Soft subspace clustering of categorical data with probabilistic distance

Lifei Chen\textsuperscript{a,b,*}, Shengrui Wang\textsuperscript{c}, Kajun Wang\textsuperscript{a,b}, Jianping Zhu\textsuperscript{d}

\textsuperscript{a} School of Mathematics and Computer Science, Fujian Normal University, Fujian 350017, China
\textsuperscript{b} Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fujian 350017, China
\textsuperscript{c} Department of Computer Science, University of Sherbrooke, Sherbrooke, Quebec, Canada J1K 2R1
\textsuperscript{d} School of Management and Data-Mining Research Center, Xiamen University, Xiamen 361005 China

A R T I C L E  I N F O

Article history:
Received 9 May 2015
Received in revised form 19 August 2015
Accepted 24 September 2015
Available online 3 October 2015

Keywords:
Subspace clustering
Categorical data
Distance measure
Attribute weighting
Kernel density estimation

A B S T R A C T

Categorical data clustering is an important subject in pattern recognition. Currently, subspace clustering of categorical data remains an open problem due to the difficulties in estimating attribute interestingness according to the statistics of categories in clusters. In this paper, a new algorithm is proposed for clustering categorical data with a novel soft feature-selection scheme, by which each categorical attribute is automatically assigned a weight that correlates with the smoothed dispersion of the categories in a cluster. In the proposed algorithm, dissimilarity between categorical data objects is measured using a probabilistic distance function, based on kernel density estimation for categorical attributes. We also make use of the probabilistic distances to define a cluster validity index for estimating the number of categorical clusters. The suitability of the proposal is demonstrated in an empirical study done with some widely used real-world data sets and synthetic data sets, and the results show its outstanding performance.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Currently, categorical data are widely used in many real-world applications. For example, in bioinformatics, the nucleotides composing a DNA sequence are each encoded in one of the four nominal categories A, G, T and C. Similarly, in medical diagnosis, patient data usually contains numerous nominal attributes describing patient profiles, and ordinal indices measuring their physiological parameters. Many applications like these generate complex data sets involving attributes of the nominal and/or ordinal types, which remain a major obstacle in applying the common pattern recognition methods [1–3].

Subspace clustering of such data, which aims at grouping data objects into clusters projected in some subspaces, is an essential process in many applications, especially with high-dimensional data. For DNA sequences, for instance, a subspace clustering algorithm can be used to recognize hidden patterns (clusters) in the sequences and, more importantly, to reveal the underlying biological concepts of interest by performing automated variable selection in an unsupervised manner. Compared to numeric data, for which numerous clustering methods have been developed [4–6], categorical data pose a unique challenge in subspace clustering tasks, due to the difference between the two data types.

Unlike the numeric case, when the data are categorical the set mean, i.e., the averaging value of a finite set of data objects, is an undefined concept. This means that the common K-means-type methods cannot be directly used for categorical data clustering. A natural solution to the problem is to transform categorical attributes into binaries by using 0 and 1 to represent the absence or presence of a discrete symbol, and then treat them like numeric data [7]. However, such a transformation typically results in a large number of correlated attributes (the transformed binaries of the same original attribute are clearly correlated with each other), which complicates the clustering model. A popular extension to the traditional K-means [1] is to represent the "mean" of a categorical cluster using the mode category on each attribute [2,3,8]. Such a method encounters difficulties when there is no dominating category distributed on the attribute because, statistically speaking, the mode-based approach can only capture partial information on the data objects in a cluster [9].

Another challenging issue is automated variable selection, used to identify the “interesteningness” of particular attributes during the subspace clustering process. Though automated variable selection has been stressed extensively in numeric data clustering [4,5,10], few attempts have been made to apply it to categorical data clustering. The main obstacle lies in the difficulty of estimating attribute interestingness based on the statistics of categories in a cluster. The measures that have been successfully used for numeric attributes, such as the popular variance of numeric data, are not well-defined for categorical data [11]. A few alternatives, such as the Gini diversity index [12] and the...
complement entropy [13], have been suggested for the purpose; however, how they connect to attribute interestingness also remains unsolved in categorical data clustering.

In this paper, we propose a novel approach to these problems using a kernel density estimation method for categorical attributes, by which a probabilistic distance function is defined to measure the distance to clusters of categorical data objects. Based on the definition, a new subspace clustering algorithm, called SCC (for Subspace Clustering of Categories), is proposed to minimize the sum of object-to-cluster distances, without representing a categorical cluster by its mode categories. Moreover, a weight calculation formula is derived to estimate the individual interest of categorical attributes based on the kernel-smoothed dispersion of the categories in a cluster. In particular, we show that the common attribute-weighting approach – i.e., that attribute weights should be computed as being inversely proportional to the sample dispersion – does not always hold for categorical data clustering, especially when the data involve ordinal attributes. We also propose a cluster validity index to measure clustering quality and estimate the number of clusters in a categorical data set. A series of experiments on real-world and synthetic categorical data have been conducted to support the conclusions.

The remainder of this paper is organized as follows: Section 2 presents some related work. In Section 3, the new clustering algorithm is presented. Experimental results are presented in Section 4. Section 5 gives our conclusion and discusses directions for future work.

2. Related work

In this section, a sampling of related work on subspace clustering of categorical data is described. Generally, the goal of a subspace clustering algorithm is to discover a set of subspace clusters, where each cluster is a subset of data objects joined with a subset of attributes spanning its projected subspace. Typically, the clusters are associated with different subspaces comprised of different combinations of attributes in the original data space [14]. Such a class-dependent projection is common in many real applications, especially with high-dimensional data, where typically the same attribute contributes individually to different data subsets. For example, in the task of subspace clustering of DNA sequences [9], a nucleotide that is important at a particular position for one pattern may be useless for the others.

Technically, subspace clustering is performed by embedding an attribute-weighting procedure in the clustering process [14,15], where each attribute of each cluster is assigned a weighting value, indicating to what extent the attribute is relevant to the cluster. On the basis of the way the weights are determined, subspace clustering algorithms can be divided into two groups: hard subspace clustering and soft subspace clustering algorithms. In the first group, which includes such algorithms as [16] and [17], attributes of each cluster are assigned weighting values of either 0 or 1. SUBCAD [17] is one of the representatives in this group, which aims at minimizing the clustering objective function by relocating data objects in different subsets and updating their associated subspaces. As there is a large number of candidate subspaces needed to examine, generally, such an algorithm has a high time complexity.

The weights can be any real number in [0,1] for the algorithms in the second group. In effect, assigning such continuous weights to attributes is equivalent to performing a soft feature selection for clusters [5,10]. Owing to its computational efficiency, currently, soft subspace clustering has become one of the mainstream methods for high-dimensional data clustering. Examples include EWKM [5], LAC [10], Essc [18] and many others. It is important to remark that, virtually all of these algorithms in this group compute the attribute weight as being inversely proportional to the dispersion of the values [46], which, typically, is measured by the variance of the numeric attribute. As they are originally designed for numeric data, when applied to nominal data, the attributes must be converted into binaries in advance, as mentioned in the previous section. For ordinal attributes, to some extent, they can be applied directly based on the assumption that the disparities between any two successive symbols are identical. However, the assumption is highly domain-dependent and is often unrealistic in real applications.

A few algorithms have been proposed for soft subspace clustering directly on categorical data, including [2,3,13]. In terms of algorithmic structure, they can be viewed as an extension to the classical K-modes [8], which is a K-means-type algorithm representing the cluster center by the mode category on each categorical attribute. The algorithms differ from each other by their attribute-weighting methods used to optimize the weights, and the methods fall into two groups. The algorithms in the first group, such as [2,3,19], compute the weight according to the frequency of the mode category. Those in the second group assign the weights using heuristic measures that correlate with the category distribution. For example, in the recently published work [13], the weights are computed in terms of the complement entropy of the categorical attributes; however, the entropy is computed on the mode-based clusters in this algorithm. Due to the problem of the mode-based representation, as mentioned previously, the weights produced by these algorithms easily yield a biased indication of the importance of attributes to clusters.

In the proposed SCC algorithm of this paper, as described below, categorical attributes are weighted according to the dispersion of the categories, more than the mode category. Basically, the weighting scheme is consistent with the method commonly used to weight a numeric attribute, as discussed above. On the other hand, we show that, in categorical data clustering, the weights might not necessarily be measured being inversely proportional to the dispersion.

3. Probabilistic distance-based subspace clustering

In this section, a new algorithm is proposed for soft subspace clustering on categorical data, followed by a discussion on the attribute-weighting scheme and a new cluster validity index defined for evaluating clustering quality as a function of the number of categorical clusters. We will begin by defining the clustering criterion that needs to be optimized by the algorithm.

3.1. Clustering criterion

We first introduce notation used throughout the paper. In what follows, the data set is denoted by \( \mathbf{DB}=\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \) from which \( K \) subspace clusters are searched for. Here \( \mathbf{x}_i=(x_{i1}, x_{i2}, \ldots, x_{id}) \) for \( i=1,2,\ldots,N \) are \( D \)-dimensional data objects, and let \( \mathbf{x}=(x_1, x_2, \ldots, x_d) \) be an arbitrary object. For the \( d \)th attribute, where \( d=1,2,\ldots,D \), we denote the set of categories by \( O_d; i.e., \), the attribute \( d \) takes \( |O_d| \) discrete values. An arbitrary category in the set is denoted by \( a_{dI} \in O_d \), where \( I \in [1,|O_d|] \). We also denote the subset of data objects in the \( k \)th cluster (\( k=1,2,\ldots,K \)) by \( \pi_k \), with \( |\pi_k| \) being its number of objects, such that \( \cup_{k=1}^K \pi_k=\mathbf{DB} \) and \( \sum_{k=1}^K |\pi_k|=N \). Unless otherwise specified, we will call \( \pi_k \) the \( k \)th cluster.

The cluster \( \pi_k \) is associated with a weight vector \( \mathbf{w}_k=(w_{k1}, w_{k2}, \ldots, w_{kd}) \), satisfying

\[
\begin{align*}
\sum_{d=1}^{D} w_{kd} &= 1, \quad k=1,2,\ldots,K \\
0 &\leq w_{kd} \leq 1, \quad k=1,2,\ldots,K; \quad d=1,2,\ldots,D.
\end{align*}
\]
Moreover, we consider that the exponential smoothing weight \( w_{kd}^\theta \) represents the contribution of attribute \( d \) in forming the cluster \( \pi_k \), with the weighting exponent \( \theta \neq 0 \) controlling the strength of the incentive for clustering on more attributes. The greater the contribution, the higher the smoothed weight. Based on these, we denote the \( k \)th subspace cluster of DB by \( (\pi_k, w_k) \), given that \( \theta \) is a constant that is independent of \( w_k \).

Given DB and \( K \), generally, the goal of soft subspace clustering is to obtain an optimal set of \( K \) subspace clusters in order to minimize the weighted within-cluster scatter \([4,10]\). We define the scatter based on object-to-cluster distances for each data object in a cluster, instead of the object-to-mode distances as used in the \( K \)-modes and its variants \([2,3,8,13]\). In this way, we can avoid the problem of representing clusters by mode categories. Letting \( D_{sk}(x, \pi_k) \) be the object-to-cluster distance of \( x \) to the \( k \)th cluster on attribute \( d \), the clustering criterion is written as follows:

\[
\min \{ D(\Pi, W) = \sum_{k=1}^{K} \sum_{d=1}^{D} w_{kd}^\theta \times D_{sk}(x, \pi_k) \}
\]

subject to Eq. (1), where \( \Pi = \{\pi_k|k = 1, 2, ..., K\} \) and \( W = \{w_k|k = 1, 2, ..., K\} \).

In Eq. (2), the term \( \sum_{d=1}^{D} w_{kd}^\theta \times D_{sk}(x, \pi_k) \) actually measures the dissimilarity of each data object \( x \) in the \( k \)th subspace cluster to all the objects in the same cluster. In the \( K \)-modes-type clustering algorithms \([2,13]\), however, the term is computed as the weighted distance between \( x \) and the mode categories of the cluster on each attribute. Obviously, such a method for distance computation easily leads to the loss of non-mode categories in the cluster. In the next subsection, we formulate \( D_{sk}(x, \pi_k) \) as a probabilistic distance measure.

### 3.2. Probabilistic distance

Computing the distance between \( x \) and \( \pi_k \) is not straightforward like the Euclidean distance for numeric data, because the data can only take discrete values \([20]\). To numerically compute the distance, we present a space-transformation method, i.e., by representing each categorical object in a probability space and then measuring the distance in the probability space. Let \( X_0 \) be a random variable associated with the observations for attribute \( d \), and \( p \) be the probability measure defined on a Borel set of the sample space \( O_d \). In the probability space, the observation \( x_0 \) of \( X_0 \) corresponds to the conditional probability distribution \( p(X_0|x_0) \), while the distribution of \( X_0 \) with regard to the \( k \)th cluster can be represented as \( p(X_0|\pi_k) \). Now, we compute the distance between \( x \) and \( \pi_k \) on attribute \( d \) by measuring the Euclidean distance of the two discrete probability distributions, yielding

\[
D_{sk}(x, \pi_k) = \sum_{a_d \in O_d} [p(X_0 = a_d|x_0) - p(X_0 = a_d|\pi_k)]^2.
\]

Based on the common assumption that all the categories in \( O_d \) are independent of each other, the probability \( p(X_0 = a_d|x_0) \) is estimated as

\[
p(X_0 = a_d|x_0) = I(a_d = x_0)
\]

where \( I() \) is an indicator function with \( I(true) = 1 \) and \( I(false) = 0 \).

To estimate \( p(X_0|\pi_k) \), we use the kernel density estimation (KDE) method \([21,22]\), which is a common non-parametric estimation method for a finite sample set. With the KDE method, the probability density of \( X_0 \) is defined by the kernel functions \( \kappa(X_0|x_0; \lambda_k) \), where \( a_d \in O_d \) and \( \lambda_k \) is the smoothing parameter called bandwidth; the probability density can then be given as

\[
p(X_0 = a_d|\pi_k) = 1/n_k \sum_{x \in \pi_k} \kappa(X_0|x_0; \lambda_k).
\]

Using the kernel function presented in \([9]\), i.e.,

\[
k(X_0|x_0; \lambda_k) = \begin{cases} 
1 - \frac{1}{2\lambda_k} - \frac{1}{2\lambda_k^2} & \text{if } x_0 = a_d, \\
\frac{1}{\lambda_k^2} & \text{if } X_0 \neq a_d.
\end{cases}
\]

with \( \lambda_k \in [0, 1] \) being the unique bandwidth for \( \pi_k \), we obtain

\[
p(X_0 = a_d|\pi_k) = \frac{1}{n_k \sum_{x \in \pi_k} \frac{1}{\lambda_k} \left(1 - \frac{1}{\lambda_k^2}\right)} \left(1 - \frac{1}{\lambda_k^2}\right) (f(a_d) = 0)
\]

where

\[
f(a_d|x_0) = \frac{1}{n_k \sum_{x \in \pi_k} 1(a_d = x_0)}
\]

is the frequency estimator of \( a_d \) with regard to \( \pi_k \).

Substituting \( p(X_0 = a_d|x_0) \) and \( p(X_0 = a_d|\pi_k) \) in \( D_{sk}(x, \pi_k) \) according to Eqs. (3) and (5), respectively, the distance measure becomes

\[
D_{sk}(x, \pi_k) = \sum_{a_d \in O_d} \left[1 + f(a_d) - \frac{1}{\lambda_k} \left(1 - \frac{1}{\lambda_k^2}\right) f(a_d)\right]^2
\]

with

\[
s_{sk}^2 = 1 - \sum_{a_d \in O_d} (f(a_d|x_0))^2
\]

being the Gini diversity index for the attribute \( d \) of \( \pi_k \). Note that the Gini diversity index \( s_{sk}^2 \in [0, (1/O_d - 1)/|O_d|] \) is useful for the sample dispersion of categorical attributes \([11,12]\). When the categories are uniformly distributed, \( s_{sk}^2 \) reaches its maximum.

We remark that Eq. (6) can be viewed as a generalization to the traditional object-to-cluster distance for numeric attributes. As discussed in Section 1, one can transform a categorical attribute into a set of binaries such that they can be treated like numeric data. In this case, each \( x_0 \) of \( x \) is \( \pi_k \) is alternatively represented as a vector \( y(x_0) = (I(a_1 = x_0), \ldots, I(a_d = x_0), \ldots, I(a_d = x_0), a_d = x_0) \). By the transformation, the cluster center on attribute \( d \) can be represented as \( \Psi(d) = (\sum f(a_d|x_0), \ldots, \sum f(a_d|x_0), \ldots, \sum f(a_d|x_0), a_d = x_0) \), like the method used in the \( K \)-representatives algorithm \([7]\) as well as in \([24]\). In this way, the object-to-cluster distance of \( x \) to \( \pi_k \) on that attribute changes to \( D_{sk}(x, \pi_k) = \|y(x_0) - \Psi(d)\|^2 \), where \( \| \cdot \| \) stands for the Euclidean norm of a vector. The distance can be rewritten as

\[
D_{sk}(x, \pi_k) = 1 + \frac{1}{\lambda_k} \left(1 - \frac{1}{\lambda_k^2}\right) f(a_d|x_0)^2 - 2 f(a_d|x_0) 1 - f(a_d|x_0) - s_{sk}^2, \text{ where clearly it is a special case of our distance measure Eq. (6) by fixing the bandwidth } \lambda_k \text{ at } 0. \text{ In our method, the value of } \lambda_k \text{ can be optimized according to the category distribution of the cluster, as described in the next subsection.}

### 3.3. Bandwidth optimization

The optimization of the bandwidth is an important issue for a KDE method, because it is the value of the bandwidth that dominates the probability distribution for a given sample set. In this subsection, we solve it by an automatic data-driven method designed to learn an optimal bandwidth that minimizes the total error of the resulting estimation.

Letting \( \hat{p}(a_d|\lambda_k) \) be the kernel estimator of \( p(a_d) \) by Eq. (4), the total error of the estimation is

\[
\sum_{d=1}^{D} \sum_{a_d \in O_d} \left[ \hat{p}(a_d|\lambda_k)^2 - 2 \sum_{d=1}^{D} \sum_{a_d \in O_d} \hat{p}(a_d|\lambda_k) p(a_d|\lambda_k) \right] \text{ which can be simplified into}
\]

\[
\phi(\lambda_k) = \sum_{d=1}^{D} \sum_{a_d \in O_d} \left[ \hat{p}(a_d|\lambda_k)^2 \right] - 2 \sum_{d=1}^{D} \sum_{a_d \in O_d} \hat{p}(a_d|\lambda_k) p(a_d|\lambda_k).
\]

Here, the constant \( \sum_{d=1}^{D} \sum_{a_d \in O_d} \hat{p}(a_d)^2 \) is irrelevant to \( \lambda_k \) is removed. The term \( \sum_{a_d \in O_d} \hat{p}(a_d|\lambda_k) p(a_d|\lambda_k) \) in \( \phi(\lambda_k) \) is the
expectation of $X C$; therefore, it can be estimated by the sample mean over all the observations [23]. Following [23], we replace the term with $1/n_k\sum_{x \in X_k} \tilde{p}(x_{ik}|x_{ik})$, where $\tilde{p}(x_{ik}|x_{ik}) = 1/(n_k - 1) \sum_{x \neq x_k} K(x_{ik}|x_{ik}, x_{ik})$ is the leave-one-out kernel estimator. Given that $\tilde{p}(x_{ik}|x_{ik}) = n_k/(n_k - 1) \tilde{p}(x_{ik}|x_{ik}) - 1/(n_k - 1)(1 - (1/D)|x_{ik}|)$ and $\tilde{p}(x_{ik}|x_{ik}) = 1/n_k \sum_{x \neq x_k} K(x_{ik}|x_{ik}, x_{ik})$ according to the kernel function defined in Eq. (4), the objective function becomes

$$\phi_k(\lambda_k) = \frac{n_k}{n_k - 1} \tilde{p}(x_{ik}|x_{ik}) - \lambda_k - \frac{1}{n_k - 1} \sum_{d=1}^{D} |O_d| - 1 - D.$$ \hspace{1cm} (8)

The optimal bandwidth of $\pi_k$, denoted by $\hat{\lambda}_k$, should minimize $\phi_k(\lambda_k)$ subject to $0 \leq \hat{\lambda}_k \leq 1$. Therefore, the gradient with respect to $\hat{\lambda}_k$ should vanish, i.e., $\partial \phi_k/\partial \hat{\lambda}_k = 0$, which yields

$$\hat{\lambda}_k = \frac{\sum_{d=1}^{D} \lambda_k^2}{n_k - 1} \sum_{d=1}^{D} (|O_d| - 1)^2.$$ \hspace{1cm} (9)

Note that $\hat{\lambda}_k$ lies within $[0, 1]$ unless $n_k = 1$ or $\lambda_k^2 > (1/D)|O_d|$, because $\forall d: 0 \leq \lambda_k^2 \leq (1/D)|O_d|$. For the two exceptions, which mean that the cluster consists of only one object or the categories on all attributes are uniformly distributed, we set $\hat{\lambda}_k = 1$ in practice. Furthermore, it can be seen that $\hat{\lambda}_k = 0$ only if all the attributes take a single categorical value such that $\lambda_k^2 > 0$. In this case, the distance defined by Eq. (6) degenerates to the object-to-cluster distance used for numeric data, as discussed in the previous subsection.

### 3.4. Clustering algorithm

This subsection presents our algorithm SCC for soft subspace clustering by minimizing Eq. (2) subject to Eq. (1), which is a constrained optimization problem. Using the Lagrangian multiplier technique, this can be transformed into an unconstrained optimization problem:

$$J(\Pi, W) = \sum_{k=1}^{K} \sum_{d=1}^{D} w_{kd} \times \text{DiS}_d(\mathbf{x}, \pi_k) + \sum_{k=1}^{K} \xi_k \left(1 - \sum_{d=1}^{D} w_{kd}\right)$$

with $\xi_k$ for $k = 1, 2, \ldots, K$ being the Lagrange multipliers corresponding to the constraints defined in Eq. (1). The usual method of achieving a local minimum of $J(\Pi, W)$ is to use partial optimization for each parameter. Following this method, minimization of $J(\Pi, W)$ can be performed by optimizing $\Pi$ and $W$ in a sequential structure analogous to the mathematics of the EM algorithm [25]. In each iteration, we first set $\Pi = \Pi$ and solve $W$ as $\Pi$ to minimize $J(\Pi, W)$. Next, $\Pi = \Pi$ is set and the optimal $W$, say $W$, is solved to minimize $J(\Pi, W)$.

The first problem can be solved by assigning each input $\mathbf{x}$ to its most similar cluster, by comparing the object-to-center distances of $\mathbf{x}$ to the $K$ clusters. Formally, we assign $\mathbf{x}$ to cluster $k$ according to

$$k' = \arg \min_{k} \sum_{d=1}^{D} \tilde{\omega}_{kd} \times \text{DiS}_d(\mathbf{x}, \pi_k).$$ \hspace{1cm} (10)

with

$$\tilde{\omega}_{kd} = \omega_{kd} \times \frac{|O_d| - 1}{|O_d|} + (1 - \lambda_k^2) \times \tilde{s}_{kd}^2$$ \hspace{1cm} (11)

for $k = 1, 2, \ldots, K$ and $d = 1, 2, \ldots, D$.

It can be seen that, by Eqs. (10) and (11), each categorical attribute is weighted according to the average dispersion of the categories on the attribute. In fact, the right side of Eq. (11) is precisely the smoothed dispersion of the samples in the cluster $\pi_k$ (recall that the sample dispersion $s_{kd}^2$ is equal to $(1/D)|O_d| - 1/|O_d|$ when the categories are uniformly distributed). This dispersion-based weighting scheme is consistent with the one used for numeric data clustering [5,10], where virtually all of the existing methods compute attribute weights according to the dispersion of the numeric values from the mean in the dimension of the cluster.

**Algorithm 1.** The outline of the SCC algorithm.

**Input:** $DB$, $K$, $\theta$ and a termination criterion which is a small positive number $\epsilon$;

**Output:** $\Pi$ and $W$;

**begin**

Let $t$ be the number of iterations, $t = 0$;

Set all the attribute weights of $W$ to $1/D$, and denote $W$ by $W^{(t)}$;

Set $\lambda_k = 0$ for $k = 1, 2, \ldots, K$;

Generate an initial partition $\Pi$ of $DB$;

**repeat**

1. Letting $W = W^{(t)}$, assign all the data objects according to the rule of Eq. (9) and denote the new $\Pi$ by $\Pi^{(t+1)}$;

2. Update the bandwidths $\lambda_k$ for $\forall k$ by $\hat{\lambda}_k$ defined in Eq. (8);

3. Update the weights using Eq. (10) and denote the new $W$ by $W^{(t+1)}$;

4. $t = t + 1$;

**until** $|J(\Pi^{(t)}, W^{(t)}) - J(\Pi^{(t-1)}, W^{(t-1)})| < \epsilon$

**Output** $\Pi^{(t)}$ and $W^{(t)}$. 

---

L. Chen et al. / Pattern Recognition 51 (2016) 322–332

325
The SCC algorithm, as outlined in Algorithm 1, performs subspace clustering on categorical data using the optimization methods presented above. Given DB and the number of clusters $K$, the algorithm starts clustering from an initial partition of the data set. To generate the initial $\Pi$, we first randomly choose $K$ data objects $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_K$ as the seeds; then, each object $\mathbf{x}$ is assigned to the nearest seed index $k_0$ according to the simple matching coefficient [8], i.e.,

$$k_0 = \arg \min_{\forall k} \frac{1}{D} \sum_{d=1}^{D} l(x_d \neq \mathbf{e}_{k_0}).$$

The resulting $K$ groups compose the initial $\Pi$. In terms of algorithmic structure, SCC can be viewed as a K-means-type algorithm [26]; therefore, the computational complexity is $O(KNDT)$, where $T$ denotes the number of iterations.

3.5. On the attribute weighting scheme

In SCC, the weight assigned to the $d$th attribute of $\mathbf{x}_k$ is

$$w_{kd} = \frac{1}{2} \left[ \left( \frac{1}{s^2} - 1 \right) / \|O_d \| + \left( 1 - \frac{1}{s^2} \right) \|l \| - \theta \right],$$

with the parameter $\theta \neq 0$ smoothing the weight distribution, according to Eqs. (10) and (11). When $\theta = 1$, the attribute along which the samples exhibit the smallest dispersion will be weighted one while the others will receive zero weights. The objective function is also minimized in this case; however, there is always only one attribute selected for each cluster. Thus, we set $\theta \neq 0$ and $\theta \neq 1$ in practice.

From the computational perspective, it is now generally accepted that the attribute weight should be computed as being inversely proportional to the dispersion of the samples from the cluster center in that attribute, as discussed in Section 2. The dispersion is usually estimated by the sample variance (for numeric attributes) or the Gini diversity index $s^2$ (for categorical attributes). In fact, most of the existing methods, including the numerous algorithms developed for numeric data clustering [5,6,10] and the recently published algorithms for categorical clustering [9,13], measure the importance of an attribute in this way. This is also the case for SCC when $\theta < 0$ or $\theta > 1$.

However, the weighting scheme of SCC with $0 < \theta < 1$ conflicts with the common consensus, which in this case dictates that the attribute weight will be proportional to the Gini diversity index $s^2$ of each attribute. We argue that the new scheme is also reasonable for categorical data clustering, because the Gini diversity index may yield a biased indication of the sample dispersion of categorical attributes in real applications. Let us take for example two attributes representing users’ ratings for two items in a recommendation system: $A_1 = \{5, 5, 5, 5, 1\}$ and $A_2 = \{3, 3, 3, 2, 2, 2\}$. The categories are generated using a five scoring system, with 5 being the highest score. Measuring the sample dispersions in terms of $s^2$, we obtain that $s_1^2 \approx 0.44$ and $s_2^2 = 0.5$. Based on the common consensus, $A_2$ is less important than $A_1$ since $s_2^2 > s_1^2$; however, it may not be the case as the opinions in $A_2$ are more similar to each other (note that the scores 2 and 3 are close in the scoring system) than those in $A_1$, which suggests that $A_2$ should be more important than $A_1$. In other words, the attribute weight should be proportional to the Gini diversity index in this case. More real examples will be given in the experiment section (Section 4).

The above analysis hints that both weighting schemes (say, weighting the attributes in favor of or against the sample dispersion measured by the Gini diversity index) are necessary for categorical data clustering. SCC offers the two schemes using different settings of the parameter $\theta$, i.e., $0 < \theta < 1$ and $\theta > 1$ or $\theta < 0$. Since $\theta$ is unique for all of the attributes, the desirable value of $\theta$ and accordingly the preferred weighting scheme are dependent on the overall distribution of the attributes. We will choose $\theta$ based on the mutual information of the category distribution with regard to the resulting clusters, as described in Section 4.

3.6. Distance-based cluster validation

To estimate the number of clusters in a categorical data set, in this subsection, we define a new cluster validity index based on the probabilistic distance to evaluate clustering quality for a partition-based algorithm, such as SCC. In the traditional trial-and-error method [28], the most appropriate number of clusters is determined by comparing the values of the index computed on different clustering results generated by the algorithm using various $K$ ranging in $[K_{\min}, K_{\max}]$.

The new index, denoted by $V_{cc}(K)$, is an extension to the traditional distance-based validity indices, such as [27] and [28], depending on a linear combination of the average within-cluster scatter $\text{Scat}(K)$ and between-cluster separation $\text{Sep}(K)$. The difficulties in applying such existing indices to categorical data clustering lie in the formulation of both $\text{Scat}(K)$ and $\text{Sep}(K)$, which are typically defined on the Euclidean distance for numeric clusters. To measure the “goodness” of $K$ categorical clusters, we reformulate the measures by replacing the Euclidean distance with the probabilistic distance, as discussed in Section 3.2. In the forms of those presented in [28], the two measures are reformulated as

$$\text{Scat}(K) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{D} \sum_{d=1}^{D} \sum_{i \neq k}^{K} \text{Dis}_{sd}(\mathbf{x}, \mathbf{x}_k)$$

and

$$\text{Sep}(K) = \max_{k < k'} \frac{\delta(\pi_k, \pi_{k'})}{\min_{k \neq k'} \delta(\pi_k, \pi_{k'})} \sum_{k=1}^{K} \left( \sum_{k \neq k'}^{K} \delta(\pi_k, \pi_{k'}) \right)^{-1}.$$

Here, $\text{Dis}_{sd}(\mathbf{x}, \mathbf{x}_k)$ is the object-to-cluster distance defined in Eq. (6) and $\delta(\pi_k, \pi_{k'})$ is the distance between clusters $\pi_k$ and $\pi_{k'}$ with $k, k' \in \{1, K\}$, measured by the Euclidean distance of the two probability distributions $p(X_d|\pi_k)$ and $p(X_d|\pi_{k'})$ on each attribute $d$, i.e.,

$$\delta(\pi_k, \pi_{k'}) = \sum_{d=1}^{D} \sum_{a \neq a'} \left| p(X_d = a|\pi_k) - p(X_d = a'|\pi_{k'}) \right|^2.$$

Based on the reformulations, the new cluster validity index is defined as follows:

$$V_{cc}(K) = \frac{\text{Scat}(K)}{\max_{K_{\min} \leq K \leq K_{\max}} \text{Scat}(k)} + \frac{\text{Sep}(K)}{\max_{K_{\min} \leq K \leq K_{\max}} \text{Sep}(k)}.$$

The two terms in Eq. (12) are normalized within-cluster scatter and between-cluster separation, respectively, in the sense of probabilistic distances for categorical clusters. The index offers therefore a trade-off between the two factors. The minimal $V_{cc}$ value is considered to be associated with the clustering results produced by the algorithm using the optimal number of clusters.

4. Experimental evaluation

In this section, we evaluate the performance of the proposed SCC on synthetic and real-world categorical data sets, and we also experimentally compare SCC with a few other mainstream clustering algorithms.

4.1. Experimental setup and evaluation measures

Five categorical data clustering algorithms, SCC, K-representatives (KR) [7], the weighted $K$-modes (WKM) [2], the mixed-attributes-weighting $K$-modes (MWKM) [3] and the recently published complement-entropy-weighting $K$-modes (CWKM) [13], were compared in our experiments. DWKM, MWKM and CWKM are mode-based algorithms, since they represent the clusters by
the mode categories. KR was chosen as a representative for non-mode clustering algorithms. Moreover, we will call WKM, MWKM, CWKM and KCC the subspace clustering algorithms, as all of them automatically weight the attributes during the clustering process. We also used LAC [10], a soft subspace clustering algorithm developed for numeric data, to provide a reference point on real-world data with numeric or ordinal attributes.

The subspace clustering algorithms (excluding CWKM) require a parameter for attribute weighting: \( \beta \) for WMK and MWKM, \( h \) for LAC and \( \theta \) for SCC. We set the parameter for them by running each algorithm on each data set with various values, and subsequently choosing the value that results in the highest category mutual information (MI). This is an index evaluating the clustering results for categorical data [29], defined by

\[
MI = \frac{1}{N} \sum_{k=1}^{K} \sum_{d=1}^{D} \sum_{a \in D} f_k(a_d) \times \frac{\log f_k(a_d)}{\log f_{WKM}(a_d)} \frac{m_k}{N}
\]

with \( f_k(a_d) \) being the frequency of \( a_d \) with regard to the entire data set. In the case where the clustering results contain a single cluster, i.e., \( K = 1 \), the index is computed as 0.

We calculated the clustering accuracy (CA) to evaluate the performance of a clustering:

\[
CA = \frac{1}{N} \sum_{k=1}^{K} a_k
\]

where \( a_k \) is the number of data objects in the majority class corresponding to \( x_k \). Clearly, such a measure requires that the ground truth of the data sets be known and \( K \) be set to the true number of the classes in the data, which is the case in our experiments. We also use the normalized mutual information (NMI) [5] for the evaluation, which is computed by

\[
NMI = \frac{\sum_{k=1}^{K} \sum_{j=1}^{J} m_{jk} \log \frac{N \cdot m_{jk}}{n_k m_j}}{\sqrt{\left( \sum_{k=1}^{K} n_k \log \frac{n_k}{N} \right) \left( \sum_{j=1}^{J} m_j \log \frac{m_j}{N} \right)}}
\]

where \( m_j \) is the size of the \( j \)-th class \( c_j \) and \( m_{jk} \) is the number of data objects occurring in both class \( c_j \) and cluster \( x_k \). The larger the CA or NMI, the better the clustering performance.

4.2. Experiment on real data

This set of experiments was designed to examine the performance of SCC in real-world applications. The experiments were conducted on real data with attributes of different types, i.e., nominal, ordinal and mixed nominal and ordinal types.

4.2.1. Real data sets

Four data sets obtained from the UCI Machine Learning Repository were used in the experiments: the Promoters Gene Sequences data (Promoters for short), the Splice-Junction Gene Sequences data (Splice for short), the Wisconsin Breast Cancer data (WBC for short) and the Australian Credit Card data (Credit for short). Both Promoters and Splice have only nominal attributes; WBC has only ordinal attributes while Credit contains attributes of mixed nominal and ordinal types.

The task of clustering Promoters is to recognize promoters in DNA sequences. The data set contains 106 samples in two classes: promoter and non-promoter. The samples feature 57 attributes, each associated with one of the four symbols A, G, T or C starting at position \(-50\) (the 5′ end of the sequence) and ending at position \(+7\) (the 3′ end). The task for Splice is to recognize the boundaries between exons and introns, which are classified into three classes: EI, IE and Neither (say, exon/intron boundaries, intron/exon boundaries and neither). Splice contains 3190 samples, each consisting of a window that covers 30 nucleotides before and after the possible junction location. The attributes are numbered according to the relative location, starting at position \(-30\) (the 5′ end) and ending at position \(+30\) (the 3′ end).

The third data set WBC contains 699 samples, which are classified into two classes: Benign and Malignant. The sample in WBC consists of nine ordinal attributes, each taking its values from the 10 integers \( 1 \ldots 10 \). It is a pure ordinal data set and can thus be directly clustered using a numeric data clustering algorithm like the LAC algorithm [10]. The fourth data set called Credit has been used in related work, such as [4]. It contains 653 samples, each with nine nominal and six numeric attributes. The samples are classified into two classes: rejected and approved. To enable the categorical data clustering algorithms on Credit, we transformed its six numeric attributes into ordinal attributes, using an equal-width discretization method. The number of bins was set to 10, which is the same as that of WBC. In applying the LAC algorithm [10] to Credit, all the nominal attributes were binarized, and finally we obtained 40 binary attributes.

4.2.2. Comparison of clustering quality

To estimate the number of clusters \( K \) for SCC, we first ran SCC with \( \theta = 0 \) (the weight assigned to each attribute is fixed at 1 in this case) and each integer \( K \) in the range \([2,10]\) for each data set. Then, we chose the value corresponding to the smallest \( V_{CC} \) for \( K \).

Fig. 1(a) reports the average \( V_{CC} \) values computed on 20 clustering results for each data set. The results indicate that the numbers of clusters should be 2 (for Promoters), 3 (for Splice), 2 (for WBC) and 2 (for Credit), respectively, which are precisely the true numbers of clusters in the four real data sets.

Fig. 1(b) shows the average \( MI \) obtained by SCC for each data set using the correct number of clusters and different \( \theta \) ranging from \(-0.95\) to \(0.95\) with increment 0.1 (such that 0 and 1 are excluded).

---

Fig. 1. (a) Change in the cluster validity index \( V_{CC} \) with various \( K \). (b) Change in the category mutual information with various \( \theta \) obtained by SCC on the real data sets.
Again, the average MI for each $\theta$ is calculated using 20 clustering results. The results of $\theta < -0.25$ for Splice and $\theta < -0.35$ for Credit are eliminated since SCC failed in clustering the data sets using the settings. We observe that SCC behaves similarly on Promoters and Splice, with the best clustering results achieved when $\theta = 0.05$ for Promoters and $\theta = -0.15$ for Splice) and the clustering quality obviously dropped with $\theta \in (0, 1)$. Note that in SCC with $\theta < 0$, the smaller the sample dispersion on the attribute, the larger the weight assigned to that attribute. It indicates that, for data sets containing only nominal attributes, such as Promoters and Splice used in the experiments, the attribute weights should be computed as being inversely proportional to the dispersion of the categories. This is consistent with the weighting scheme used in the existing methods [5, 6, 9, 10, 13].

However, different results are observed on both WBC and Credit. According to Fig. 1(b), SCC is able to achieve higher MI with $\theta \in (0, 1)$ than $\theta < 0$. In fact, the highest MI values occur when $\theta = 0.45$ for WBC and $\theta = 0.05$ for Credit, while the values drop significantly when $\theta < 0$ for Credit. This result is considerably different from that obtained on Promoters and Splice. This means that, for categorical data sets involving ordinal attributes, the interestingness of attributes could be computed as being proportional to their sample dispersion, which is opposite to the existing weighting scheme, as discussed in Section 3.5.

The weighting parameters in DWKM and MWKM were chosen in a similar way, and we obtained $\beta = 2$ for them, which is also the author-recommended value in both cases [2, 3]. For the parameter $h$ of LAC, we found that $h = 3.5$ and $h = 4$ yielded the highest MI on WBC and Credit, respectively. Using the above settings, each data set was clustered by each algorithm for 100 executions. The average performances are reported in Tables 1 and 2 in the format average $\pm$ 1 standard deviation, and the best results are marked in bold typeface. In Table 2 for WBC and Credit, the numeric data clustering algorithm LAC [10] is used as the baseline.

Table 1 summaries the average performances on Promoters and Splice, from which we can see that SCC is significantly more accurate than all of the competing algorithms in terms of both CA and NMI. All the mode-based algorithms perform poorly on these nominal data sets, especially on Splice which is a relatively large data set; on this data set, KR outperforms the mode-based algorithms considerably. According to the results in Table 2 corresponding to the ordinal data set WBC, it can be seen that only SCC outperforms LAC whereas the four competing algorithms underperform LAC. On the Credit data set with mixed-type attributes, SCC significantly outperforms LAC, while the other categorical clustering algorithms (excluding WKM) also outperform LAC in varying degrees.

4.2.3. Attribute-weighting results

To understand the reason for the performance shown in Tables 1 and 2, the attribute weights in the best clustering results were used for further analysis. Figs. 2 and 3 show the weight distributions for the classes of Promoters and Splice, respectively, generated by MWKM, CWKM, and our SCC. The results of WKM are omitted because the weighting approach used in WKM is similar to that in MWKM, where the weights are computed as being inversely proportion to the mode frequency. In the figures, the x-axis shows the DNA nucleotides, sorted by their positions, and the y-axis indicates the logarithmic weights.

In Figs. 2(c) and 3(c) corresponding to our SCC, there is no significant change in the weights over all positions for the non-promoter class of Promoters and the Neither class of Splice. These are expected outcomes because both classes do not contain interesting biological patterns (say, promoters in Promoters and intron/exon or exon/intron boundaries in Splice). On the other hand, for the promoter class shown in Fig. 2(c), we can see three slices of successive DNA nucleotides that are assigned obviously high weights compared to those for the non-promoter class. According to the domain theory [30], the three slices (the left side to right side of the figure) exactly correspond to the promoters named conformation@-45, contact-minus_35@-36 and contact-minus_10@-14, respectively. As Fig. 3(c) shows, for the EI and IE classes, one can also see two clear boundaries at the positions $p - 2 - p + 2$, which are precisely the positions of the “donor” and “acceptor” sites [31]. Compared with Figs. 2(a), (b) and 3(a), (b) generated by the competing algorithms, the weight distributions yielded by SCC clearly identify the most important attributes and consequently allow the generation of high-quality clusters.

Fig. 4 shows the attribute weights generated by MWKM, LAC and SCC on the WBC data set. Again, WKM is not used for comparison. CWKM is not chosen because the weight attribution yielded by the algorithm is close to that generated by LAC. From the figure, one can see the distinctive behaviors of different attribute-weighting approaches in identifying the importance of attributes. For example, SCC assigns the largest and the smallest weights to A13 and A9 of the Benign class, respectively, whereas MWKM and LAC yield approximately the opposite results. To examine the rationality of the weighting scheme used in SCC, we created two reduced data sets on the original WBC data by removing A1 and A9, respectively. Fig. 5 shows the change in clustering accuracy of the different algorithms on the data sets. As expected, all the mode-based algorithms achieve markedly higher clustering accuracy with A9 removed, and the accuracies of all the algorithms, including our SCC, drop to varying degrees when A1 is absent. The results indicate that the new weighting scheme of SCC, although not allowed by the existing methods (see Section 3.5), can yield a more relevant assessment than the built-in scheme for each of the methods.

Table 3 shows the attribute weights in the best clustering results generated by the different algorithms on some attributes of the Credit data set. WKM is eliminated since it failed in weighting the approved class. For SCC, A15 is an unimportant attribute, as it is assigned small weights for both classes. However, this is not the case in the other algorithms, as evidenced by the weighting results shown in the table. Note that on the same data set, [4] reported a similar result (i.e., A15 is a less important attribute) and showed that better clustering results can be obtained with both A14 and A15 removed. According to Table 3, the difference is that SCC suggests removing A13 and A15. So it is interesting to examine the change in clustering performance with the two different attribute subsets removed. Table 4 gives the results, where $A$ denotes the entire attribute set. KM and KR were eliminated because they do
not distinguish the different interestingness of attributes in clustering.

Table 4 shows that WKM, CWKM and SCC obtain more accurate results on the reduced attribute set $\mathcal{A} - \{A13,A15\}$ than on the set $\mathcal{A} - \{A14,A15\}$. LAC and MWKM retain their clustering accuracy for the two sets. Note that both A14 and A15 are numeric attributes, while A13 is a nominal attribute and was automatically identified by SCC during its clustering process on the entire data set. This indicates that SCC can make a favorable trade-off among the mixed-type attributes in the same data set and more precisely capture their individual importance to clusters, whence its good performance in soft subspace clustering of complex categorical data.

4.3. Experiments on synthetic data

This set of experiments was conducted on synthetic data sets. The use of synthetic data sets is motivated by the fact that their cluster structures can be controlled, which facilitates performance evaluation.

4.3.1. Synthetic data sets

The synthetic data sets were generated using a method similar to that suggested by Aggarwal et al. [15], with the parameters $N$ (the number of data objects), $D$ (the number of attributes), $K$ (the number of clusters), $l$ (the percentage of relevant attributes per cluster) and $|\mathcal{O}|$ (the number of categories in each attribute) characterizing the data sets and controlling the data generation process.

Each synthetic data set was generated in two steps. In the first step, we generated a primary data set consisting of numeric attributes; then, the values on each numeric attribute were discretized into categories to create categorical attributes in the second step. During the generation process, the attributes for different clusters were created to share some correlated subsets. On an irrelevant attribute the coordinates of the objects were uniformly distributed, while on a relevant attribute they were generated according to a normal distribution with the mean randomly chosen from $[0,100]$ and the variance determined by $4q^2$, where $q$ was chosen uniformly at random in the range $[1,2]$. All the coordinates were generated in the range $[0,100]$. In the second step, we created categorical attributes by discretizing each generated attribute using an equal-width binning method, where the number of bins was set to the given $|\mathcal{O}|$.

Three groups, each containing five data sets, were generated to evaluate the performance of SCC with respect to different number of objects, categories on each attribute, and attributes themselves. To examine the relationship of clustering performance to the number of categories, we generated the first group with $N=6000$, $D=80$, $l=20\%$, $K=4$ and $|\mathcal{O}| \in \{4,8,16,24,32\}$. For each data set in the second group, we set $N=6000$, $l=20\%$, $K=4$, $|\mathcal{O}| = 4$ and $D \in \{40,80,120,160,200\}$ to evaluate the performance with increasing number of attributes. The third group was generated with $D=80,$
Clustering accuracy of the algorithms on the WBC data set with various attribute sets.

The accuracy of the different algorithms with increasing numbers of objects, since the cluster structure remains unchanged while the accuracy is slightly affected on the data set with large number of data objects, tends to drop on the high-dimensional data as it lacks an adaptive scheme to distinguish different interestingness of attributes to the clusters.

4.3.2. Evaluation results

Fig. 6(a) summarizes the average clustering accuracies achieved by the algorithms on the five data sets in the first group. Each data set was clustered by each algorithm for 20 executions and the average clustering accuracies are reported. It can be seen that SCC performs the best clustering results, especially on the high-dimensional data sets. MWKM and CWKM achieve higher accuracy when the number of attributes increases, due to the small number of categories and the combination of automated variable selection methods; however, they both underperform SCC by more than 20%. WKM performs unstably on these data sets because, as pointed out by [3], the weighting scheme used in WKM might inversely reduce the dissimilarity of samples on an important attribute. The accuracy of KR tends to drop on the high-dimensional data as it lacks an adaptive scheme to distinguish different interestingness of attributes to the clusters.

4.3.3. Scalability

In this subsection, we test the scalability of SCC with respect to the numbers of attributes and the data size. Fig. 7 shows the average times used by SCC, as well as its average clustering accuracy, on the data sets in the second group and the third group.

The relationships between the runtime (in seconds) and the number of the attributes are illustrated in Fig. 7(a), from which we can see that the runtime of SCC increases linearly with respect to the number of attributes, accompanied by high clustering accuracy. The sensitivity with respect to the number of data objects was tested on synthetic data sets containing data objects in varying sizes up to 48,000. The results are shown in Fig. 7(b), where the y-axis indicates the logarithmic seconds for ease of drawing. From the figure, it can be seen that SCC scales linearly with the data sizes. The clustering accuracy is slightly affected on the data set with large number of data objects, since the cluster structure remains unchanged while the number of data objects is increasing. In this case, the bandwidths estimated by SCC become small (see Eq. (8)). Consequently, the clustering accuracy is affected.

5. Conclusion and perspectives

In this paper, we began by discussing the difficulty soft subspace clustering algorithms encounter in clustering categorical data, due to the fact that general statistical measures such as mean and variance, which are common in numeric data, are undefined for categorical data. We derived a probabilistic distance function for measuring the distance of categorical objects to clusters (namely, the object-to-cluster distance), using a kernel density estimation method. Based on the probabilistic distances, we proposed a partition-based clustering algorithm, named SCC, using an automated variable selection method which identifies the individual importance of attributes to clusters according to the kernel-smoothed sample dispersion. In particular, we showed that the importance of a categorical attribute might not necessarily be measured by the same approach that has
been successfully applied for numerical data clustering, especially when the data set contains ordinal attributes. We also extended the traditional cluster validity index originally designed for numeric data to categorical data, and proposed a new index called \( V_{CC} \) to evaluate the clustering quality of categorical clusters. Experiments were conducted on synthetic data sets and four complex real-world data sets, and the results show the outstanding effectiveness of the proposed methods compared with state-of-the-art methods.

There are many directions that are clearly of interest for future exploration. One avenue of further study is to assign each cluster an individual weighting exponent and estimate the parameters adaptively. Our further efforts will also be directed toward extending the method to the general kernel functions and testing the method on various kernels.

Conflict of interest

None declared.

Acknowledgments

The work has been partly funded by the Discovery Accelerator Supplements program of Natural Sciences and Engineering Research Council of Canada (No. 396097-2010) granted to Prof. Shengrui Wang. L. Chen's work was also supported by the National Natural Science Foundation of China (No. 61175123), and the Fujian Normal University Innovative Research Team (IRT1207). J. Zhu's work was supported by the National Social Science Foundation of China (Major Program 13&ZD148).

References


Table 4

<table>
<thead>
<tr>
<th>Attribute set</th>
<th>LAC</th>
<th>WKM</th>
<th>MWKM</th>
<th>CWKM</th>
<th>SCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{A} \rightarrow {A14, A15} )</td>
<td>0.68 ± 0.13</td>
<td>0.69 ± 0.09</td>
<td>0.73 ± 0.10</td>
<td>0.72 ± 0.11</td>
<td>0.83 ± 0.02</td>
</tr>
<tr>
<td>( \mathcal{A} \rightarrow {A13, A15} )</td>
<td>0.68 ± 0.13</td>
<td>0.65 ± 0.10†</td>
<td>0.73 ± 0.11</td>
<td>0.73 ± 0.11†</td>
<td>0.84 ± 0.01†</td>
</tr>
</tbody>
</table>

† better, in comparison with the results obtained on \( \mathcal{A} \rightarrow \{A14, A15\} \), using the paired \( t \)-test with significance level 0.05.


