_1(W^{(t)},Z^{(t+1)},\Lambda^{(t)})=F_1(W^{(t)},Z^{(t)},\Lambda^{(t)})$, then stop; otherwise goto Step 3.

Step 3: Determine $\Lambda^{(t+1)}$ such that $F_1(W^{(t)},Z^{(t+1)},\Lambda^{(t+1)})$ is minimized. If $F_1(W^{(t)},Z^{(t+1)},\Lambda^{(t+1)})=F_1(W^{(t)},Z^{(t+1)},\Lambda^{(t)})$, then stop; otherwise goto Step 4.

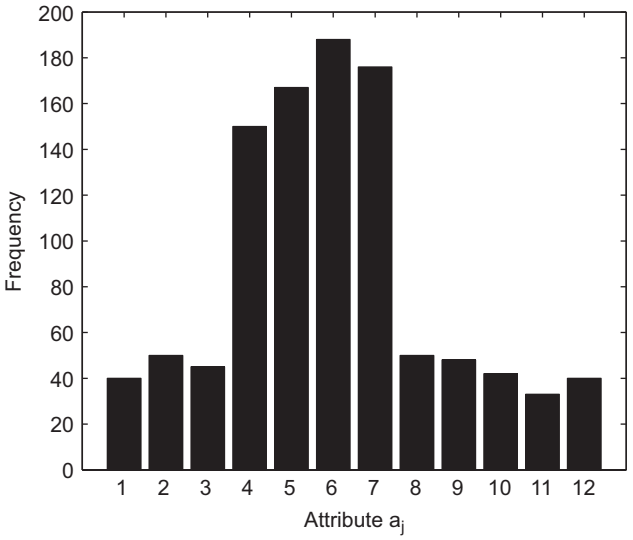

Fig. 1. An example of the frequency of each attribute value of its mode in a cluster, where each bar corresponds to each attribute.

Step 4: Determine $W^{(t+1)}$ such that $F(W^{(t+1)},Z^{(t+1)},\Lambda^{(t+1)})$ is minimized. If $F(W^{(t+1)},Z^{(t+1)},\Lambda^{(t+1)})=F(W^{(t)},Z^{(t+1)},\Lambda^{(t+1)})$, then stop; otherwise set $t=t+1$ and goto Step 2.

In the algorithm, given a data partition, the principal for attribute weighting is to assign a larger weight to an attribute that has a smaller sum of the within cluster distances and a smaller one to an attribute that has a larger sum of the within cluster distances. This principal conforms to reality. In the *k*-modes algorithm, a cluster is represented by "mode", which is composed of the most frequent attribute value in each attribute domain in this cluster. For an attribute, if the most frequent one of its attribute values in the cluster is very low, the dispersion of the objects from the mode in the attribute of the cluster is very large. This also means that the mode in the attribute has very weak representability in the cluster. Therefore, it is thought that the smaller the sum of the within cluster distances in the attribute is, the more important role the attribute plays in identifying the cluster. For example, Fig. 1 shows the frequency of each attribute value of its mode in a cluster. We see that the mode values in the attributes a_4, a_5, a_6 and a_7 have higher frequencies than other attributes. This indicates that the attributes a_4, a_5, a_6 and a_7 have more representability than other attributes in the cluster. Therefore, the attributes a_4, a_5, a_6 and a_7 are thought to be important for identifying the cluster.

According to (3) and (4), we remark that the dissimilarity between an object and a cluster center in a categorical attribute only has two values, 0 and 1, which is different from numerical attribute. When they have an identical value in the categorical attribute, the dissimilarity is 0, otherwise 1. The weighted dissimilarity measure between object x_i and the center z_l is defined as follows:

$$d_w(z_l,x_i)=\sum_{j=1}^{m}\psi_{a_j}(z_l,x_i), \tag{13}$$

where

$$\psi_{a_j}(z_l,x_i)=\begin{cases}\lambda_{lj}^{\beta}, & f(z_l,a_j)\neq f(x_i,a_j),\\ 0, & f(z_l,a_j)=f(x_i,a_j).\end{cases} \tag{14}$$

We remark that the weight λ_{lj} does not work while $f(z_l,a_j)=f(x_i,a_j)$. Let us consider the following example to

Table 1
An example data set.

Objects	Attributes	
	a_1	a_2
x_1	A	A
x_2	A	A
x_3	A	B
x_4	B	C
x_5	A	D
The mode z_l of the set	A	A

demonstrate the problem. The example data set in Table 1 is described with two categorical attributes a_1 and a_2.

When the dissimilarity between the object x_1 and the mode z_l is measured by using $d_w(.,.)$, we find that $\psi_{a_1}(z_l,x_1)=\psi_{a_2}(z_l,x_1)=0$. This means that $d_w(.,.)$ cannot discriminate the weights of a_1 and a_2 which are treated equally in the above situation. However, we can observe from (11) that the weight value of an attribute in a cluster is inversely proportional to the dispersion of the values from the center in the attribute of the cluster. Since the dispersions are different in different attributes of different clusters, the weight values for different clusters are different. The high weight indicates a small dispersion in the attribute of the cluster. Therefore, we can see that since the frequency of the attribute value 'A' in a_1 is higher than the attribute value 'A' in a_2, the dispersion in a_1 is smaller than that in a_2, which means that a_1 is more important to identify the cluster than a_2. However, if the value of the object in an attribute is the same as the center of the cluster, their dissimilarity in the attribute is 0, which makes the importance of attributes not reflected. When we replace (4) with the following equation

$$\delta'_{a_j}(z_l,x_i)=\begin{cases}1, & f(z_l,a_j)\neq f(x_i,a_j),\\ c, & f(z_l,a_j)=f(x_i,a_j),\end{cases} \tag{15}$$

where c is a constant, $c<1$ and $c\neq 0$, (15) removes the effect of 0 on the weights of attributes. Eq. (14) is changed as follows:

$$\psi'_{a_j}(z_l,x_i)=\begin{cases}\lambda_{lj}^{\beta}, & f(z_l,a_j)\neq f(x_i,a_j),\\ \lambda_{lj}^{\beta}c, & f(z_l,a_j)=f(x_i,a_j).\end{cases} \tag{16}$$

Eq. (12) is also changed as

$$\begin{aligned}D'_{lj}&=\sum_{i=1}^{n}w_{li}\delta'_{a_j}(z_l,x_i)=\sum_{i=1,f(z_l,a_j)\neq f(x_i,a_j)}^{n}w_{li}+c\sum_{i=1,f(z_l,a_j)=f(x_i,a_j)}^{n}w_{li}\\&=|c_l|-(1-c)|c_{lj}|\end{aligned} \tag{17}$$

for $1\le j\le m$, $1\le l\le k$. Here, $|c_l|$ is the number of objects in the lth cluster, given by

$$|c_l|=|\{w_{li}|w_{li}=1,1\le i\le n\}|, \tag{18}$$

and $|c_{lj}|$ is the number of objects with category $f(z_l,a_j)$ of the jth attribute in the lth cluster, given by

$$|c_{lj}|=|\{w_{li}|f(x_i,a_j)=f(\hat{z}_l,a_j),\quad w_{li}=1,1\le i\le n\}|. \tag{19}$$

Let us consider the above example again. We find that the method also has brought some new problems, which are described as follows:

(1) $c>0$. In this case, according to (11) and (17), we see that $D'_{lj}>0$ and D'_{lj} is inversely proportional to λ_{lj} and $|c_{lj}|$. We calculate $\psi'_{a_1}(z_l,x_1)=\lambda_{l1}^{\beta}c$ and $\psi'_{a_2}(z_l,x_1)=\lambda_{l2}^{\beta}c$. By (11), we know that $\lambda_{l1}^{\beta}>\lambda_{l2}^{\beta}$. Therefore, we have $\psi'_{a_1}(z_l,x_1)>\psi'_{a_2}(z_l,x_1)$ which illustrates that the similarity between the object x_1 and the center z_l in the attribute a_1 is less than that in the attribute a_2. This conclusion is contradictory to the original idea of attribute weighting.

(2) $c<0$. In this case, we cannot guarantee that D'_{lj} is always positive, which means that λ_{lj} in (11) might be negative for some data objects. This conclusion is inconsistent with the constraint conditions in (9).

From the above analysis, we think that a weighted dissimilarity measure for categorical data should have the following properties:

(1) if $f(x_i,a_h)=f(z_l,a_h)$ and $f(x_i,a_j)\neq f(z_l,a_j)$, then the dissimilarity between the object x_i and the center z_l in the attribute a_j is not less than that in the attribute a_h.

(2) if $|c_{lh}|>|c_{lj}|$, $f(x_i,a_h)\neq f(z_l,a_h)$ and $f(x_i,a_j)\neq f(z_l,a_j)$, then the dissimilarity between the object x_i and the center z_l in the attribute a_h is more than that in the attribute a_j.

(3) if $|c_{lh}|>|c_{lj}|$, $f(x_i,a_h)=f(z_l,a_h)$ and $f(x_i,a_j)=f(z_l,a_j)$, then the dissimilarity between the object x_i and the center z_l in the attribute a_h is less than that in the attribute a_j.

To satisfy the above properties, we will propose a new weighting method for categorical attributes. For categorical data, while comparing an object x_i with a center z_l in an attribute a_j, there are two results, either $f(x_i,a_j)=f(z_l,a_j)$ or $f(x_i,a_j)\neq f(z_l,a_j)$. In the new method, we will assign different weights for the two results, Λ and S, respectively. If $f(x_i,a_j)\neq f(z_l,a_j)$, λ_{lj} is used to measure the dissimilarity between the object x_i and the center z_l in the attribute a_j. λ_{lj} is proportional to $|c_{lj}|$. Λ can be computed by using (11). If $f(x_i,a_j)=f(z_l,a_j)$, s_{lj} is used to measure the dissimilarity between the object x_i and the center z_l in the attribute a_j. s_{lj} is inversely proportional to $|c_{lj}|$. The computing method of S will be introduced in the next section.

4. A new weighted dissimilarity measure

In this section, we present a new weighted dissimilarity measure which is applied to the k-modes algorithm. Let $S=[s_{lj}]$ be a k-by-n [0, 1] matrix, $s_{lj}\in[0,1]$ be the weight for the jth attribute in the lth cluster and $\beta\in(1,+\infty)$ be a parameter for controlling attribute weight s_{lj}. The objective function (1) is modified as follows:

$$F_n(W,Z,S)=\sum_{l=1}^{k}\sum_{i=1}^{n}w_{li}d_n(z_l,x_i) \tag{20}$$

subject to the constraints in (2) and

$$\begin{cases}s_{lj}\in[0,1], & 1\le l\le k,\ 1\le j\le m,\\ \sum_{j=1}^{m}s_{lj}=1, & 1\le l\le k.\end{cases} \tag{21}$$

The new weighted dissimilarity measure $d_n(z_l,x_i)$ is defined as follows:

$$d_n(z_l,x)=\sum_{j=1}^{m}\phi_{a_j}(z_l,x_i), \tag{22}$$

where

$$\phi_{a_j}(z_l,x_i)=\begin{cases}1, & f(z_l,a_j)\neq f(x_i,a_j),\\ s_{lj}^{\beta}, & f(z_l,a_j)=f(x_i,a_j).\end{cases} \tag{23}$$

According to the definition of $\phi(\cdot,\cdot)$, one can see that s_{lj} has an effect when the value of an object x_i in the jth attribute is equal to

that of the center z_l of the lth cluster, i.e., $f(x_i,a_j)=f(z_l,a_j)$. The larger s_{lj} is, the larger the dissimilarity between x_i and z_l in the attribute a_j is.

Similar to solving (8), we minimize (20) by iteratively solving the following three minimization problems:

1. Problem P_1: Fix $Z=\hat{Z}$ and $S=\hat{S}$, solve the reduced problem $F_n(W,\hat{Z},\hat{S})$;
2. Problem P_2: Fix $W=\hat{W}$ and $S=\hat{S}$, solve the reduced problem $F_n(\hat{W},Z,\hat{S})$;
3. Problem P_3: Fix $W=\hat{W}$ and $Z=\hat{Z}$, solve the reduced problem $F_n(\hat{W},\hat{Z},S)$.

Problem P_1 is solved by

$$\hat{w}_{li}=\begin{cases}1 & \text{if } d_n(\hat{z}_l,x_i)\le d_n(\hat{z}_h,x_i),\quad 1\le h\le k,\\ 0 & \text{otherwise}\end{cases} \tag{24}$$

for $1\le l\le k$ and $1\le i\le n$.

The solution to problem P_2 is given in Theorem 1.

Theorem 1. *Let U be a set of n categorical objects described by categorical attributes $a_1,a_2,\ldots,a_m$ and $V_{a_j}=\{a_j^{(1)},a_j^{(2)},\ldots,a_j^{(n_j)}\}$, where n_j is the number of categories of attribute a_j for $1\le j\le m$. Let the cluster centers z_l be represented by $[f(z_l,a_1),f(z_l,a_2),\ldots,f(z_l,a_m)]$ for $1\le l\le k$. Then, the quantity $\sum_{l=1}^{k}\sum_{i=1}^{n}w_{li}d_n(z_l,x_i)$ is minimized iff $f(z_l,a_j)=a_j^{(r)}\in V_{a_j}$ where*

$$|\{w_{li}|f(x_i,a_j)=a_j^{(r)},w_{li}=1\}|\ge|\{w_{li}|f(x_i,a_j)=a_j^{(t)},w_{li}=1\}|,\quad 1\le t\le n_j$$

for $1\le j\le m$.

Proof. For any given $\hat{W}$ and $\hat{S}$, all the inner sums of the quantity

$$\sum_{l=1}^{k}\sum_{i=1}^{n}\hat{w}_{li}d_n(z_l,x_i)=\sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m}\hat{w}_{li}\phi_{a_j}(z_l,x_i)$$

are nonnegative and independent. Minimizing the quantity is equivalent to minimizing each inner sum. We write the l,j th inner sum ($1\le l\le k$ and $1\le j\le m$) as

$$\varphi_{l,j}=\sum_{i=1}^{n}\hat{w}_{li}\phi_{a_j}(z_l,x_i).$$

When $f(z_l,a_j)=a_j^{(t)}$, we have

$$\begin{aligned}\varphi_{l,j}&=\sum_{i=1,f(x_i,a_j)\ne a_j^{(t)}}^{n}\hat{w}_{li}+\hat{s}_{lj}\sum_{i=1,f(x_i,a_j)=a_j^{(t)}}^{n}\hat{w}_{li}\\&=\sum_{i=1}^{n}\hat{w}_{li}-\sum_{i=1,f(x_i,a_j)=a_j^{(t)}}^{n}\hat{w}_{li}+\hat{s}_{lj}\sum_{i=1,f(x_i,a_j)=a_j^{(t)}}^{n}\hat{w}_{li}\\&=\sum_{i=1}^{n}\hat{w}_{li}-(1-\hat{s}_{lj})\sum_{i=1,f(x_i,a_j)=a_j^{(t)}}^{n}\hat{w}_{li}.\end{aligned}$$

When $\hat{W}$ and $\hat{S}$ are given, $\sum_{i=1}^{n}\hat{w}_{li}$ and $\hat{s}_{lj}$ are fixed. Since $1-\hat{s}_{lj}\ge 0$, it is clear that $\varphi_{l,j}$ is minimized iff

$$\sum_{i=1,f(x_i,a_j)=a_j^{(t)}}^{n}\hat{w}_{li}=|\{\hat{w}_{li}|f(x_i,a_j)=a_j^{(t)},\hat{w}_{li}=1\}|$$

is maximal for $1\le t\le n_j$. The result follows. □

Theorem 1 tells us that the cluster centers Z are updated in the same manner as the original k-modes algorithm even when we use a new weighted dissimilarity measure. It implies that computing the minimizer $\hat{Z}$ is independent of $\hat{S}$.

Now, the key issue is how to compute S to solve problem P_3. Theorem 2 rigorously shows the updating formula of S.

Theorem 2. *Let $\hat{W}$ and $\hat{Z}$ be fixed and $\beta>1$, $F_n(\hat{W},\hat{Z},S)$ reaches a local minimum only if S satisfies the following conditions*:

$$\hat{s}_{lj}=\frac{1}{\sum_{h=1}^{m}\left[\frac{|c_{lj}|}{|c_{lh}|}\right]^{1/(\beta-1)}}. \tag{25}$$

Proof. We rewrite problem P_3 as

$$\begin{aligned}F_n(\hat{W},\hat{Z},S)&=\sum_{l=1}^{k}\sum_{j=1}^{m}\sum_{i=1}^{n}w_{li}\phi_{a_j}(z_l,x_i)\\&=\sum_{l=1}^{k}\sum_{j=1}^{m}[|c_l|-|c_{lj}|]+\sum_{l=1}^{k}\sum_{j=1}^{m}s_{lj}^{\beta}|c_{lj}|,\end{aligned}$$

where $|c_l|$ and $|c_{lj}|$ are constants for fixed $\hat{W}$ and $\hat{Z}$. This means that minimizing $F_n(\hat{W},\hat{Z},S)$ is equivalent to minimizing

$$\sum_{l=1}^{k}\sum_{j=1}^{m}s_{lj}^{\beta}|c_{lj}|. \tag{26}$$

The Lagrangian multiplier technique is used to obtain the following unconstrained minimization problem:

$$\tilde{P}(S,\alpha)=\sum_{l=1}^{k}\sum_{j=1}^{m}s_{lj}^{\beta}|c_{lj}|-\sum_{l=1}^{k}\alpha_l\left(\sum_{j=1}^{m}s_{lj}-1\right), \tag{27}$$

where $\alpha=[\alpha_1,\alpha_2,\ldots,\alpha_k]$ is the vector containing the Lagrangian multipliers. If $(\hat{S},\hat{\alpha})$ is a minimizer of $\tilde{P}(S,\alpha)$, the gradients in both sets of variables must vanish. Thus,

$$\frac{\partial\tilde{P}(S,\alpha)}{\partial s_{lj}}=\beta|c_{lj}|s_{lj}^{\beta-1}-\alpha_l=0,\quad 1\le l\le k,\ 1\le j\le m \tag{28}$$

and

$$\frac{\partial\tilde{P}(S,\alpha)}{\partial\alpha_l}=\sum_{j=1}^{m}s_{lj}-1=0,\quad 1\le l\le k. \tag{29}$$

From (28) and (29), we obtain

$$\hat{s}_{li}=\frac{1}{\sum_{h=1}^{m}\left[\frac{|c_{lj}|}{|c_{lh}|}\right]^{1/(\beta-1)}}. \tag{30}$$

This shows that (25) is the necessary conditions for the optimization problem P_3 to reach its minimum when W and Z are fixed. □

According to Theorem 2, the dominant level of the mode category is considered in the calculation of the dissimilarity measure. The importance of an attribute in clustering is measured by the frequencies of the mode category in the cluster. The larger the $|c_{lj}|$ is, the more representability the mode category has in the cluster and the more important role the attribute a_j plays in identifying the cluster. We see that if $|c_{lj}|>|c_{lh}|$, then $s_{lj}<s_{lh}$. This means that when $f(x,a_j)=f(z_l,a_j)$ and $f(x,a_h)=f(z_l,a_h)$, if the frequency of $f(z_l,a_j)$ in the lth cluster is more than $f(z_l,a_h)$, the similarity between x and z_l in the attribute a_j is larger than that in the attribute a_h. Let us consider the example in Section 3 again. Without loss of generality, we set $\beta=2$. Using Theorem 2, we obtain $s_{l1}=\frac{1}{3}$ and $s_{l2}=\frac{2}{3}$. Thus, $\phi_{a_1}(z_l,x_1)<\phi_{a_2}(z_l,x_1)$ which fits reality.

The convergence of the k-modes algorithm with the new weighted dissimilarity measure is obtained as in Theorem 3.

Theorem 3. *The k-modes algorithm with the new weighted dissimilarity measure converges to a local minimal solution in a finite number of iterations.*

Proof. We first note that there are only a finite number of possible partitions W. We then show that each possible partition W appears at most once by the algorithm. Assume that $W^{(t_1)}=W^{(t_2)}$, where $t_1\neq t_2$. We note that, given $W^{(t)}$, we can compute the minimizer $Z^{(t)}$ which is independent of $S^{(t)}$. For $W^{(t_1)}$ and $W^{(t_2)}$, we have the minimizers $Z^{(t_1)}$ and $Z^{(t_2)}$, respectively. Using $W^{(t_1)}$ and $Z^{(t_1)}$, and $W^{(t_2)}$ and $Z^{(t_2)}$, we can compute the minimizers $S^{(t_1)}$ and $S^{(t_2)}$, respectively, according to Theorem 2. Although $Z^{(t_1)}$ may be not equal to $Z^{(t_2)}$, $|c_{lj}^{(t_1)}|=|c_{lj}^{(t_2)}|$ for $1\leq j\leq m$, $1\leq l\leq k$. It is clear that $S^{(t_1)}=S^{(t_2)}$. Therefore, we obtain

$$F_n(W^{(t_1)},Z^{(t_1)},S^{(t_1)})=F_n(W^{(t_2)},Z^{(t_2)},S^{(t_2)}).$$

However, the sequence $F_n(\cdot,\cdot,\cdot)$ generated by the algorithm is strictly decreasing. Hence, the result follows. □

The result of Theorem 3 guarantees the convergence of the k-modes algorithm with the new weighted dissimilarity measure.

5. The mixed attribute weighting algorithm

In this section, we integrate Chan's attribute weighting method [38] and the proposed weighting method in Section 4 to form a mixed attribute weighting k-modes algorithm (MWKM) for high-dimensional categorical data. Let $E=\{\Lambda,S\}$. The mixed objective function is written as follows:

$$\begin{aligned}F_e(W,Z,E)&=F_1(W,Z,\Lambda)+F_n(W,Z,S)+T_v\sum_{l=1}^{k}\sum_{j=1}^{m}\lambda_{lj}^{\beta}+T_s\sum_{l=1}^{k}\sum_{j=1}^{m}s_{lj}^{\beta}\\&=\sum_{l=1}^{k}\sum_{i=1}^{n}w_{li}d_e(z_l,x_i)+T_v\sum_{l=1}^{k}\sum_{j=1}^{m}\lambda_{lj}^{\beta}+T_s\sum_{l=1}^{k}\sum_{j=1}^{m}s_{lj}^{\beta}\end{aligned}\tag{31}$$

subject to the same conditions as in (2), (9) and (21). The mixed weighted dissimilarity measure $d_e(z_l,x_i)$ is defined as follows:

$$d_e(z_l,x)=\sum_{j=1}^{m}\sigma_{a_j}(z_l,x_i),\tag{32}$$

where

$$\sigma_{a_j}(z_l,x_i)=\begin{cases}1+\lambda_{lj}^{\beta}, & f(z_l,a_j)\neq f(x_i,a_j),\\ s_{lj}^{\beta}, & f(z_l,a_j)=f(x_i,a_j).\end{cases}\tag{33}$$

$d_e(z_l,x)$ satisfies the three properties mentioned in Section 3. λ_{lj} is inversely proportional to the dispersion of the values from the center in the attribute of the cluster. The high λ_{lj} indicates a small dispersion in the jth attribute of the cluster. Therefore, that attribute is more important in identifying the cluster. However, in contrast to λ_{lj}, s_{lj} is proportional to the dispersion of the values from the center in the attribute of the cluster. The high s_{lj} indicates a large dispersion in the jth attribute of the cluster. Therefore, this attribute is less important in identifying the cluster.

In the objective function (31), $\sum_{l=1}^{k}\sum_{i=1}^{n}w_{li}d_e(z_l,x_i)$ is the sum of the within cluster dispersions that we want to minimize. $T_v\sum_{l=1}^{k}\sum_{j=1}^{m}\lambda_{lj}^{\beta}$ and $T_s\sum_{l=1}^{k}\sum_{j=1}^{m}s_{lj}^{\beta}$ are used to stimulate more attributes to contribute to the identification of clusters in the clustering process. By the Lagrangian multiplier technique, we can obtain the minimum value of $\sum_{j=1}^{m}\lambda_{lj}^{\beta}$:

$$\tilde{\varphi}(\Lambda_l,\alpha)=\sum_{j=1}^{m}\lambda_{lj}^{\beta}+\alpha\left(\sum_{j=1}^{m}\lambda_{lj}-1\right),\tag{34}$$

where $\Lambda_l=[\lambda_{l1},\lambda_{l2},\ldots,\lambda_{lm}]$. If $(\hat{\Lambda}_l,\hat{\alpha})$ is a minimizer of $\tilde{\varphi}(\Lambda_l,\alpha)$, the gradients in both sets of variables must vanish. Thus,

$$\frac{\partial\tilde{\varphi}(\Lambda_l,\alpha)}{\partial\lambda_{lj}}=\beta\lambda_{lj}^{(\beta-1)}+\alpha=0,\quad 1\leq j\leq m\tag{35}$$

and

$$\frac{\partial\tilde{\varphi}(\Lambda_l,\alpha)}{\partial\alpha}=\sum_{j=1}^{m}\lambda_{lj}-1=0.\tag{36}$$

From (35) and (36), we obtain

$$\hat{\lambda}_{lj}=\frac{1}{m},\quad 1\leq j\leq m.\tag{37}$$

We see that when the λ_{lj} are the same for $1\leq j\leq m$, $\sum_{j=1}^{n_j}\lambda_{lj}^{\beta}$ achieves the minimum value, i.e.,

$$\min\sum_{j=1}^{m}\lambda_{lj}^{\beta}=m\left(\frac{1}{m}\right)^{\beta}.$$

We also know that

$$\sum_{j=1}^{m}\lambda_{lj}^{\beta}\leq\left(\sum_{j=1}^{h}\lambda_{lj}\right)^{\beta}+\sum_{q=h+1}^{m}\lambda_{lj}^{\beta}\leq\left(\sum_{j=1}^{m}\lambda_{lj}\right)^{\beta}=1.$$

If only one of the λ_{lj} for $1\leq j\leq m$ is nonzero, $\sum_{j=1}^{m}\lambda_{lj}^{\beta}$ achieves the maximum value, i.e.,

$$\max\sum_{j=1}^{m}\lambda_{lj}^{\beta}=1.$$

This means that the smaller $\sum_{j=1}^{m}\lambda_{lj}^{\beta}$ is, the more attributes the weights are assigned to. Using the same analysis, we also can obtain minimal and maximal values of $\sum_{j=1}^{m}s_{lj}^{\beta}$, i.e.,

$$m\left(\frac{1}{m}\right)^{\beta}\leq\sum_{j=1}^{m}s_{lj}^{\beta}\leq 1.$$

Therefore, the last two terms $T_v\sum_{l=1}^{k}\sum_{j=1}^{m}\lambda_{lj}^{\beta}$ and $T_s\sum_{l=1}^{k}\sum_{j=1}^{m}s_{lj}^{\beta}$ are added to the objective function so that we can simultaneously minimize the within cluster dispersion and minimize them to stimulate more dimensions to contribute to the identification of clusters. The parameter $T_v(\geq 0)$ and $T_s(\geq 0)$ are used to balance which part plays a more important role in the minimization process of (31). The larger T_v and T_s are, the more the last two terms contribute in the optimization process and the "smoother" or fuzzier of the resulting Λ and S are. However, the values of T_v and T_s should not be too large. The reason is that when T_v and T_s are very large so that each element in Λ and S is close to $1/m$. In this case, the mixed dissimilarity measure (33) becomes the simple matching dissimilarity measure (4) plus a constant, i.e.,

$$\sigma_{a_j}(z_j,x_i)=\delta_{a_j}(z_j,x_i)+\frac{1}{m}$$

and the mixed objective function (31) also becomes the original objective function (1) plus constants, i.e.,

$$F_e(W,Z,\Lambda)=F(W,Z)+n+(T_v+T_s)km\left(\frac{1}{m}\right)^{\beta}.$$

This will make the clustering process back to the standard k-modes.

To minimize the mixed objective function, the matrices W, Z and E are updated according to the following methods.

Given $\hat{Z}$ and $\hat{E}$ are fixed, W is updated as

$$\hat{w}_{li}=\begin{cases}1 & \text{if } d_e(\hat{z}_l,x_i)\leq d_e(\hat{z}_h,x_i),\ 1\leq h\leq k,\\ 0 & \text{otherwise}\end{cases}\tag{38}$$

for $1\leq i\leq n$, $1\leq l\leq k$.

Given $\hat{W}$ and $\hat{E}$ are fixed, Z is updated in the same manner as that in Section 2.

Theorem 4. *Let U be a set of n categorical objects described by categorical attributes $a_1,a_2,\ldots,a_m$ and $V_{a_j}=\{a_j^{(1)},a_j^{(2)},\ldots,a_j^{(n_j)}\}$, where n_j is the number of categories of attribute a_j for $1\le j\le m$. Let the cluster centers z_l be represented by $[f(z_l,a_1), f(z_l,a_2),\ldots f(z_l,a_m)]$ for $1\le l\le k$. Then, the mixed objective function (31) is minimized iff $f(z_l,a_j)=a_j^{(r)}\in V_{a_j}$ where*

$$|\{w_{li}|f(x_i,a_j)=a_j^{(r)},w_{li}=1\}|\ge|\{w_{li}|f(x_i,a_j)=a_j^{(t)},w_{li}=1\}|,\quad 1\le t\le n_j$$

for $1\le j\le m$.

Proof. For a given $\hat{W}$ and $\hat{E}$, $T_v\sum_{l=1}^{k}\sum_{j=1}^{m}\lambda_{lj}^{\beta}$ and $T_s\sum_{l=1}^{k}\sum_{j=1}^{m}s_{lj}^{\beta}$ are constants. Minimizing F_e is equivalent to minimizing each inner sum of the quantity

$$\sum_{l=1}^{k}\sum_{i=1}^{n}\hat{w}_{li}d_e(z_l,x_i)=\sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m}\hat{w}_{li}\sigma_{a_j}(z_l,x_i),$$

which is nonnegative and independent. We write the l,jth inner sum ($1\le l\le k$ and $1\le j\le m$) as

$$\varphi_{l,j}=\sum_{i=1}^{n}\hat{w}_{li}\sigma_{a_j}(z_l,x_i).$$

When $f(z_l,a_j)=a_j^{(t)}$, we have

$$\begin{aligned}\varphi_{l,j}&=(1+\hat{\lambda}_{lj})\sum_{i=1,f(x_i,a_j)\ne a_j^{(t)}}^{n}\hat{w}_{li}+\hat{s}_{lj}\sum_{i=1,f(x_i,a_j)=a_j^{(t)}}^{n}\hat{w}_{li}\\&=(1+\hat{\lambda}_{lj})\sum_{i=1}^{n}\hat{w}_{li}-(1+\hat{\lambda}_{lj})\sum_{i=1,f(x_i,a_j)=a_j^{(t)}}^{n}\hat{w}_{li}+\hat{s}_{lj}\sum_{i=1,f(x_i,a_j)=a_j^{(t)}}^{n}\hat{w}_{li}\\&=\sum_{i=1}^{n}\hat{w}_{li}-(1+\hat{\lambda}_{lj}-\hat{s}_{lj})\sum_{i=1,f(x_i,a_j)=a_j^{(t)}}^{n}\hat{w}_{li}.\end{aligned}$$

When $\hat{W}$ and $\hat{S}$ are given, $\sum_{i=1}^{n}\hat{w}_{li}$ and $\hat{s}_{lj}$ are fixed. Since $1+\hat{\lambda}_{lj}-\hat{s}_{lj}\ge 0$, it is clear that $\varphi_{l,j}$ is minimized iff

$$\sum_{i=1,f(x_i,a_j)=a_j^{(t)}}^{n}\hat{w}_{li}=|\{\hat{w}_{li}|f(x_i,a_j)=a_j^{(t)},\hat{w}_{li}=1\}|$$

is maximal for $1\le t\le n_j$. The result follows. □

Theorem 4 tells us that it is safe to update the cluster centers Z in the same manner as the original k-modes algorithm does even when we use the mixed weighted dissimilarity measure.

Given $\hat{W}$ and $\hat{Z}$ are fixed, E is updated by using Theorem 5 to minimize the mixed objective function.

Theorem 5. *Let $\hat{W}$ and $\hat{Z}$ be fixed and $\beta>1$, $F_e(\hat{W},\hat{Z},E)$ reaches a local minimum only if Λ and S satisfy the following conditions*:

$$\hat{\lambda}_{lj}=\frac{1}{\sum_{h=1}^{m}\left[\frac{|c_l|-|c_{lj}|+T_v}{|c_l|-|c_{lh}|+T_v}\right]^{1/(\beta-1)}}\tag{39}$$

and

$$\hat{s}_{lj}=\frac{1}{\sum_{h=1}^{m}\left[\frac{|c_{lj}|+T_s}{|c_{lh}|+T_s}\right]^{1/(\beta-1)}}.\tag{40}$$

Proof. We rewrite problem P_3 as

$$\begin{aligned}&F_e(\hat{W},\hat{Z},E)\\&=\sum_{l=1}^{k}\sum_{j=1}^{m}\sum_{i=1}^{n}w_{li}\phi_{a_j}(z_l,x_i)+T_v\sum_{l=1}^{k}\sum_{j=1}^{m}\lambda_{lj}^{\beta}+T_s\sum_{l=1}^{k}\sum_{j=1}^{m}s_{lj}^{\beta}\\&=\sum_{l=1}^{k}\sum_{j=1}^{m}\lambda_{lj}^{\beta}[|c_l|-|c_{lj}|]+T_v\sum_{l=1}^{k}\sum_{j=1}^{m}\lambda_{lj}^{\beta}+\sum_{l=1}^{k}\sum_{j=1}^{m}s_{lj}^{\beta}|c_{lj}|+T_s\sum_{l=1}^{k}\sum_{j=1}^{m}s_{lj}^{\beta},\end{aligned}$$

where $|c_l|$ and $|c_{lj}|$ are constants for fixed $\hat{W}$ and $\hat{Z}$. The Lagrangian multiplier technique is used to obtain the following unconstrained minimization problem:

$$\begin{aligned}&\tilde{P}(\Lambda,S,\alpha,\eta)\\&=\sum_{l=1}^{k}\sum_{j=1}^{m}\lambda_{lj}^{\beta}[|c_l|-|c_{lj}|]+T_v\sum_{l=1}^{k}\sum_{j=1}^{m}\lambda_{lj}^{\beta}+\sum_{l=1}^{k}\sum_{j=1}^{m}s_{lj}^{\beta}|c_{lj}|\\&\quad+T_s\sum_{l=1}^{k}\sum_{j=1}^{m}s_{lj}^{\beta}-\sum_{l=1}^{k}\alpha_l\left(\sum_{j=1}^{m}\lambda_{lj}-1\right)-\sum_{l=1}^{k}\eta_l\left(\sum_{j=1}^{m}s_{lj}-1\right)\end{aligned}\tag{41}$$

where $\alpha=[\alpha_1,\alpha_2,\ldots,\alpha_k]$ and $\eta=[\eta_1,\eta_2,\ldots,\eta_k]$ are two vectors containing the Lagrangian multipliers. If $(\hat{\Lambda},\hat{S},\hat{\alpha},\hat{\eta})$ is a minimizer of $\tilde{P}(\Lambda,S,\alpha,\eta)$, the gradients in both sets of variables must vanish. Thus,

$$\frac{\partial\tilde{P}(\Lambda,S,\alpha,\eta)}{\partial\lambda_{lj}}=\beta\lambda_{lj}^{\beta-1}[|c_l|-|c_{lj}|]+\beta\lambda_{lj}^{\beta-1}T_v-\alpha_l=0,\quad 1\le l\le k,\ 1\le j\le m,\tag{42}$$

$$\frac{\partial\tilde{P}(\Lambda,S,\alpha,\eta)}{\partial s_{lj}}=\beta s_{lj}^{\beta-1}|c_{lj}|+\beta s_{lj}^{\beta-1}T_v-\eta_l=0,\quad 1\le l\le k,\ 1\le j\le m,\tag{43}$$

$$\frac{\partial\tilde{P}(\Lambda,S,\alpha,\eta)}{\partial\alpha_l}=\sum_{j=1}^{m}\lambda_{lj}-1=0,\quad 1\le l\le k\tag{44}$$

and

$$\frac{\partial\tilde{P}(\Lambda,S,\alpha,\eta)}{\partial\eta_l}=\sum_{j=1}^{m}s_{lj}-1=0,\quad 1\le l\le k.\tag{45}$$

From (42), (43), (44) and (45), we obtain

$$\hat{\lambda}_{lj}=\frac{1}{\sum_{h=1}^{m}\left[\frac{|c_l|-|c_{lj}|+T_v}{|c_l|-|c_{lh}|+T_v}\right]^{1/(\beta-1)}}\tag{46}$$

and

$$\hat{s}_{lj}=\frac{1}{\sum_{h=1}^{m}\left[\frac{|c_{lj}|+T_s}{|c_{lh}|+T_s}\right]^{1/(\beta-1)}}.\tag{47}$$

This shows that (39) and (40) are the necessary conditions for the optimization problem P_3 to reach its minimum when W and Z are fixed. We also see that computing Λ and S are independent of each other. □

The MWKM algorithm that minimizes (31) is summarized as follows:

Algorithm—MWKM
Input: The number of clusters k and the parameters β, T_v and T_s;
Randomly choose k cluster centers and set all initial weights of Λ and S to $1/m$;
REPEAT
Update the partition matrix W by (38);
Update the cluster centers Z by Theorem 4;
Update the dimension weights E by Theorem 5;
UNTIL (the objective function obtains its local minimum value).

Theorem 6. *The mixed attribute weighting algorithm converges to a local minimal solution in a finite number of iterations.*

Proof. We first note that there are only a finite number of possible partitions W. We then show that each possible partition W appears at most once by the algorithm. Assume that $W^{(t_1)}=W^{(t_2)}$, where $t_1\neq t_2$. We note that, given $W^{(t)}$, we can compute the minimizer $Z^{(t)}$ which is independent of $S^{(t)}$. For $W^{(t_1)}$ and $W^{(t_2)}$, we have the minimizers $Z^{(t_1)}$ and $Z^{(t_2)}$, respectively. Using $W^{(t_1)}$ and $Z^{(t_1)}$, and $W^{(t_2)}$ and $Z^{(t_2)}$, we can compute the minimizers $E^{(t_1)}=\{\Lambda^{(t_1)},S^{(t_1)}\}$ and $E^{(t_2)}=\{\Lambda^{(t_2)},S^{(t_2)}\}$, respectively. Although $Z^{(t_1)}$ may be not equal to $Z^{(t_2)}$, $|c_{lj}^{(t_1)}|=|c_{lj}^{(t_2)}|$ for $1\leq j\leq m$, $1\leq l\leq k$. It is clear that $\Lambda^{(t_1)}=\Lambda^{(t_2)}$ and $S^{(t_1)}=S^{(t_2)}$. Therefore, we obtain

$$F_e(W^{(t_1)},Z^{(t_1)},E^{(t_1)})=F_e(W^{(t_2)},Z^{(t_2)},E^{(t_2)}).$$

However, the sequence $F_e(\cdot,\cdot,\cdot)$ generated by the algorithm is strictly decreasing. Hence, the result follows. □

In Fig. 2, we show the 100 curves, where each curve refers to the objective function values against the iterations of the proposed algorithm with different initial cluster centers on the soybean data set from UCI [55]. It is clear from the figure that the objective function values are decreasing in each curve. With our results, we show that the objective function values are decreasing when the mixed weighting method is used. We also see in Fig. 2 that the algorithm stops after a finite number of iterations, i.e., the objective function values do not decrease any more. This is exactly the results we showed in Theorem 6. The algorithm can be used safely.

The proposed algorithm is scalable to either the number of objects, attributes or clusters. This is because the new algorithm only adds a new step to the k-modes clustering process to calculate the attribute weights of each cluster. The runtime complexity can be analyzed as follows. We only consider the three major computational steps:

- *Partitioning the objects*: After initialization of the attribute weights of each cluster and the cluster centers, a cluster membership is assigned to each object. This process simply compares the summation of

$$d_e(z_l,x_i)=\sum_{j=1}^{m}\sigma_{a_j}(z_l,x_i)$$

for each object in all k clusters. Thus, the complexity for this step is $O(mnk)$ operations.
- *Updating cluster centers*: Given the partition matrix W, updating cluster centers is to find the modes of the objects in the same cluster. Thus, for k clusters, the computational complexity for this step is $O(mnk)$.
- *Calculating attribute weights*: The last phase of this algorithm is to calculate the attribute weights for all clusters based on the partition matrices W and Z. In this step, we only go through the whole data set once to update the attribute weights. The computational complexity of this step is also $O(mnk)$.

If the clustering process needs t iterations to converge, the total computational complexity of this algorithm is $O(mnkt)$ which is as much as the original k-modes algorithm ($O(mnkt)$). This shows that the computational complexity increases linearly as the number of objects, attributes or clusters increases.

6. Experimental analysis

The main aim of this section is to evaluate the clustering performance and scalability of the mixed attribute weighting algorithm (MWKM). The motivation for development of the MWKM algorithm is to cluster high-dimensional categorical data. To better understand the properties of the algorithm, synthetic data with controlled cluster structures and data sparsity were first used to investigate the relationships of the dispersion and weights of attributes in each cluster, the behavior of parameters β,T_v,T_s, and the performance of the algorithm on clustering accuracy in comparison with other clustering algorithms. The structure of a synthetic categorical data set has the following characteristics: (1) it contains more than one cluster, (2) the data values of a cluster are concentrated on a subset of relevant attributes, whereas other irrelevant attributes contain sparse values, and (3) the relevant attributes for different clusters can overlap. Fig. 3 and [42] illustrates an example of a synthetic data set with four clusters.

A similar process as given by Zait and Messatfa [56] was used to generate the synthetic data sets with different cluster structures. Table 2 gives the algorithm for synthetic data generation.

In addition, we also used several real data sets downloaded from the UCI Machine Learning Repository [55]. These data sets

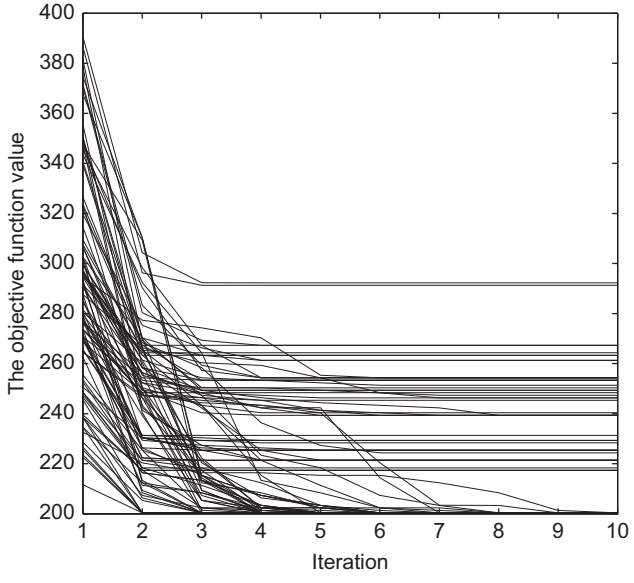

Fig. 2. The objective function values against the iterations with different initial guesses.

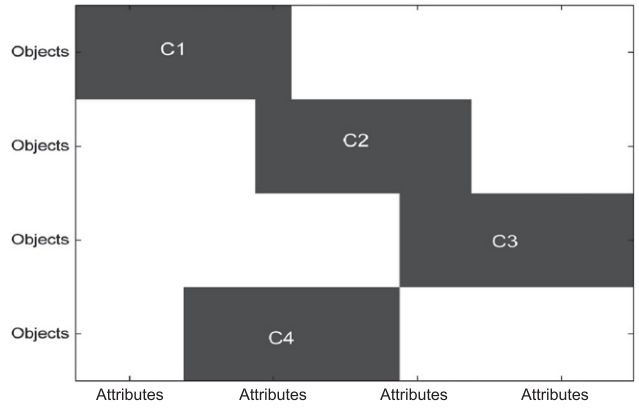

Fig. 3. The structure of a synthetic data set where the gray areas represent four clusters that are formed in different subspaces, and the white areas represent the dimensions where data entries are sparse values.

Table 2
The algorithm for generating synthetic data.

1. Specify the number of clusters k, the number of objects in each cluster $|c_l|$ for $1 \le l \le k$, the number of attributes m, the number of categories in each attribute n_j for $1 \le j \le m$, the number of relevant attributes in each cluster $r_{lj}(\le m)$ for $1 \le l \le k$, the lowest frequency of each mode value in each relevant attribute f_r and the lowest frequency of each mode value in each irrelevant attribute f_{ir};
2. For $l=1$ to k
   generate a set c_l of $|c_l|$ empty records with m attributes;
   randomly select the r_{lj} attributes from all the attributes as relevant attributes of the lth cluster;
   For $j=1$ to m
   randomly select a category $a_j^{(q)}$ from V_{a_j} as the value of the mode of the lth cluster;
   If the hth attribute is relevant to the lth clusters then
   set $|c_{lj}| = f_r + \lfloor(|c_l| - f_r) * rand()\rfloor$;
   //$rand() \in [0,1]$ is a random function.
   Else
   set $|c_{lj}| = f_{ir} + \lfloor(f_r - f_{ir}) * rand()\rfloor$;
   End If
   randomly select a set X of $|c_{lj}|$ objects from the lth cluster;
   For each x in X
   set $f(x,a_j) = a_j^{(q)}$;
   End For
   For each y in $c_l - X$
   randomly select a category $a_j^{(p)}$ from $V_{a_j} - \{a_j^{(q)}\}$ which satisfies
   its frequency in the lth cluster is less than $|c_{lj}|$;
   set $f(x,a_j) = a_j^{(p)}$;
   End For
   End For
   End For

Table 3
The seven data sets from UCI.

Data set	Objects	Attributes	Clusters
Soybean	47	35	4
Heart disease	303	13	2
Dermatology	366	33	6
Breast cancer	699	9	2
Mushroom	8124	22	2
Connect-4	67,557	45	3
Census	2,458,284	68	N/A

are shown in Table 3. If the attribute value of an object in the given data sets is missing, then we denote the attribute value by ε.

6.1. Performance analysis

In the performance analysis, we adopt the three widely used methods to evaluate the results of clustering algorithms:

The category utility function: The category utility (CU) function [57] is an internal criterion which attempts to maximize both the probability that two data objects in the same cluster obtain the same attribute values and the probability that data points from different clusters have different attributes. The expression to calculate the expected value of the CU function is shown in the following equation:

$$CU = \sum_{l=1}^{k} \frac{|c_l|}{n} \sum_{j=1}^{m} \sum_{q=1}^{n_j} [P(a_j^{(q)}|c_l)^2 - P(a_j^{(q)})^2], \tag{48}$$

where $P(a_j^{(q)}|c_l) = |\{x_i | f(x_i,a_j) = a_j^{(q)}, x_i \in c_l\}|/|c_l|$, $P(a_j^{(q)}) = |\{x_i | f(x_i,a_j) = a_j^{(q)}, x_i \in U\}|/n$, and c_l is a set of objects in the lth cluster.

The adjusted rand index: The adjusted rand index [58] is an external criterion which attempts to measure the similarity between two partitions of objects in the same data set. Given a set U of n data objects and two groupings (e.g., clusterings) of these objects, namely $C = \{c_1, c_2, \ldots, c_k\}$ and $P = \{p_1, p_2, \ldots, p'_k\}$, the overlappings between C and P can be summarized in a contingency table where n_{ij} denotes the number of common objects of groups c_i and p_j : $n_{ij} = |c_i \cap p_j|$. The adjusted rand index is defined as

$$AdjustedIndex = \frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex}$$

more specifically,

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{b_i}{2} \sum_j \binom{d_j}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{b_i}{2} + \sum_j \binom{d_j}{2}\right] - \left[\sum_i \binom{b_i}{2} \sum_j \binom{d_j}{2}\right] \Big/ \binom{n}{2}}, \tag{49}$$

where n_{ij}, b_i, d_j are values from the contingency table (Table 4). Since these given data sets contain the clustering label on each data object, we will evaluate the clustering results by using ARI to compare them with the original clustering labels. If the clustering result is close to the true class distribution, then the value of ARI is high.

Table 4
Notation for the contingency table for comparing two partitions.

C	P				$Sums$
	p_1	p_2	$\cdots$	p'_k	
c_1	n_{11}	n_{12}	$\cdots$	$n_{1k'}$	b_1
c_2	n_{21}	n_{22}	$\cdots$	$n_{2k'}$	b_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
c_k	n_{k1}	n_{k2}	$\cdots$	n'_{kk}	b_k
$Sums$	d_1	d_2	$\cdots$	d'_k	

The set matching technique: This category of methods is based on measuring the shared set cardinality between two clusterings. Similar to the adjusted rand index, the set matching technique is also an external criterion in which external information-class labels need be used. It computes the best matches between clusters (in terms of shared points) from each of the two clusterings and returns a value equal to the total number of points shared between pairs of matched clusters. The simplest form of set matching technique is called the set matching error

(*ER*) [59], which is defined as

$$ER = 1 - \sum_{i=1}^{k} \frac{\max_{j=1}^{k'} n_{ij}}{n}, \quad (50)$$

where n_{ij},k,k' are values from Table 4. If the clustering result is close to the true class distribution, then the value of *ER* is low.

Based on the above evaluation measures, we analyze the performance of the proposed algorithm on two synthetic data sets, called SDS1 and SDS2, and five real data sets: the soybean data, the heart disease data, the dermatology data, the breast cancer data and the mushroom data, respectively. On each data set, the analysis consists of the following three parts:

Part 1: We compare the MWKM algorithm with other four algorithms: the original *k*-modes algorithm (KM) [7], Chan's weighting algorithm (WKM) [38] and Jing's weighting algorithm (EWKM) [42] and the *k*-modes algorithm with the new dissimilarity measure (NWKM) proposed in Section 4. To ensure that the comparisons are in a uniform environmental condition, we first set the number of clusters is equal to the "true" number of classes for each of the given data sets. Furthermore, due to the fact that the performance of these algorithms depends on initial cluster centers, we randomly select 100 initial cluster centers and carry out 100 runs of KM, WKM, EWKM, NWKM and MWKM on each data set, respectively. In each run, the same initial cluster centers are used in the five algorithms. Finally, we fix the other parameters $\beta = 2$ which most researchers proposed, $T_v = 1$ and $T_s = 1$. In the above parameters setting, we show the average values and the best values of the 100 runs of each algorithm for *CU*, *ARI* and *ER* on these data sets in Tables 5–11. In each cell of these tables, the left value of "/" denotes the average value and the right value of "/" denotes the best value.

Part 2: To analyze the effect of the parameter β, we first fix the other parameters $T_v = 1$, $T_s = 1$ and randomly select 100 initial cluster centers for each of the given data sets. Next, we apply the WKM algorithm and the MWKM algorithm to cluster the data sets with different β values, respectively, and compute the average values for *ARI*,*CU* and *ER* in these 100 runs. Figs. 4, 6, 8, 10, 12, 14 and 16 show the comparison results of the WKM algorithm and the MWKM algorithm with different β values (the value of β is from 2 to 10 with step length of 1).

Part 3: To analyze the effect of the parameters T_v and T_s, we first randomly select 100 initial cluster centers for each of the given data sets. Next, the MWKM algorithm is used to cluster the data sets with different T_v values while $\beta = 2$ and $T_s = 1$ are fixed and the MWKM algorithm is used to cluster the data sets with different T_s values while $\beta = 2$ and $T_v = 1$ are fixed. Figs. 5, 7, 9, 11, 13, 15 and 17 show the average values for *ARI*,*CU* and *ER* in these 100 runs of the MWKM algorithm with different T_v or T_s values (the values of T_v and T_s are from 0 to n/k with step length of $n/(10 \times k)$, where n and k are the number of objects and the number of clusters in the data set, respectively).

6.1.1. Synthetic data I

The first synthetic data set SDS1 has 1000 objects, 30 attributes, and 5 clusters. The number of categories of each attribute is set to 5. In each cluster, the number of objects and the number of relevant attribute are set to 200 and 6, respectively. The parameter f_r and f_{ir} are set to 0.5 and 0.2, respectively.

Table 5
The summary results for 100 runs of five algorithms on the SDS1 data set.

	KM	WKM	EWKM	NWKM	MWKM
CU	2.0007/2.4134	1.5187/1.7868	1.5256/1.7497	2.1269/2.4322	2.3085/2.5142
ARI	0.5630/0.7692	0.3261/0.4125	0.3539/0.4654	0.6705/0.7963	0.7266/0.8461
ER	0.2479/0.0990	0.3958/0.2820	0.3737/0.2680	0.1868/0.0810	0.1425/0.0640

Table 6
The summary results for 100 runs of five algorithms on the SDS2 data set.

	KM	WKM	EWKM	NWKM	MWKM
CU	2.8417/3.1761	1.4914/1.8047	1.4194/1.5944	2.8952/3.1923	3.0997/3.2395
ARI	0.6555/0.7748	0.1467/0.2498	0.1407/0.1860	0.7086/0.7762	0.7665/0.8306
ER	0.2020/0.1080	0.5871/0.4855	0.6116/0.5360	0.1760/0.1035	0.1254/0.0800

Table 7
The summary results for 100 runs of five algorithms on the soybean data set.

	KM	WKM	EWKM	NWKM	MWKM
CU	4.6969/5.5575	3.4463/5.5575	4.6418/5.5575	4.7652/5.5575	4.8981/5.5575
ARI	0.6952/1.0000	0.4989/1.0000	0.6884/1.0000	0.7031/1.0000	0.7745/1.0000
ER	0.1500/0.0000	0.3181/0.0000	0.1717/0.0000	0.1359/0.0000	0.1164/0.0000

Table 8
The summary results for 100 runs of five algorithms on the heart disease data set.

	KM	WKM	EWKM	NWKM	MWKM
CU	0.5746/0.6867	0.5311/0.6858	0.4629/0.6552	0.5908/0.6913	0.6438/0.7174
ARI	0.2558/0.3869	0.1209/0.2730	0.0856/0.2870	0.2714/0.3954	0.2915/0.4121
ER	0.2613/0.1881	0.3406/0.2376	0.3702/0.2310	0.2576/0.1848	0.2389/0.1782

Table 9
The summary results for 100 runs of five algorithms on the dermatology data set.

	KM	WKM	EWKM	NWKM	MWKM
CU	3.9374/4.5821	2.3193/4.1610	3.3288/4.5599	4.0158/4.6969	4.1368/4.7100
ARI	0.4505/0.6945	0.3349/0.7173	0.4731/0.7436	0.4833/0.7469	0.5081/0.7536
ER	0.3131/0.1831	0.4496/0.2104	0.3218/0.1475	0.2986/0.1557	0.2728/0.1530

Table 10
The summary results for 100 runs of five algorithms on the breast cancer data set.

	KM	WKM	EWKM	NWKM	MWKM
CU	0.8337/1.0431	0.5411/0.9624	0.5301/0.5703	0.8540/1.0421	0.8960/1.0431
ARI	0.5137/0.7244	0.2848/0.5630	0.2898/0.3123	0.5524/0.7236	0.5957/0.7244
ER	0.1518/0.0730	0.2197/0.1245	0.2173/0.2060	0.1405/0.0744	0.1231/0.0730

Table 11
The summary results for 100 runs of five algorithms on the mushroom data set.

	KM	WKM	EWKM	NWKM	MWKM
CU	1.4090/1.7306	0.0471/1.2952	0.6434/1.6471	1.4453/1.7362	1.4928/1.7362
ARI	0.2526/0.6129	0.0004/0.0439	0.1096/0.7047	0.2988/0.6198	0.3218/0.6198
ER	0.2824/0.1054	0.4810/0.3951	0.3854/0.0803	0.2664/0.1064	0.2550/0.1064

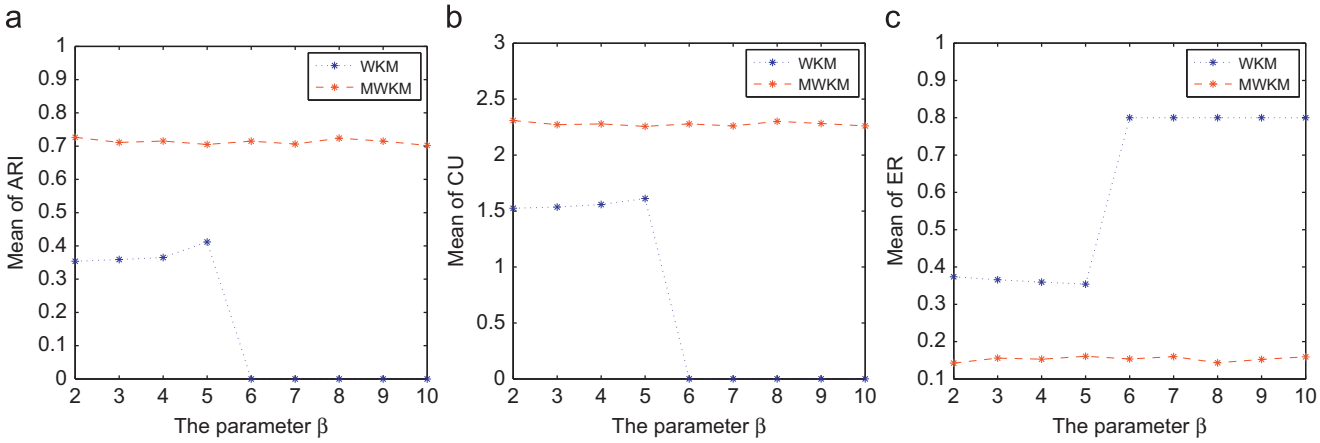

Fig. 4. (a) Means of ARI with respect to different values of β on the SDS1 data. (b) Means of CU with respect to different values of β on the SDS1 data. (c) Means of ER with respect to different values of β on the SDS1 data.

6.1.2. *Synthetic data II*

The second synthetic data set SDS2 has 2000 objects, 50 attributes, and 10 clusters. The number of categories of each attribute is set to 5. In each cluster, the number of objects and the number of relevant attribute are set to 200 and 5, respectively. The parameter f_r and f_{ir} are set to 0.5 and 0.2, respectively.

6.1.3. *Soybean data*

The soybean data set has 47 records, each of which is described by 35 attributes. Each record is labeled as one of the four diseases: Diaporthe Stem Canker, Charcoal Rot, Rhizoctonia Root Rot, and Phytophthora Rot. Except for Phytophthora Rot which has 17 records, all other diseases have 10 records each.

6.1.4. *Heart disease data*

This data set generated at the Cleveland Clinic has 303 instances with eight categorical and five numeric features. It contains two classes: normal (164 data objects) and heart patient (139 data objects). In the test, all numerical attributes are removed from the data set.

6.1.5. *Dermatology data*

This data set describes clinical features and histopathological features of erythemato-squamous diseases in dermatology. It contains 366 elements and 33 categorical attributes. It has six clusters: psoriasis (112 data objects), seboreic dermatitis (61 data objects), lichen planus (72 data objects), pityriasis rosea (49 data objects), chronic dermatitis (52 data objects) and pityriasis rubra pilaris (20 data objects).

6.1.6. *Breast cancer data*

This breast cancer domain was obtained from the University Medical Center, Institute of Oncology, Ljubljana, Yugoslavia. It

consists of 699 data objects and 9 categorical attributes. It has two clusters, benign (458 data objects) and malignant (241 data objects).

6.1.7. Mushroom data

This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family. It consists of 8124 data objects and 23 categorical attributes. Each object belongs to one of two classes, edible (4208 objects) and poisonous (3916 objects).

According to Tables 5–11, we see that the WKM and EWKM algorithms have poor performance in clustering categorical data. This indicates that these weighting algorithms are not fit for clustering categorical data. Furthermore, we also see that the performance of the proposed algorithm is superior to the KM, WKM, EWKM and NWKM algorithms for *ARI*, *CU* and *ER* in

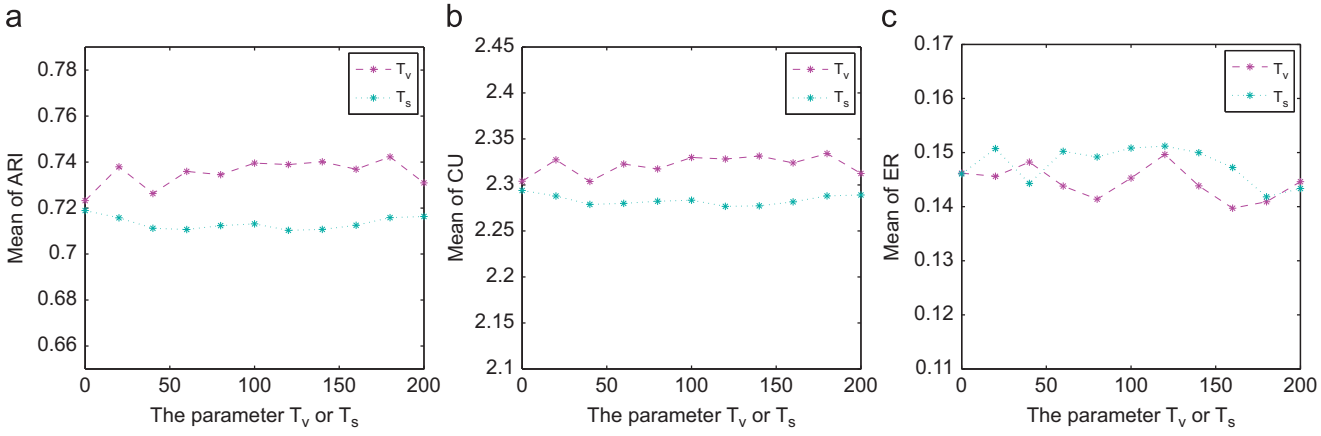

Fig. 5. (a) Means of ARI with respect to different values of T_v or T_s on the SDS1 data. (b) Means of CU with respect to different values of T_v or T_s on the SDS1 data. (c) Means of ER with respect to different values of T_v or T_s on the SDS1 data.

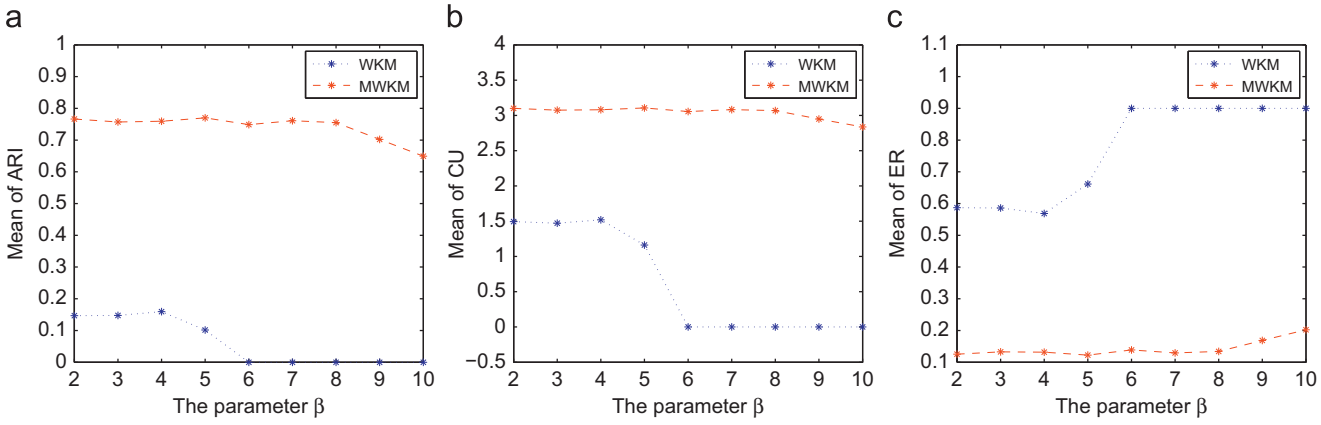

Fig. 6. (a) Means of ARI with respect to different values of β on the SDS2 data. (b) Means of CU with respect to different values of β on the SDS2 data. (c) Means of ER with respect to different values of β on the SDS2 data.

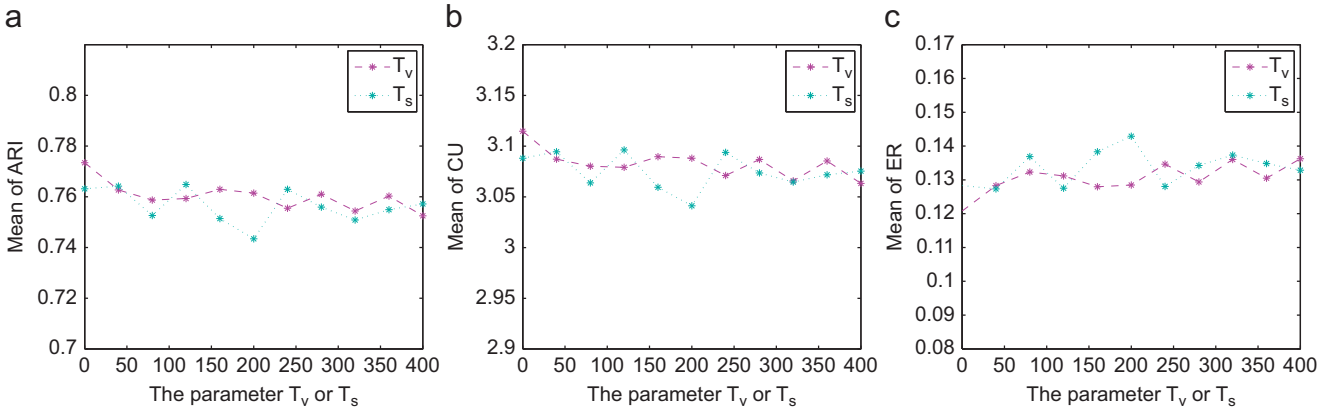

Fig. 7. (a) Means of ARI with respect to different values of T_v or T_s on the SDS2 data. (b) Means of CU with respect to different values of T_v or T_s on the SDS2 data. (c) Means of ER with respect to different values of T_v or T_s on the SDS2 data.

clustering the data sets. Figs. 4–17 show that the clustering results of the proposed algorithm were not sensitive to the change of β, T_v and T_s values. This means that the proposed algorithm has very good robustness.

6.2. Scalability analysis

In the scalability analysis, we test the KM algorithm, the WKM algorithm, the EWKM algorithm and the MWKM algorithm on the connect-4 data and the census data from UCI [55]. The computational results are performed by using a machine with an Intel Q9400 and 2G RAM. The computational times of algorithms are plotted with respect to the number of objects, attributes and clusters, while the other corresponding parameters are fixed.

6.2.1. Connect-4 data

The connect-4 data contains all legal 8-ply positions in the game of connect-4 in which neither player has won yet, and in

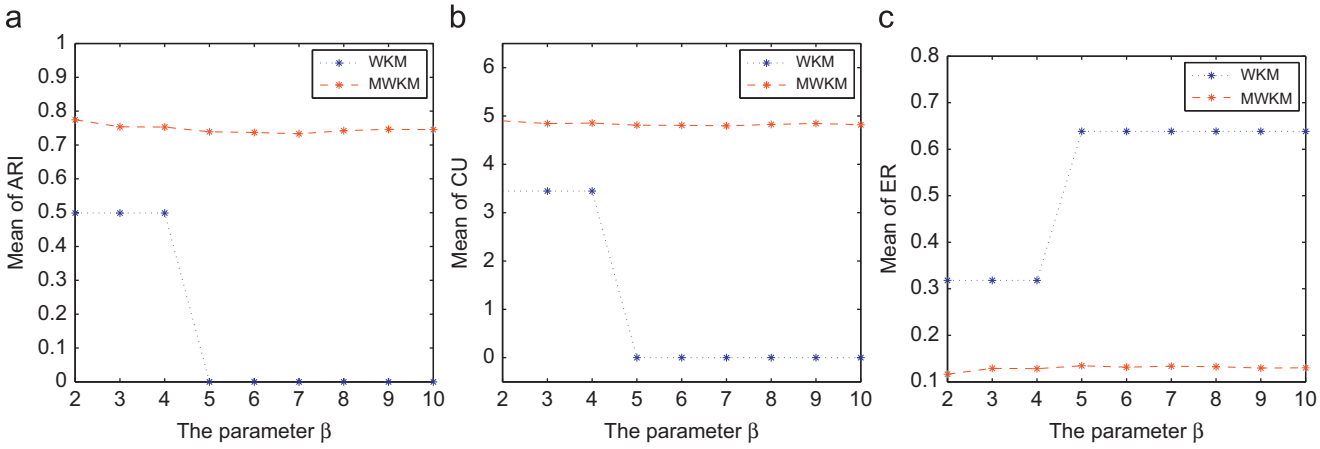

Fig. 8. (a) Means of ARI with respect to different values of β on the soybean data. (b) Means of CU with respect to different values of β on the soybean data. (c) Means of ER with respect to different values of β on the soybean data.

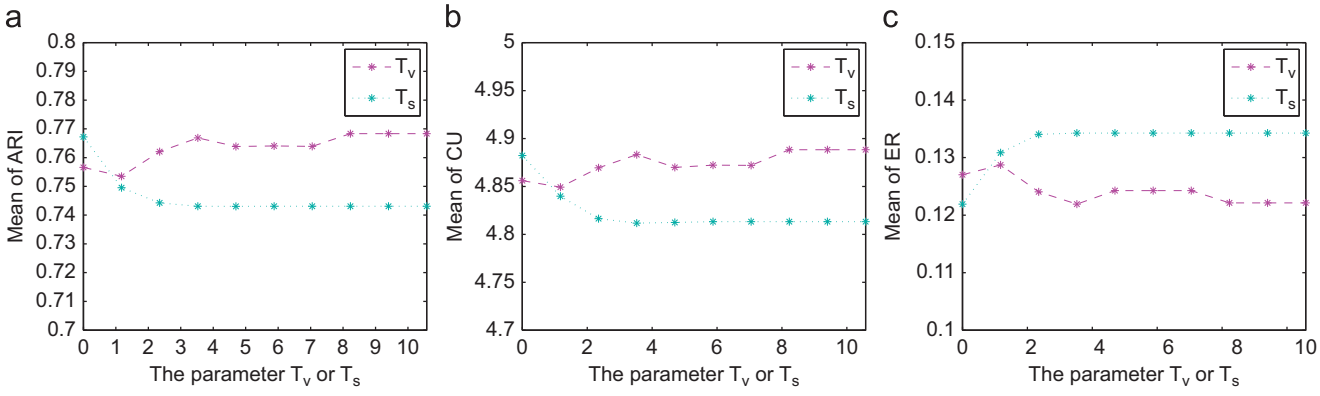

Fig. 9. (a) Means of ARI with respect to different values of T_v or T_s on the soybean data. (b) Means of CU with respect to different values of T_v or T_s on the soybean data. (c) Means of ER with respect to different values of T_v or T_s on the soybean data.

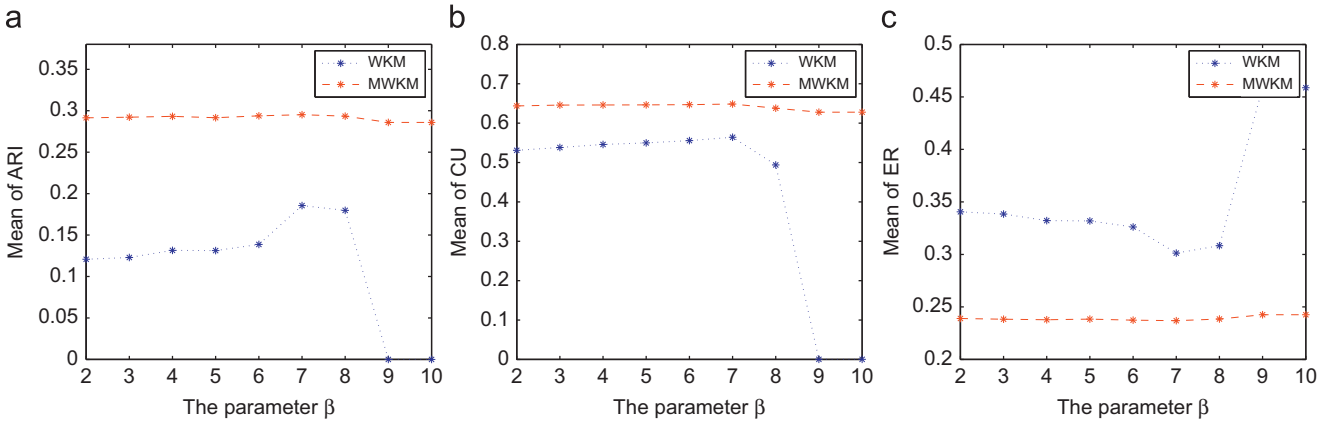

Fig. 10. (a) Means of ARI with respect to different values of β on the heart disease data. (b) Means of CU with respect to different values of β on the heart disease data. (c) Means of ER with respect to different values of β on the heart disease data.

which the next move is not forced. This data set contains 67,557 instances and 42 categorical attributes. It has three class: win (44,473), loss (16,635) and draw (6449). Fig. 18(a) shows the computational times against the number of objects, while the number of attributes is 42 and the number of clusters is 3. Fig. 18(b) shows the computational times against the number of attributes, while the number of clusters is 3 and the number of objects is 680,000. Fig. 18(c) shows the computational times against the number of clusters, while the number of attributes is 42 and the number of objects is 680,000.

6.2.2. Census data

The census data has 2,458,284 records with 68 categorical attributes, about 352 megabytes in total. It was derived from the

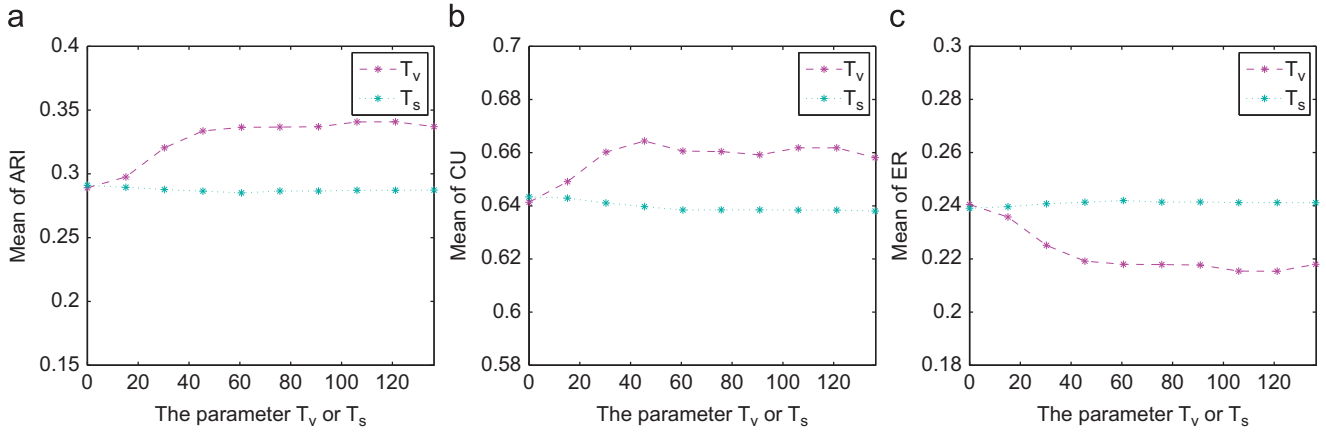

Fig. 11. (a) Means of ARI with respect to different values of T_v or T_s on the heart disease data. (b) Means of CU with respect to different values of T_v or T_s on the heart disease data. (c) Means of ER with respect to different values of T_v or T_s on the heart disease data.

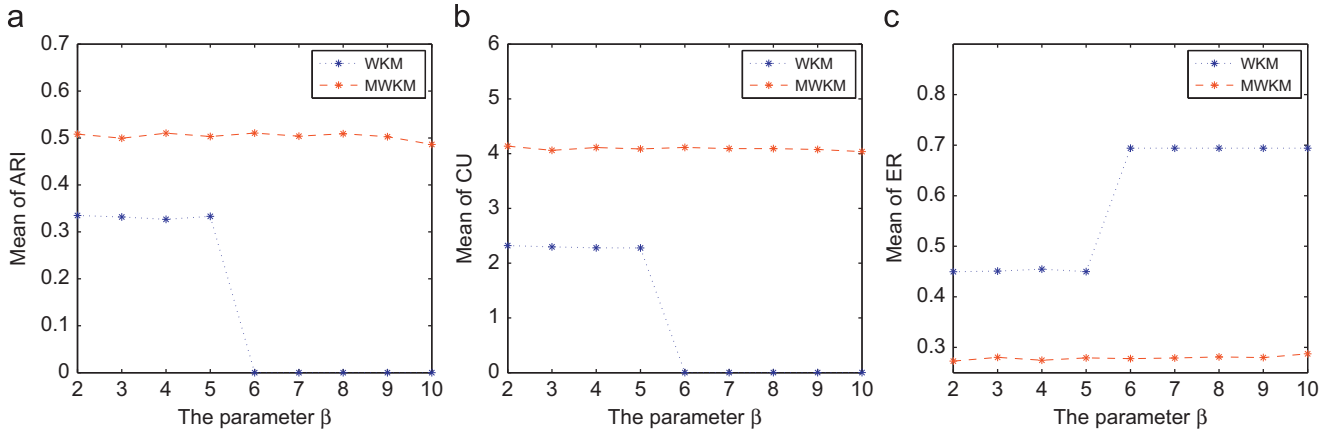

Fig. 12. (a) Means of ARI with respect to different values of β on the dermatology data. (b) Means of CU with respect to different values of β on the dermatology data. (c) Means of ER with respect to different values of β on the dermatology data.

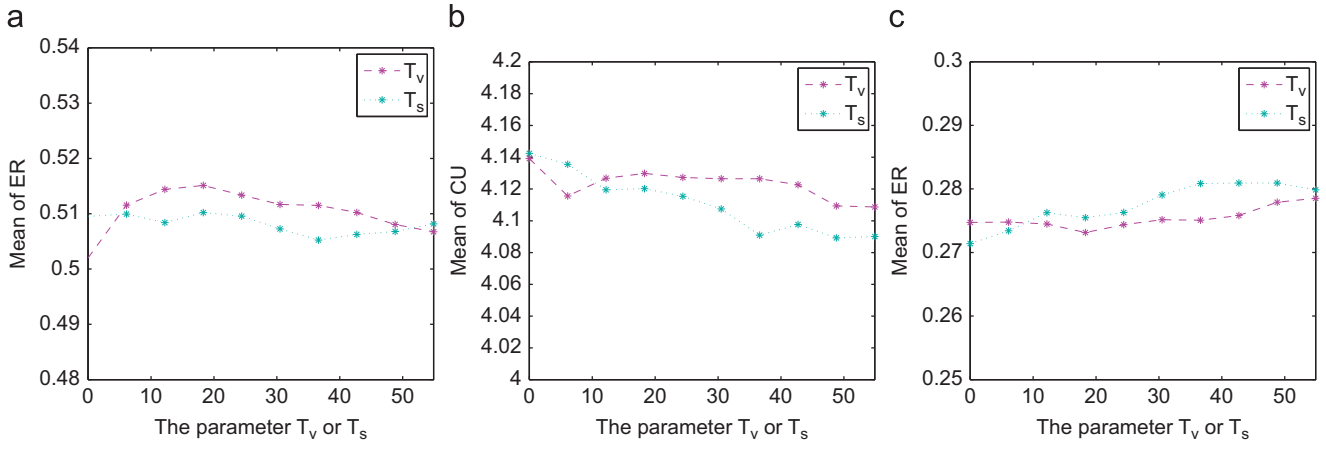

Fig. 13. (a) Means of ARI with respect to different values of T_v or T_s on the dermatology data. (b) Means of CU with respect to different values of T_v or T_s on the dermatology data. (c) Means of ER with respect to different values of T_v or T_s on the dermatology data.

USCensus1990raw data set which was obtained from the (U.S. Department of Commerce) Census Bureau website using the Data Extraction System. We take 100,000 records from this data set to test the scalability of the MWKM algorithm. Fig. 19(a) shows the computational times against the number of objects, while the number of attributes is 68 and the number of clusters is 3. Fig. 19(b) shows the computational times against the number of attributes, while the number of clusters is 3 and the number of objects is 100,000. Fig. 19(c) shows the computational times against the number of clusters, while the number of attributes is 68 and the number of objects is 100,000.

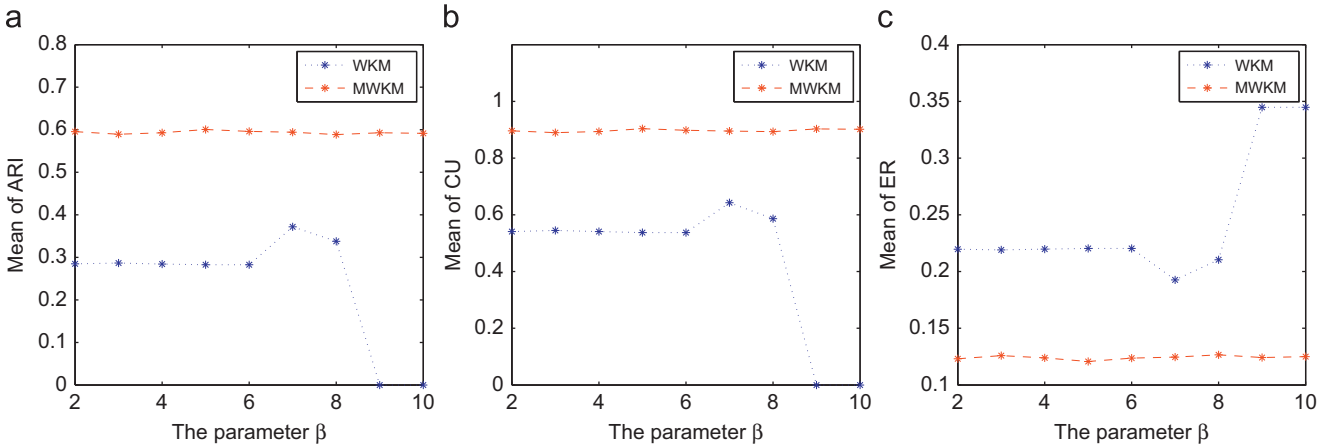

Fig. 14. (a) Means of ARI with respect to different values of β on the breast cancer data. (b) Means of CU with respect to different values of β on the breast cancer data. (c) Means of ER with respect to different values of β on the breast cancer data.

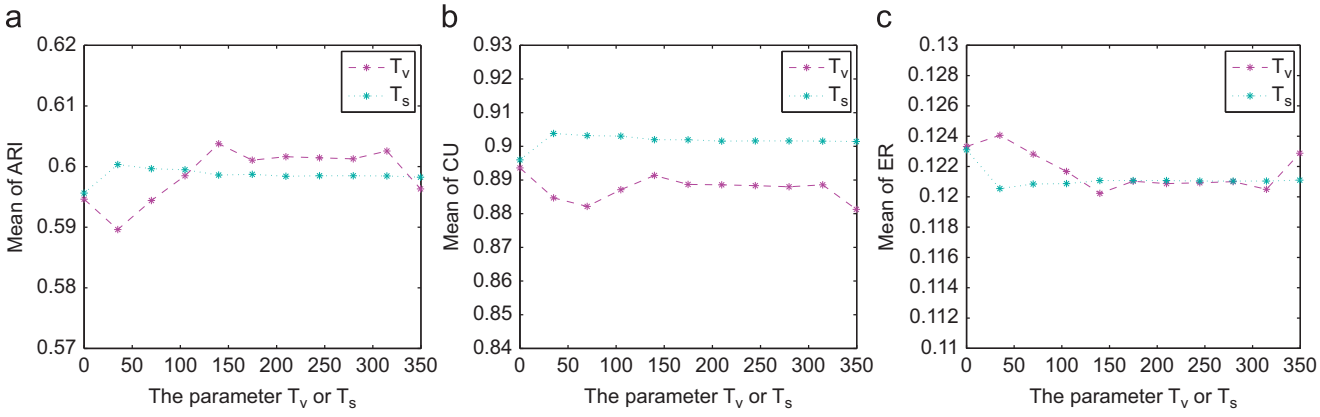

Fig. 15. (a) Means of ARI with respect to different values of T_v or T_s on the breast cancer data. (b) Means of CU with respect to different values of T_v or T_s on the breast cancer data. (c) Means of ER with respect to different values of T_v or T_s on the breast cancer data.

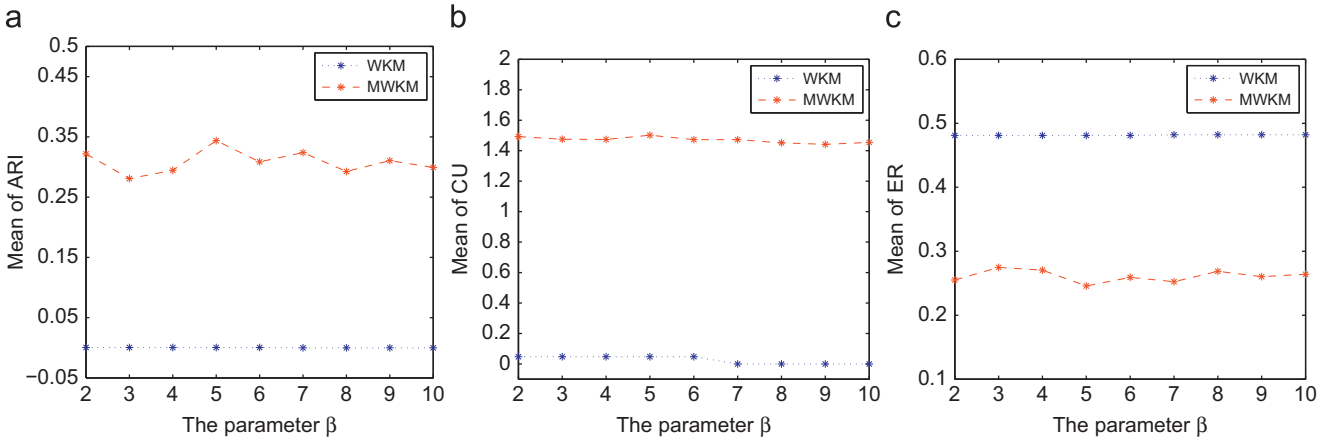

Fig. 16. (a) Means of ARI with respect to different values of β on the mushroom data. (b) Means of CU with respect to different values of β on the mushroom data. (c) Means of ER with respect to different values of β on the mushroom data.

According to Figs. 18 and 19, we see that the four algorithms are scalable, i.e., the computational times increase linearly with respect to the number of objects, attributes, or clusters. The MWKM algorithm requires slightly more computational time than other *k*-modes algorithms. It is an expected outcome since the calculation of the weighting information requires some additional arithmetic operations. However, according to the tests, the computational time of the MWKM algorithm is still scalable, i.e., it can cluster large categorical data efficiently.

7. Conclusions

In this paper, we have presented MWKM, a mixed attribute weighting algorithm for high-dimensional categorical data which is an extension of the *k*-modes algorithm. In this algorithm, a new weighted dissimilarity measure has been proposed to eliminate the effect of a problem that the attribute weight does not work while the comparative result of an object and a cluster center in an attribute is 0. Moreover, it has been integrated with Chan's

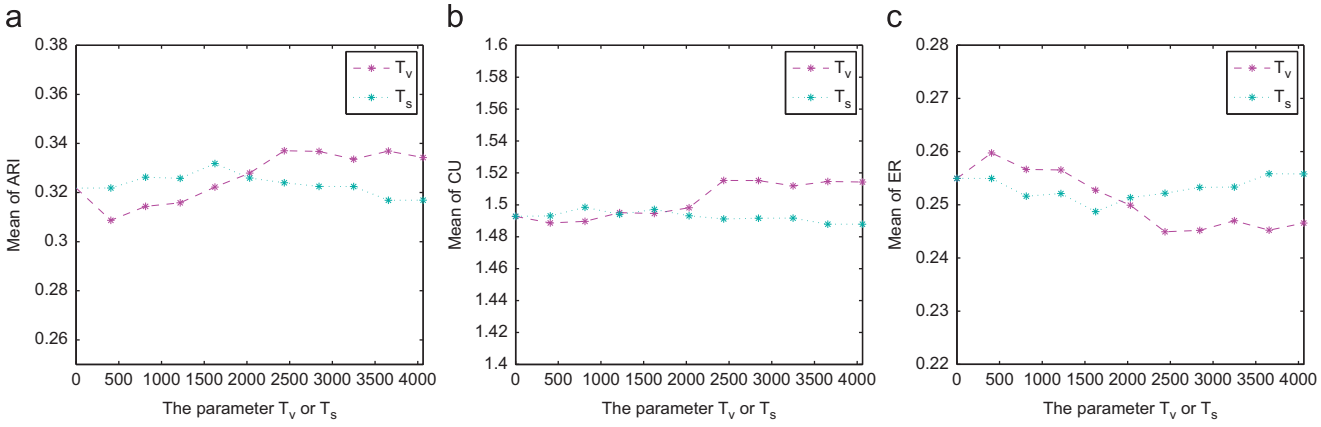

Fig. 17. (a) Means of ARI with respect to different values of T_v or T_s on the mushroom data. (b) Means of CU with respect to different values of T_v or T_s on the mushroom data. (c) Means of ER with respect to different values of T_v or T_s on the mushroom data.

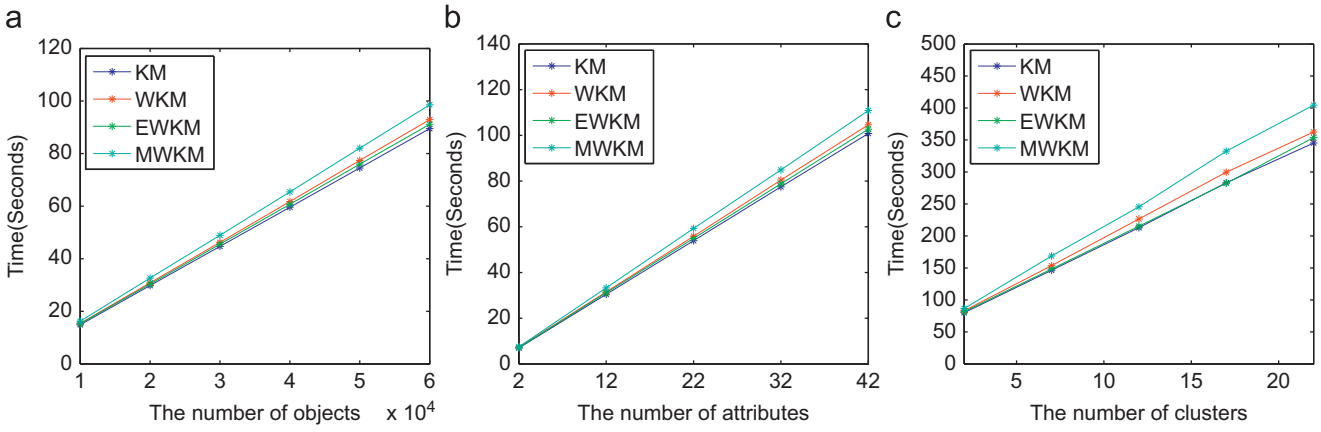

Fig. 18. (a) Computational times for different numbers of objects on the connect-4 data. (b) Computational times for different numbers of attributes on the connect-4 data. (c) Computational times for different numbers of clusters on the connect-4 data.

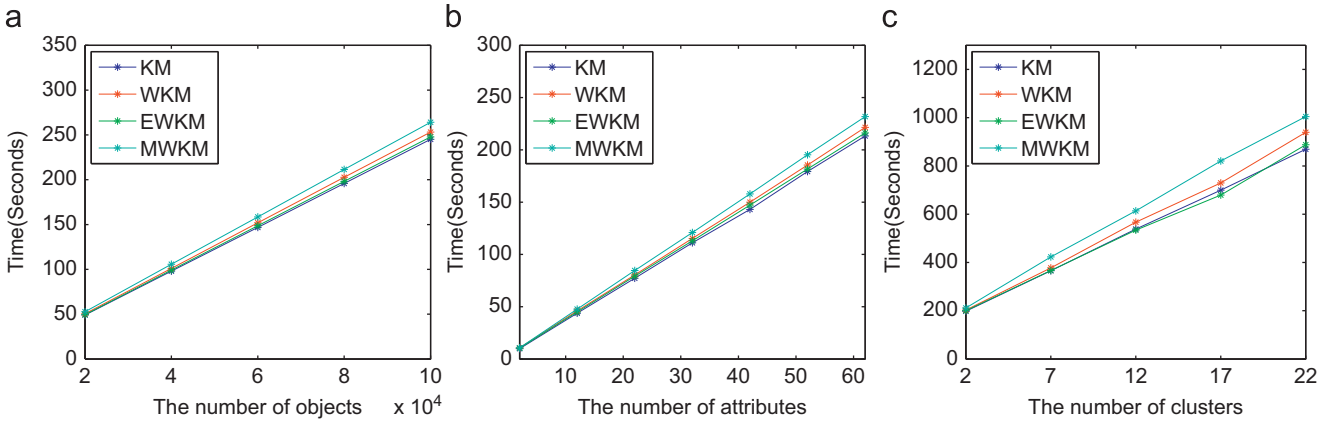

Fig. 19. (a) Computational times for different numbers of objects on the census data. (b) Computational times for different numbers of attributes on the census data. (c) Computational times for different numbers of clusters on the census data.

weighted dissimilarity measure to form a mixed weighted dissimilarity measure which has been applied to the proposed algorithm. We have rigorously derived the updating formulas of the MWKM algorithm and proved the convergence of the algorithm under the optimization framework. Experimental results show that the proposed algorithm is effective and efficient in clustering high-dimensional categorical data sets.

Acknowledgment

The authors are very grateful to the editors and reviewers for their valuable comments and suggestions. This work was supported by the National Natural Science Foundation of China (Nos. 71031006 and 70971080), GRF: CityU 112809 of Hong Kong SAR Government, the National Key Basic Research and Development Program of China (973) (No. 2007CB311002), the Foundation of Doctoral Program Research of Ministry of Education of China (No. 20101401110002), the Natural Science Foundation of Shanxi (No. 2010021016-2).

References

[1] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.
[2] D.H. Fisher, Knowledge acquisition via incremental conceptual clustering, Machine Learning 2 (2) (1987) 139–172.
[3] S. Guha, R. Rastogi, S. Kyuseok, ROCK: a robust clustering algorithm for categorical attributes, in: Proceedings of the 15th International Conference on Data Engineering, Sydney, Australia, vols. 23–26, 1999, pp. 512–521.
[4] V. Ganti, J.E. Gekhre, R. Ramakrishnan, CACTUS-clustering categorical data using summaries, in: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining1999, pp. 73–83.
[5] D. Barbara, Y. Li, J. Couto, Coolcat: an entropy-based algorithm for categorical clustering, in: Information and Knowledge Management2002, pp. 582–589.
[6] Z.X. Huang, A fast clustering algorithm to cluster very large categorical data sets in data mining, in: Proceedings of SIGMOD Workshop Research Issues on Data Mining and Knowledge Discovery1997, pp. 1–8.
[7] 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.
[8] K.K. Chen, L. Liu, "Best K": critical clustering structures in categorical datasets, Knowledge and Information Systems 20 (1) (2008) 1–33.
[9] F.Y. Cao, J.Y. Liang, L. Bai, A new initialization method for categorical data clustering, Expert Systems with Applications 33 (7) (2009) 10223–10228.
[10] F.Y. Cao, J.Y. Liang, L. Bai, A framework for clustering categorical time-evolving data, IEEE Transactions on Fuzzy Systems 18 (5) (2010) 872–882.
[11] T.F. Cox, M.A.A. Cox, Multidimensional Scaling, Chapman and Hall, 2001.
[12] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, in: Proceedings of the ACM SIGMOD International Conference on Management of Data1998, pp. 94–105.
[13] L. Parsons, E. Haque, H. Liu, Subspace clustering for high dimensional data: a review, ACM SIGKDD Explorations Newsletters 6 (1) (2004) 90–105.
[14] K. Mali, S. Mitra, Clustering and its validation in a symbolic framework, Pattern Recognition Letters 24 (2003) 2367–2376.
[15] K. Kailing, H.P. Kriegel, P. Kroger, Density-connected subspace clustering for high-dimensional data, in: Proceedings of the 4th SIAM International Conference on Data Mining (SDM)2004.
[16] A. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artificial Intelligence 97 (1997) 245–271.
[17] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, 1998.
[18] Y.H. Qian, J.Y. Liang, W. Pedrycz, C.Y. Dang, Positive approximation: an accelerator for attribute reduction in rough set theory, Artificial Intelligence 174 (5–6) (2010) 597–618.
[19] Q.H. Hu, Z.X. Xie, D.R. Yu, Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation, Pattern Recognition 40 (2007) 3509–3521.
[20] S.C. Madeira, A.L. Oliveira, Biclustering algorithms for biological data analysis: a survey, IEEE Transactions on Computational Biology and Bioinformatics 1 (2004) 24–45.
[21] S. Busygin, O. Prokopyev, P.M. Pardalos, Biclustering in data mining, Computers and Operations Research 35 (2008) 2964–2987.
[22] R.G. Pensa, C. Robardet, J.F. Boulicaut, A bi-clustering framework for categorical data, in: Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'05)2005, pp. 643–650.
[23] C.H. Cheng, A.W. Fu, Y. Zhang, Entropy-based subspace clustering for mining numerical data, in: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge and Data Mining1999, pp. 84–93.
[24] S. Goil, H. Nagesh, A. Choudhary, Mafia: efficient and scalable subspace clustering for very large data sets, Technical Report CPDC-TR-9906-010, Northwest University, 1999.
[25] C. Aggarwal, C. Procopiuc, J.L. Wolf, P.S. Yu, J.S. Park, Fast algorithms for projected clustering, in: Proceedings of the ACM SIGMOD International Conference on Management of Data1999, pp. 61–72.
[26] C.C. Aggarwal, P.S. Yu, Finding generalized projected clusters in high dimensional spaces, in: Proceedings of the ACM SIGMOD International Conference on Management of Data2000, pp. 70–81.
[27] K.G. Woo, J.H. Lee, Find it: a fast and intelligent subspace clustering algorithm using dimension voting, Ph.D. Dissertation, Korea Advanced Institute of Science and Technology, 2002.
[28] C.M. Procopiuc, M. Jones, P.K. Agarwal, T.M. Murali, A Monte Carlo algorithm for fast projective clustering, in: Proceedings of the ACM SIGMOD Conference on Management of Data2002, pp. 418–427.
[29] J. Yang, W. Wang, H. Wang, P. Yu, D-clusters: capturing subspace correlation in a large data set, in: Proceedings of the 18th International Conference on Data Engineering2002, pp. 517–528.
[30] K.Y. Yip, D.W. Cheung, M.K. Ng, A practical projected clustering algorithm, IEEE Transactions on Knowledge and Data Engineering 16 (11) (2004) 1387–1397.
[31] K. Chakrabarti, S. Mehrotra, Local dimensionality reduction: a new approach to indexing high dimensional spaces, in: Proceedings of the 26th International Conference on Very Large Data Bases2000, pp. 89–100.
[32] G. Gan, J. Wu, Subspace clustering for high dimensional categorical data, ACM SIGKDD Explorations Newsletters 6 (2) (2004) 87–94.
[33] M. Zaki, M. Peters, I. Assent, T. Seidl, CLICK: an effective algorithm for mining subspace clusters in categorical datasets, Data and Knowledge Engineering 60 (2007) 51–70.
[34] E. Cesario, G. Manco, R. Ortale, Top-down parameter-free clustering of high-dimensional categorical data, IEEE Transactions on Knowledge and Data Engineering 19 (12) (2007) 1607–1624.
[35] G. De Soete, Optimal variable weighting for ultrametric and additive tree clustering, Quality and Quantity 20 (1986) 169–180.
[36] V. Makarenkov, P. Legendre, Optimal variable weighting for ultrametric and additive trees and *k*-mean spartitioning: methods and software, Journal of Classification 18 (2001) 245–271.
[37] Z.X. Huang, M.K. Ng, H. Rong, Z. Li, Automated variable weighting in *k*-means type clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (5) (2005) 657–668.
[38] Y. Chan, W. Ching, M.K. Ng, Z.X. Huang, An optimization algorithm for clustering using weighted dissimilarity measures, Pattern Recognition 37 (5) (2004) 943–952.
[39] H. Frigui, O. Nasraoui, Unsupervised learning of prototypes and attribute weights, Pattern Recognition 37 (3) (2004) 567–581.
[40] C. Domeniconi, D. Papadopoulos, D. Gunopulos, S. Ma, Subspace clustering of high dimensional data, in: Proceedings of the SIAM International Conference on Data Mining2004.
[41] J.H. Friedman, J.J. Meulman, Clustering objects on subsets of attributes, Journal of the Royal Statistical Society 66 (4) (2004) 815–849.
[42] L.P. Jing, M.K. Ng, Z.X. Huang, An entropy weighting *k*-means algorithm for subspace clustering of high-dimensional sparse data, IEEE Transactions on Knowledge and Data Engineering 19 (8) (2007) 1026–1041.
[43] G.J. Gan, J.H. Wu, Z.J. Yang, A fuzzy subspace algorithm for clustering high dimensional data, in: X. Li, O. Zaiane, Z. Li (Eds.), Lecture Notes in Artificial Intelligence, vol. 4093, Springer, Berlin, 2006, pp. 271–278.
[44] G.J. Gan, J.H. Wu, A convergence theorem for the fuzzy subspace clustering (FSC) algorithm, Pattern Recognition 41 (2008) 1939–1947.
[45] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, D. Papadopoulos, Locally adaptive metrics for clustering high dimensional data, Data Mining and Knowledge Discovery 14 (2007) 63–97.
[46] J.C. Bezdek, A convergence theorem for the fuzzy ISODATA clustering algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence 2 (1980) 1–8.
[47] S. Miyamoto, M. Mukaidono, Fuzzy c-means as a regularization and maximum entropy approach, Proceedings of the 7th International Fuzzy Systems Association World Congress, vol. 2, 1997, pp. 86–92.
[48] J. Yu, H. Shi, H. Huang, et al., Counterexamples to convergence theorem of maximum-entropy clustering algorithm, Science in China, Series F: Information Sciences 46 (2003) 321–326.
[49] S.J. Ren, Y.D. Wang, A proof of the convergence theorem of maximum-entropy clustering algorithm, Science in China, Series F: Information Sciences 53 (2010) 1151–1158.
[50] Z.H. Deng, K.S. Choi, F.L. Chung, ShitongWang, Enhanced soft subspace clustering integrating within-cluster and between-cluster information, Pattern Recognition 43 (2010) 767–781.
[51] Z. Pawlak, Rough Sets-Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht, Boston, London, 1991.
[52] J.Y. Liang, D.Y. Li, Uncertainty and Knowledge Acquisition in Information Systems, Science Press, Beijing, China, 2005.
[53] 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.
[54] K.C. Gowda, E. Diday, Symbolic clustering using a new dissimilarity measure, Pattern Recognition 24 (6) (1991) 567–578.
[55] UCI Machine Learning Repository 〈http://www.ics.uci.edu~/mlearn/MLRepository.html〉, 2010.

[56] M. Zait, H. Messatfa, A comparative study of clustering methods, Future Generation Computer Systems 13 (1997) 149–159.
[57] M.A. Gluck, J.E. Corter, Information uncertainty and the utility of categories, in: Proceedings of the Seventh Annual Conference of Cognitive Science Society1985, pp. 283–287.
[58] L. Hubert, P. Arabie, Comparing partitions, Journal of Classification 2 (1) (1985) 193–218.
[59] Y.M. Yang, An evaluation of statistical approaches to text categorization, Journal of Information Retrieval 1 (1–2) (1999) 67–88.

Liang Bai is a Ph.D. candidate from the School of Computer and Information Technology at Shanxi University, China. He received his M.S. degree in computer science from Shanxi University in 2009. He is currently a Research Assistant in the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong. His research interests are in the areas of data mining and machine learning.

Jiye Liang received the M.S. and Ph.D degrees from Xi'an Jiaotong University, Xi'an, China, in 1990 and 2001, respectively. He is currently a Professor with the School of Computer and Information Technology and the Key Laboratory of Computational Intelligence and Chinese Information Processing of the Ministry of Education, Shanxi University, Taiyuan, China. He has authored or coauthored more than 70 journal papers in his research fields. His current research interests include computational intelligence, granular computing, data mining, and knowledge discovery.

Chuangyin Dang received the Ph.D. degree in operations research/economics from the University of Tilburg, Tilburg, the Netherlands, in 1991 and the M.S. degree in applied mathematics from Xidian University, Xi'an, China, in 1986. He is currently an Associate Professor with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong. His research interests include computational intelligence and optimization theory and technology.

Fuyuan Cao received the M.S. and Ph.D degrees in computer science in 2004 and 2009, respectively, from Shanxi University, Taiyuan, China, where he is currently an Associate Professor with the School of Computer and Information Technology. His research interests include data mining and machine learning.