Pattern Recognition 46 (2013) 2228–2238

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number

Yiu-ming Cheung [a,b,*], Hong Jia [a]

[a] Department of Computer Science and Institute of Computational and Theoretical Studies, Hong Kong Baptist University, Hong Kong, China
[b] United International College, Beijing Normal University-Hong Kong Baptist University, Zhuhai, China

ARTICLE INFO

Article history:
Received 19 March 2012
Received in revised form
29 December 2012
Accepted 23 January 2013
Available online 31 January 2013

Keywords:
Clustering
Similarity metric
Categorical attribute
Numerical attribute
Number of clusters

ABSTRACT

Most of the existing clustering approaches are applicable to purely numerical or categorical data only, but not the both. In general, it is a nontrivial task to perform clustering on mixed data composed of numerical and categorical attributes because there exists an awkward gap between the similarity metrics for categorical and numerical data. This paper therefore presents a general clustering framework based on the concept of object-cluster similarity and gives a unified similarity metric which can be simply applied to the data with categorical, numerical, and mixed attributes. Accordingly, an iterative clustering algorithm is developed, whose outstanding performance is experimentally demonstrated on different benchmark data sets. Moreover, to circumvent the difficult selection problem of cluster number, we further develop a penalized competitive learning algorithm within the proposed clustering framework. The embedded competition and penalization mechanisms enable this improved algorithm to determine the number of clusters automatically by gradually eliminating the redundant clusters. The experimental results show the efficacy of the proposed approach.

1. Introduction

To discover the natural group structure of objects represented in numerical or categorical attributes [1], clustering analysis has been widely applied to a variety of scientific areas such as computer science [2] and bioinformatics [3]. Traditionally, clustering analysis concentrates on purely numerical data only. The typical clustering algorithms include the *k*-means [4], EM algorithm [5] and their variants. Since the objective functions of these two algorithms are both numerically defined, they are not essentially applicable to the data sets with categorical attributes. Under the circumstances, a straightforward way to overcome this problem is to transform the categorical values into numerical ones, e.g. the binary strings, and then apply the aforementioned numerical-value based clustering methods. Nevertheless, such a method has ignored the similarity information embedded in the categorical values and cannot faithfully reveal the similarity structure of the data sets [6]. Hence, it is desirable to solve this problem by finding a unified similarity metric for categorical and numerical attributes such that the metric gap between numerical and categorical data can be eliminated. Subsequently, a general clustering algorithm which is applicable to numerical and categorical data can be presented based on this unified metric. During the past decades, some works which try to find a unified similarity metric for categorical and numerical attributes have been presented, e.g. see [7]. However, a computational efficient similarity measure remains to be developed.

Another challenging problem encountered in clustering is how to determine the number of clusters. To the best of our knowledge, a lot of popular clustering methods, e.g. the *k*-means algorithm for numerical data clustering and the *k*-modes algorithm [8] for categorical data clustering, need to pre-assign the number of clusters exactly. Otherwise, they will almost always lead to a poor clustering result [9,10]. Unfortunately, in many cases, this vital information is not always available from the practical viewpoint. Hence, to explore an algorithm which can conduct clustering without knowing cluster number is also a significant work in clustering analysis. To address this issue, variant researches have been conducted in the literature and some feasible methods that can determine the number of clusters for purely numerical or categorical data have been presented [9–11]. Nevertheless, to the best of our knowledge, how to automatically select cluster number for mixed data during clustering process is still an unsolved problem.

In this paper, we will propose a unified clustering approach that is capable of selecting the cluster number automatically for both categorical and numeric data sets. Firstly, we present a general clustering framework based on the concept of object-

* Corresponding author. Tel.: +852 34115155.
E-mail addresses: ymc@comp.hkbu.edu.hk (Y.-m. Cheung), hjia@comp.hkbu.edu.hk (H. Jia).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.01.027

cluster similarity. Then, a new metric for both of numerical and categorical attributes is proposed. Under this metric, the object-cluster similarity for either categorical or numerical attributes has a uniform criterion. Hence, transformation and parameter adjustment between categorical and numerical values in data clustering are circumvented. Subsequently, an iterative clustering algorithm is introduced. This algorithm conducts a parameter-free clustering analysis and is applicable to the three types of data: numerical, categorical, or mixed data, i.e., the data with the both of numerical and categorical attributes. Moreover, empirical studies show that the proposed algorithm has higher accuracy as well as lower computational cost compared to the popular k-modes algorithm for categorical data clustering. For mixed data clustering, compared to k-prototype algorithm [12], the proposed method can get much better clustering results, but no parameter needs to be adjusted at all. Additionally, to overcome the cluster number selection problem, we further present a penalized competitive learning algorithm within the proposed clustering framework. The competition and penalization mechanisms in this improved algorithm can gradually fade out the redundant clusters. Hence, the number of clusters can be determined automatically during the clustering process. Experimental results on benchmark data sets have shown the effectiveness of this method.

The rest of this paper is organized as follows. Related works are reviewed in Section 2. Section 3 proposes a general clustering framework based on object-cluster similarity, whose metric is also defined. Section 4 describes an iterative clustering algorithm and Section 5 presents an improved one with capability of automatically selecting cluster number. Experiments are conducted in Section 6. Finally, we draw a conclusion in Section 7.

2. Related works

This section reviews the related works on: (1) data clustering with categorical-and-numerical attributes and (2) cluster number selection.

In the former, several methods have been presented which can be grouped into two lines. In the first line, the algorithms are essentially designed for purely categorical data, although they have been applied to the mixed data as well by transforming the numerical attributes to categorical ones via a discretization method. Along this line, several methods have been proposed based on the perspective of similarity metric, graph partitioning or information entropy. For example, ROCK algorithm proposed by Guha et al. [13] is an agglomerative hierarchical clustering procedure based on the concepts of neighbors and links. In this method, a pair of objects are regarded as neighbors if their similarity exceeds a certain threshold, and the desired cluster structure is obtained by merging the clusters sharing a pre-assigned number of neighbors gradually. ROCK has shown its superiority over traditional hierarchical algorithms in the experiments, but its performance is generally sensitive to the setting of similarity threshold. Also, the computation of links between objects is quite time-consuming [14]. By contrast, CLICKS algorithm proposed in [15] mines subspace clusters for categorical data sets. This method encodes a data set into a weighted graph structure, where each weighted vertex stands for an attribute value and two nodes are connected if there is a sample in which the corresponding attribute values co-occur. Experiments have shown that CLICKS outperforms ROCK algorithm and scales better for high-dimensional data sets. However, its performance also depends upon a set of parameters whose tuning is quite difficult from the practical viewpoint. Additionally, the COOLCAT algorithm, an entropy-based method proposed by Barbara et al. [16], utilizes the information entropy to measure the closeness between objects and presents a scheme to find a clustering structure via minimizing the expected entropy of clusters. The performance of this algorithm is stable for different data sizes and parameter settings. Furthermore, a scalable algorithm for categorical data clustering called LIMBO [17], which is proposed based on the Information Bottleneck (IB) framework [18], employs the concept of mutual information to find a clustering with minimum information loss. In general, all of the above-stated algorithms can be applied to mixed data via a discretization process, which may, however, cause loss of important information, e.g. the difference between numerical values.

By contrast, the second line attempts to design a generalized clustering criterion for numerical-and-categorical attributes. For example, Li and Biswas [7] presented the Similarity Based Agglomerative Clustering (SBAC) algorithm which is based on Goodall similarity metric [19] that assigns a greater weight to uncommon feature value matching in similarity computations without the prior knowledge of the underlying distributions of the feature values. This method has a good capability of dealing with the mixed attributes, but its computation is quite laborious. He et al. [20] extended the Squeezer algorithm to cluster mixed data and proposed the usm-squeezer method, in which the similarity measure for categorical attributes is the same as the Squeezer while the similarity of numerical attributes is defined by relative difference. However, the clustering effectiveness of this method has not been sufficiently demonstrated. In [21], an Evidence-Based Spectral Clustering (EBSC) algorithm has been proposed for mixed data clustering by integrating the evidence based similarity metric into the spectral clustering structure. Moreover, the AUTOCLASS proposed by Cheeseman and Stutz [22] assumes a classical finite mixture distribution model on mixed data and utilizes a Bayesian method to derive the most probable class distribution for the data given prior information. Among this category of approaches, the most cost-effective one may be the k-prototype algorithm proposed by Huang [12]. In this method, the distance between two categorical values is defined as 0 if they are the same, and 1 otherwise while the distance between numerical values are quantified with Euclidean distance. Subsequently, the k-means paradigm is utilized for clustering. However, since different metrics are adopted for numerical and categorical attributes, a user-defined parameter is utilized to control the proportions of numerical distance and categorical distance. Nevertheless, the clustering result is very sensitive to the setting of this parameter. A simplified version of k-prototype algorithm namely k-modes [8,23,24], which is applicable for purely categorical data clustering, has also been widely utilized. Thus far, different improvement strategies on this method have been explored, e.g. see [25–27].

In general, all of the aforementioned methods need to pre-assign the number of clusters exactly, which is, however, a nontrivial task from the practical viewpoint. In the literature, a variety of methods have been proposed for cluster number estimation. For example, some computational demanding methods choose the optimal number of clusters via different statistic criteria, such as Akaike's Information Criterion (AIC) [28] and Schwarz's Bayesian inference criterion (BIC) [29]. By contrast, another kind of methods within the framework of competitive learning often introduce some competitive mechanisms, such as penalization [9,11] and cooperation [30], into the clustering process so that the number of clusters can be automatically selected. Nevertheless, these existing methods focus on numerical data only and cannot be directly applied to data sets with categorical attributes. Recently, Liao and Ng [10] have introduced an entropy penalty term into the objective function of k-modes algorithm. Then, by choosing different values for the regularization parameter, variant clustering results with different cluster

numbers can be obtained. Subsequently, the cluster number accompanying with the most stable clustering result is selected. As the learning process needs to be repeated for a large range of values of regulation parameter, the computation of this method is much more expensive than the original k-modes algorithm.

3. Clustering problem and object-cluster similarity metric

The general task of clustering is to classify the given objects into several clusters such that the similarities between objects in the same group are high while the similarities between objects in different groups are low [31,32]. Therefore, clustering a set of N objects (also called *inputs* interchangeably), $\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N\}$, into k different clusters, denoted as $C_1,C_2,\ldots,C_k$, can be formulated to find the optimal $\mathbf{Q}^*$ via the following objective function:

$$\mathbf{Q}^* = \arg\max_{\mathbf{Q}} F(\mathbf{Q}) = \arg\max_{\mathbf{Q}} \left[\sum_{j=1}^{k}\sum_{i=1}^{N} q_{ij}s(\mathbf{x}_i,C_j)\right], \tag{1}$$

where $s(\mathbf{x}_i,C_j)$ is the similarity between object $\mathbf{x}_i$ and Cluster C_j, and $\mathbf{Q}=(q_{ij})$ is an $N\times k$ partition matrix satisfying

$$\sum_{j=1}^{k} q_{ij} = 1, \quad \text{and} \quad 0 < \sum_{i=1}^{N} q_{ij} < N, \tag{2}$$

with

$$q_{ij}\in[0,1], \quad i=1,2,\ldots,N,\ j=1,2,\ldots,k. \tag{3}$$

Evidently, the desired clusters can be obtained by Eq. (1) as long as the metric of object-cluster similarity is determined. In the following sub-sections, we shall therefore study the similarity metric.

3.1. Similarity metric for mixed data

This sub-section will study the object-cluster similarity metric for mixed data. Suppose the mixed data $\mathbf{x}_i$ with d different attributes consists of d_c categorical attributes and d_u numerical attributes, i.e., d_c+d_u+d. Hence, x_i can be denoted as $[\mathbf{x}_i^{cT},\mathbf{x}_i^{uT}]^T$ with $\mathbf{x}_i^c=(x_{i1}^c,x_{i2}^c,\ldots,x_{id_c}^c)^T$ and $\mathbf{x}_i^u=(x_{i1}^u,x_{i2}^u,\ldots,x_{id_u}^u)^T$. Then, x_{ir}^u $(r=1,2,\ldots,d_u)$ belongs to $\mathbf{R}$ and x_{ir}^c $(r=1,2,\ldots,d_c)$ belongs to $dom(A_r)$, where $\{A_1,A_2,\ldots,A_{d_c}\}$ are the d_c categorical attributes and $dom(A_r)$ contains all the possible values that can be chosen by attribute A_r. For categorical attributes, as the value domains are finite and unordered, $dom(A_r)$ with m_r elements can be represented with $dom(A_r)=\{a_{r1},a_{r2},\ldots,a_{rm_r}\}$.

Firstly, we focus on the difference between categorical attributes and numerical attributes. For categorical attributes, each attribute can usually represent an important feature of the given object. Therefore, when we conduct classification or clustering analysis, we often investigate the categorical attributes one by one such as Decision Tree method. By contrast, the numerical attributes are often treated as a vector and handled together in clustering analysis. That is, we pay more attention to the total effect of numerical attributes. Based on these observations, for the mixed data $\mathbf{x}_i$, the numerical part $\mathbf{x}_i^u$ can be treated as a whole but the d_c categorical attributes should be investigated individually. Consequently, although the dimensionality of $\mathbf{x}_i$ is d, the number of features that contributes to clustering analysis will be d_c+1 (i.e., d_c categorical features and 1 numerical vector). Let the object-cluster similarity between $\mathbf{x}_i$ and cluster C_j, denoted as $s(\mathbf{x}_i,C_j)$, be the average of the similarity calculated based on each feature, we will then have

$$s(\mathbf{x}_i,C_j) = \frac{1}{d_f}[s(x_{i1}^c,C_j)+s(x_{i2}^c,C_j)+\cdots+s(x_{id_c}^c,C_j)+s(\mathbf{x}_i^u,C_j)]$$

$$= \frac{1}{d_f}\sum_{r=1}^{d_c} s(x_{ir}^c,C_j)+\frac{1}{d_f}s(\mathbf{x}_i^u,C_j), \tag{4}$$

where $d_f=d_c+1$. If we denote the similarity between $\mathbf{x}_i^c$ and C_j as $s(\mathbf{x}_i^c,C_j)$, we can get

$$s(\mathbf{x}_i^c,C_j) = \frac{1}{d_c}\sum_{r=1}^{d_c} s(x_{ir}^c,C_j) = \sum_{r=1}^{d_c}\frac{1}{d_c}s(x_{ir}^c,C_j). \tag{5}$$

Then, Eq. (4) can be further rewritten as

$$s(\mathbf{x}_i,C_j) = \frac{d_c}{d_f}\sum_{r=1}^{d_c}\frac{1}{d_c}s(x_{ir}^c,C_j)+\frac{1}{d_f}s(\mathbf{x}_i^u,C_j) = \frac{d_c}{d_f}s(\mathbf{x}_i^c,C_j)+\frac{1}{d_f}s(\mathbf{x}_i^u,C_j), \tag{6}$$

where $s(\mathbf{x}_i^c,C_j)$ is the similarity on categorical attributes and $s(\mathbf{x}_i^u,C_j)$ is the similarity on numerical attributes. Subsequently, the object-cluster similarity metric can be obtained based on the definitions of $s(\mathbf{x}_i^c,C_j)$ and $s(\mathbf{x}_i^u,C_j)$.

3.1.1. Similarity metric for categorical attributes

In Eq. (5) we have assumed that each categorical attribute has the same contribution to the calculation of similarity on categorical part. But in practice, due to the different distributions of attribute values, categorical attributes each often have unequal importance for clustering analysis. In light of this characteristic, Eq. (5) should be further modified with

$$s(\mathbf{x}_i^c,C_j) = \sum_{r=1}^{d_c} w_r s(x_{ir}^c,C_j), \tag{7}$$

where w_r is the weight of categorical attribute A_r satisfying $0\le w_r\le 1$ and $\sum_{r=1}^{d_c} w_r = 1$. That is, the object-cluster similarity for categorical part is the weighted summation of the similarity between the cluster and each attribute value. The weight factor w_r describes the importance of each categorical attribute and is utilized to control the contribution of attribute-cluster similarity to object-cluster similarity.

Definition 1. The similarity between a categorical attribute value x_{ir}^c and Cluster C_j, $i\in\{1,2,\ldots,N\}$, $r\in\{1,2,\ldots,d_c\}$, $j\in\{1,2,\ldots,k\}$, is defined as

$$s(x_{ir}^c,C_j) = \frac{\sigma_{A_r=x_{ir}^c}(C_j)}{\sigma_{A_r\neq NULL}(C_j)}, \tag{8}$$

where $\sigma_{A_r=x_{ir}^c}(C_j)$ counts the number of objects (also called *instances* hereinafter) in Cluster C_j that have the value x_{ir}^c for attribute A_r, NULL refers to the empty, and $\sigma_{A_r\neq NULL}(C_j)$ means the number of objects in Cluster C_j that have the attribute A_r whose value is not equal to NULL.

From Definition 1, we can find that this metric of attribute-cluster similarity is defined from probabilistic viewpoint and has the following properties:

(1) $0\le s(x_{ir}^c,C_j)\le 1$;
(2) $s(x_{ir}^c,C_j)=1$ only if all the instances belonging to Cluster C_j have the value x_{ir}^c for attribute A_r, and $s(x_{ir}^c,C_j)=0$ only if no instance belonging to Cluster C_j has the value x_{ir}^c for attribute A_r.

According to Eqs. (7) and (8), the object-cluster similarity for categorical part can be therefore calculated by

$$s(\mathbf{x}_i^c,C_j) = \sum_{r=1}^{d_c} w_r s(x_{ir}^c,C_j) = \sum_{r=1}^{d_c} w_r \frac{\sigma_{A_r=x_{ir}^c}(C_j)}{\sigma_{A_r\neq NULL}(C_j)}, \tag{9}$$

where $i\in\{1,2,\ldots,N\}$ and $j\in\{1,2,\ldots,k\}$.

Remark 1. Since $0 \le s(x_{ir}^c, C_j) \le 1$ and $\sum_{r=1}^{d_c} w_r = 1$, we have

$$s(\mathbf{x}_i^c, C_j) = \sum_{r=1}^{d_c} w_r s(x_{ir}^c, C_j) \ge \sum_{r=1}^{d_c} (w_r \cdot 0) = 0,$$

and

$$s(\mathbf{x}_i^c, C_j) = \sum_{r=1}^{d_c} w_r s(x_{ir}^c, C_j) \le \sum_{r=1}^{d_c} (w_r \cdot 1) = \sum_{r=1}^{d_c} w_r = 1.$$

That is, for any $i \in \{1,2,\ldots,N\}$ and $j \in \{1,2,\ldots,k\}$, the value of $s(\mathbf{x}_i^c, C_j)$ will fall into the interval [0,1].

Next, we discuss how to estimate the importance of each categorical attribute. From the view point of information theory, the significance of an attribute can be regarded as the inhomogeneity degree of the data set with respect to this attribute. Furthermore, according to the Measure III proposed in [33], if the information content of an attribute is high, the inhomogeneity of the data set is also high for this attribute. Hence, the importance of an arbitrary attribute A can be quantified by the following entropy metric:

$$H_A = -\int p(x(A)) \log\left(p(x(A))\right) dx(A), \tag{10}$$

where $x(A)$ is the value of attribute A, and $p(x(A))$ is the probability density function of $x(A)$ along this dimension. For categorical attributes, since the possible attribute values are finite, discrete and independent, the information content of an attribute can be estimated by the average information content of all possible attribute values and the probability of each attribute value can be computed by counting its frequency in the whole data set. Consequently, the importance of any categorical attribute A_r ($r \in \{1,2,\ldots,d_c\}$) can be calculated by

$$H_{A_r} = -\sum_{t=1}^{m_r} p(a_{rt}) \log p(a_{rt}), \tag{11}$$

with

$$p(a_{rt}) = \frac{\sigma_{A_r = a_{rt}}(X)}{\sigma_{A_r \ne NULL}(X)}, \tag{12}$$

where $a_{rt} \in dom(A_r)$, m_r is the total number of values that can be chosen by A_r and X is the whole data set. Furthermore, according to Eq. (11), the more different values an attribute has, the higher its significance is. However, in practice, an attribute with too many different values may have little contribution to clustering. For example, the ID number of instances is unique for each instance, but this information is useless for clustering analysis. Hence, Eq. (11) can be further modified with

$$H_{A_r} = -\frac{1}{m_r}\sum_{t=1}^{m_r} p(a_{rt}) \log p(a_{rt}). \tag{13}$$

That is, the importance of an attribute is quantified by its average entropy over each attribute value. The weight of each attribute is then computed as

$$w_r = \frac{H_{A_r}}{\sum_{t=1}^{d_c} H_{A_t}}, \quad r = 1,2,\ldots,d_c. \tag{14}$$

Subsequently, the object-cluster similarity on categorical part can be given by

$$s(\mathbf{x}_i^c, C_j) = \sum_{r=1}^{d_c} \left(\frac{H_{A_r}}{\sum_{t=1}^{d_c} H_{A_t}} \cdot \frac{\sigma_{A_r = x_{ir}^c}(C_j)}{\sigma_{A_r \ne NULL}(C_j)} \right). \tag{15}$$

In practice, for an attribute A_r, if all the instances to be classified have the same value a, it can be obtained from Eqs. (12) and (13) that the importance of this attribute will be zero as $p(a) = 1$. Then, the corresponding attribute weight will also be zero. This implies that this attribute will have no contribution at all to the whole clustering learning.

3.1.2. Similarity metric for numerical attributes

Since the distance between each vector $\mathbf{x}_i^u$ can be numerically calculated, the similarity metric for numerical attributes can be defined based on the measure of distance.

Definition 2. The object-cluster similarity between numerical vector $\mathbf{x}_i^u$ and cluster C_j, $i \in \{1,2,\ldots,N\}$, $j \in \{1,2,\ldots,k\}$, is given by

$$s(\mathbf{x}_i^u, C_j) = \frac{\exp(-0.5 Dis(\mathbf{x}_i^u, \mathbf{c}_j))}{\sum_{t=1}^{k} \exp(-0.5 Dis(\mathbf{x}_i^u, \mathbf{c}_t))}, \tag{16}$$

where $\mathbf{c}_j$ is the center of all numerical vectors in cluster C_j and $Dis(\cdot)$ stands for a distance function. It can be seen that the values of this similarity metric also fall into the interval [0, 1].

In practice, if the Mahalanobis distance metric is adopted, we will have

$$Dis(\mathbf{x}_i^u, \mathbf{c}_j) = (\mathbf{x}_i^u - \mathbf{c}_j)^T \Sigma_j^{-1} (\mathbf{x}_i^u - \mathbf{c}_j), \tag{17}$$

where Σ_j is the covariance matrix of numerical vectors in jth cluster. Further, if we utilize the Euclidean distance, the similarity metric can become

$$s(\mathbf{x}_i^u, C_j) = \frac{\exp(-0.5\|\mathbf{x}_i^u - \mathbf{c}_j\|^2)}{\sum_{t=1}^{k} \exp(-0.5\|\mathbf{x}_i^u - \mathbf{c}_t\|^2)}. \tag{18}$$

Actually, it can be derived that this similarity metric is equivalent to the posterior probability of $\mathbf{x}_i^u$ belonging to cluster C_j provided that the probability density function of each vector is a mixture of standard normal distribution with equal mixture coefficients.

3.2. Object-cluster similarity metric

According to Eqs. (6), (15) and (16), the object-cluster similarity metric for mixed data is defined as

$$s(\mathbf{x}_i, C_j) = \frac{d_c}{d_f} s(\mathbf{x}_i^c, C_j) + \frac{1}{d_f} s(\mathbf{x}_i^u, C_j) = \frac{d_c}{d_f} \sum_{r=1}^{d_c} \left(\frac{H_{A_r}}{\sum_{t=1}^{d_c} H_{A_t}} \cdot \frac{\sigma_{A_r = x_{ir}^c}(C_j)}{\sigma_{A_r \ne NULL}(C_j)} \right) + \frac{1}{d_f} \frac{\exp(-0.5 Dis(\mathbf{x}_i^u, \mathbf{c}_j))}{\sum_{t=1}^{k} \exp(-0.5 Dis(\mathbf{x}_i^u, \mathbf{c}_t))}, \tag{19}$$

where $i = 1,2,\ldots,N$, $j = 1,2,\ldots,k$. It can be seen that the defined similarities for categorical and numerical attributes in Eq. (19) are in the same scale. That is, the values for $s(\mathbf{x}_i^c, C_j)$ and $s(\mathbf{x}_i^u, C_j)$ are within the interval [0,1]. Hence, unlike k-prototype method, additional parameters to control the proportions of numerical and categorical distances are not needed any more.

Specially, if the data to be classified contain categorical attributes only, there does not exist the numerical vector $\mathbf{x}_i^u$ of each object $\mathbf{x}_i$. Then, we can get $\mathbf{x}_i = \mathbf{x}_i^c$, $d_c = d$ and $d_f = d$. Therefore, for purely categorical data, the object-cluster similarity is calculated with

$$s(\mathbf{x}_i, C_j) = \sum_{r=1}^{d} \left(\frac{H_{A_r}}{\sum_{t=1}^{d} H_{A_t}} \cdot \frac{\sigma_{A_r = x_{ir}}(C_j)}{\sigma_{A_r \ne NULL}(C_j)} \right). \tag{20}$$

By contrast, when clustering analysis is conducted on purely numerical data, with $d_c = 0$ and $d_f = 1$, the defined object-cluster similarity metric will degenerate to

$$s(\mathbf{x}_i, C_j) = \frac{\exp(-0.5 Dis(\mathbf{x}_i, \mathbf{c}_j))}{\sum_{t=1}^{k} \exp(-0.5 Dis(\mathbf{x}_i, \mathbf{c}_t))}. \tag{21}$$

4. Iterative clustering algorithm

In this section, we will present an iterative clustering algorithm based on the proposed object-cluster similarity metric to conduct clustering analysis.

This paper concentrates on hard partition only, i.e., $q_{ij} \in \{0,1\}$, although it can be easily extended to the soft partition in terms of posterior probability. Under the circumstances, given a set of N objects, the optimal $\mathbf{Q}^* = \{q_{ij}^*\}$ in Eq. (1) can be given by

$$q_{ij}^* = \begin{cases} 1 & \text{if } s(\mathbf{x}_i, C_j) \geq s(\mathbf{x}_i, C_r) \quad \forall 1 \leq r \leq k, \\ 0 & \text{otherwise,} \end{cases} \tag{22}$$

where $i = 1,2,\ldots,N$ and $j = 1,2,\ldots,k$. That is, each object $\mathbf{x}_i$ will be assigned to the cluster that has the largest object-cluster similarity with it among the k clusters. Therefore, an iterative algorithm can be conducted as Algorithm 1 to implement the clustering analysis.

Algorithm 1. Iterative clustering learning based on object-cluster similarity metric (OCIL).

Input: data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, number of clusters k
Output: cluster label $Y = \{y_1, y_2, \ldots, y_N\}$
1. Calculate the importance of each categorical attribute according to Eq. (11), if applicable
2. Set $Y = \{0,0,\ldots,0\}$ and select k initial objects, one for each cluster
repeat Initialize *noChange*=*true*.
for $i=1$ **to** N **do**
3. $y_i^{(new)} = \text{argmax}_{j \in \{1,\ldots,k\}}[s(\mathbf{x}_i, C_j)]$
if $y_i^{(new)} \neq y_i^{(old)}$ **then**
noChange=*false*
4. Update the information of clusters $C_{y_i^{(new)}}$ and $C_{y_i^{(old)}}$, including the frequency of each categorical value and the centroid of numerical vectors.
end if
end for
until *noChange* is *true*

In step 3 of Algorithm 1 (also called *OCIL algorithm* hereinafter), the object-cluster similarity $s(\mathbf{x}_i, C_j)$ is calculated with Eqs. (19), (20) or (21) for mixed, categorical, or numerical data, respectively. Additionally, in order to update the cluster information conveniently in step 4, two auxiliary matrices for each cluster are maintained. One matrix is to record the frequency of each categorical value occurring in this cluster, and the other matrix stores the mean vector of the numerical parts of all objects belonging to this cluster. Moreover, like the existing clustering algorithms with similar framework, the positions of initialized k seed points in step 2 will somewhat influence the final clustering accuracy. In the literature, different initialization methods for clustering performance improvement have been presented, such as Refs. [34,35] for numerical data clustering and Refs. [25,26] for categorical data clustering. However, to the best of our knowledge, such initialization refinement for mixed data clustering has not been studied yet. As the studies of this issue have been beyond the scope of this paper, we shall therefore utilize the random initialization method with multiple repetition to get the statistic information for clustering performance evaluation.

To illustrate the learning process of OCIL algorithm, we have generated a set of three-dimensional mixed data for clustering analysis as shown in Fig. 1(a). The different point patterns stand for the two categorical values and the numerical values are randomly distributed in the space of $[1,5] \times [1,3]$. Specially, we have selected two points which are very close to each other as the seed points of the two clusters. After one learning epoch, i.e., a scan of the whole data set, the obtained cluster membership by OCIL has been visualized in Fig. 1(b). It can be seen that most data points have been assigned to a reasonable cluster except four points. Subsequently, during the second learning epoch, these inaccurate points are reassigned and the final result is obtained. Furthermore, in the clustering space, we have drawn the moving trace for the center of all numerical vectors in each cluster as shown in Fig. 1(d). These traces give us a visual description about the change of members in the two clusters during the iterative learning of OCIL algorithm.

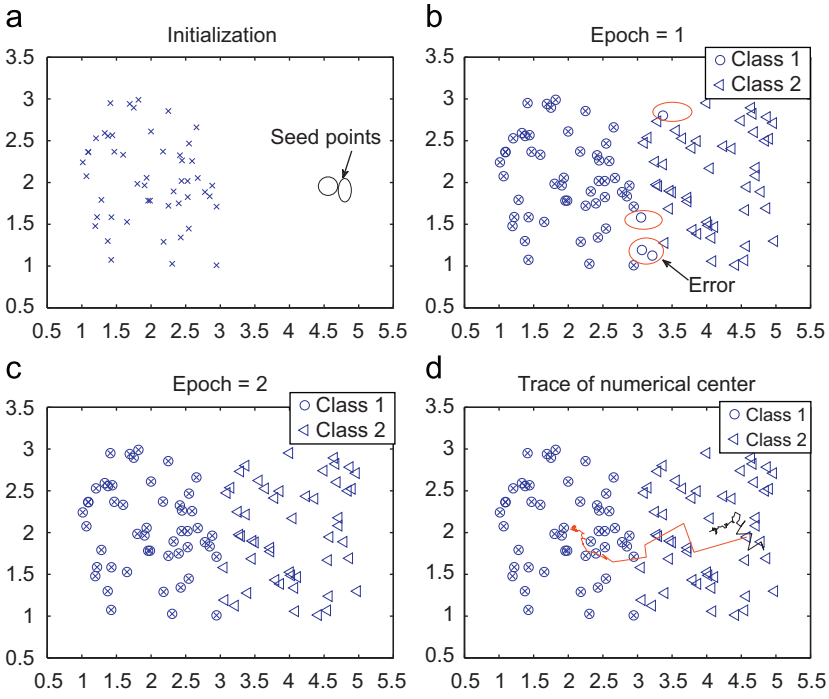

Fig. 1. Illustration of OCIL algorithm on synthetic data.

In practice, when Algorithm 1 is applied to purely numerical data and Euclidean distance is utilized to calculate $Dis(\mathbf{x}_i,\mathbf{c}_j)$, according to the similarity metric defined by Eq. (21), we can get

$$\begin{aligned} s(\mathbf{x}_i,C_j) \geq s(\mathbf{x}_i,C_r) &\Leftrightarrow \frac{\exp(-0.5\|\mathbf{x}_i-\mathbf{c}_j\|^2)}{\sum_{t=1}^{k}\exp(-0.5\|\mathbf{x}_i-\mathbf{c}_t\|^2)} \\ &\geq \frac{\exp(-0.5\|\mathbf{x}_i-\mathbf{c}_r\|^2)}{\sum_{t=1}^{k}\exp(-0.5\|\mathbf{x}_i-\mathbf{c}_t\|^2)} \\ &\Leftrightarrow \exp(-0.5\|\mathbf{x}_i-\mathbf{c}_j\|^2) \geq \exp(-0.5\|\mathbf{x}_i-\mathbf{c}_r\|^2) \\ &\Leftrightarrow \|\mathbf{x}_i-\mathbf{c}_j\|^2 \leq \|\mathbf{x}_i-\mathbf{c}_r\|^2, \end{aligned} \tag{23}$$

where "$\Leftrightarrow$" means "equivalent to". Then, the clustering criterion formulated by Eq. (22) can be simplified as

$$q_{ij}^{*} = \begin{cases} 1 & \text{if } \|\mathbf{x}_i-\mathbf{c}_j\|^2 \leq \|\mathbf{x}_i-\mathbf{c}_r\|^2 \ \forall 1 \leq r \leq k, \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

That is, each object will be assigned to the cluster whose centroid is closest to it. Under the circumstances, the proposed algorithm is equivalent to the k-means method.

Next, we further give the time complexity analysis of OCIL algorithm. It can be observed that the computation cost of step 1 is $O(mNd_c)$. For each iteration, the cost of the "**for**" statement is $O(mNkd_c+Nkd_u)$, where m is the average number of different values that can be chosen by each categorical attribute. Therefore, the total time cost of this algorithm is $O(t(mNkd_c+Nkd_u))$, where t is the number of iterations. From the practical viewpoint, we often have $k \ll N$, $m \ll N$ and $t \ll N$. Subsequently, the time complexity of this algorithm is $O(dN)$. Hence, the proposed algorithm is efficient for data clustering, particularly for a large data set.

5. Automatic selection of cluster number

Similar to the k-prototype [12] and k-modes [8,27] algorithms, the OCIL algorithm proposed in Section 4 still suffers from a selection problem of cluster number. That is, the cluster number k should be preassigned exactly equal to the true one; otherwise, OCIL will lead to an incorrect clustering result. To overcome this problem, in the following, we further present a penalized competitive clustering algorithm based on the object-cluster similarity metric. The competition and penalization mechanisms in this improved method will enable it to do automatic cluster number selection by gradually eliminating the redundant clusters.

5.1. Competition mechanism

Suppose N objects come from k^* unknown clusters. Initially, we set k ($k \geq k^*$) clusters $\{C_1,C_2,\ldots,C_k\}$, and assign one object as a seed point to each cluster. According to the competitive learning proposed in [36], given an input $\mathbf{x}_i$ each time, the winner C_v among the k clusters is determined by the dissimilarity between $\mathbf{x}_i$ and each cluster as well as the winning frequency of this cluster in the past. In our proposed method, the newly defined object-cluster similarity $s(\mathbf{x}_i,C_j)$ is utilized to estimate the similarity between an object and a cluster. Since the value of $s(\mathbf{x}_i,C_j)$ falls into the interval [0,1], we can evaluate the dissimilarity between object $\mathbf{x}_i$ and a cluster C_j with $(1-s(\mathbf{x}_i,C_j))$. Then, the clustering task based on the object-cluster similarity metric formulated by Eq. (1) can be rewritten as

$$\mathbf{Q}^{*} = \arg\min_{\mathbf{Q}} \left[\sum_{j=1}^{k}\sum_{i=1}^{N} q_{ij}(1-s(\mathbf{x}_i,C_j)) \right]. \tag{25}$$

Actually, $\mathbf{Q}^*$ obtained from Eq. (25) is equal to that obtained from Eq. (1), because we have

$$\begin{aligned} &\arg\min_{\mathbf{Q}} \left[\sum_{j=1}^{k}\sum_{i=1}^{N} q_{ij}(1-s(\mathbf{x}_i,C_j)) \right] \Leftrightarrow \arg\min_{\mathbf{Q}} \left[\sum_{j=1}^{k}\sum_{i=1}^{N} (q_{ij}-q_{ij}s(\mathbf{x}_i,C_j)) \right] \\ &\Leftrightarrow \arg\min_{\mathbf{Q}} \left[\sum_{j=1}^{k}\sum_{i=1}^{N} q_{ij} - \sum_{j=1}^{k}\sum_{i=1}^{N} q_{ij}s(\mathbf{x}_i,C_j) \right] \\ &\Leftrightarrow \arg\min_{\mathbf{Q}} \left[N - \sum_{j=1}^{k}\sum_{i=1}^{N} q_{ij}s(\mathbf{x}_i,C_j) \right] \\ &\Leftrightarrow \arg\max_{\mathbf{Q}} \left[\sum_{j=1}^{k}\sum_{i=1}^{N} q_{ij}s(\mathbf{x}_i,C_j) \right]. \end{aligned} \tag{26}$$

Subsequently, analogous to [36], given an object $\mathbf{x}_i$ each time, the winner C_v among the k clusters is determined by

$$v = \arg\min_{1 \leq j \leq k} [\gamma_j(1-s(\mathbf{x}_i,C_j))], \tag{27}$$

with the relative winning frequency γ_j of C_j defined as

$$\gamma_j = \frac{n_j}{\sum_{t=1}^{k} n_t}, \tag{28}$$

where n_j is the winning times of C_j in the past. That is, the winning chance of a cluster is controlled by the object-cluster similarity as well as its winning frequency in the past competitions. Here, reducing the winning rate of frequent winners is to solve the dead-unit problem encountered by competitive learning [36]. After selecting out the winning cluster C_v, we assign $\mathbf{x}_i$ to it and update the statistic information of C_v, which includes the center of numerical part $\mathbf{c}_v$ and the frequency of each categorical value accompanying with $\mathbf{x}_i$ in C_v. Meanwhile, the winning times of C_v is adjusted by

$$n_v^{(new)} = n_v^{(old)} + 1. \tag{29}$$

Therefore, a competitive learning version of the OCIL algorithm can be summarized as Algorithm 2.

Algorithm 2. Competitive learning based on object-cluster similarity metric (CL-OC).

Input: data set X, number of clusters k
Output: cluster label $Y=\{y_1,y_2,\ldots,y_N\}$
1. Select k initial objects, one for each cluster, and set $Y=\{0,0,\ldots,0\}$, $n_j=1$ for $j=1,2,\ldots,k$.
repeat
  Initialize *noChange*=*true*.
  for $i=1$ **to** N **do**
    2. $v=\arg\min_{1 \leq j \leq k}[\gamma_j(1-s(\mathbf{x}_i,C_j))]$
    3. Let $y_i^{(new)}=v$, $n_v^{(new)}=n_v^{(old)}+1$, and update the statistic information of C_v based on $\mathbf{x}_i$.
    if $y_i^{(new)} \neq y_i^{(old)}$ **then**
      noChange=*false*
    end if
  end for
until *noChange* is *true*

5.2. Penalization mechanism

It has been demonstrated in [9,11] that the penalization mechanism can enable the clustering algorithm to select the cluster number automatically during the learning process by gradually fading out the redundant clusters. Hence, in this paper,

we also utilize this mechanism to solve the selection problem of cluster number.

The basic idea of the penalization mechanism is that, for each input $\mathbf{x}_i$, not only the winning cluster is updated based on $\mathbf{x}_i$, but also the rival nearest to the winner (i.e., the runner-up) is penalized according to a specific criterion. Generally, in this kind of method, the cluster number k is initialized not less than the true one (i.e., $k \geq k^*$) and the main task is to fade out the redundant clusters. Therefore, in our approach, a weight is assigned to each cluster. This weight is utilized to measure the importance of each cluster to the whole cluster structure. Specifically, all clusters with an equal weight means that each of them has the same contribution to the cluster structure. In case a cluster has a very low weight, then the number of objects assigned to it will decrease and finally this cluster will be eliminated. Subsequently, similar to Eq. (27), given an object $\mathbf{x}_i$ each time, the winner C_v among k clusters satisfies

$$v = \arg \min_{1 \leq j \leq k} [\gamma_j (1 - \lambda_j s(\mathbf{x}_i, C_j))], \tag{30}$$

and its nearest rival C_r is determined by

$$r = \arg \min_{j \neq v} [\gamma_j (1 - \lambda_j s(\mathbf{x}_i, C_j))], \tag{31}$$

where λ_j is the weight of cluster C_j and the similarity between $\mathbf{x}_i$ and C_j is further regulated by it.

After selecting out the winning cluster and its nearest rival, on the one hand, we assign $\mathbf{x}_i$ to the winner C_v and update the statistic information of this cluster as well as its winning times. On the other hand, we further reward the winner by increasing its weight according to

$$\lambda_v^{(new)} = \lambda_v^{(old)} + \eta, \tag{32}$$

and meanwhile penalize the nearest rival C_r by decreasing its weight with

$$\lambda_r^{(new)} = \max(0, \lambda_r^{(old)} - \eta s(\mathbf{x}_i, C_r)), \tag{33}$$

where η is a small learning rate and the "max( )" function is to make sure that all the cluster weights are nonnegative. From Eq. (33), we can see that the rival-penalized strength increases with the similarity between $\mathbf{x}_i$ and the rival. Consequently, the main steps of the penalized competitive leaning based on the object-cluster similarity can be summarized as Algorithm 3.

Algorithm 3. Penalized competitive learning based on object-cluster similarity metric (PCL-OC).

Input: data set X, learning rate η and a initial value of k ($k \geq k^*$)
Output: cluster label $Y = \{y_1, y_2, \ldots, y_N\}$ and cluster number k^*
1. Select k initial objects, one for each cluster, and set $Y = \{0,0,\ldots,0\}$, $n_j = 1$ and $\lambda_j = 1$ for $j = 1,2,\ldots,k$.
repeat
  Initialize *noChange* = *true*.
  for $i = 1$ **to** N **do**
    2. Determine v and r according to Eqs. (30) and (31), respectively.
    3. Let $y_i^{(new)} = v$, $n_v^{(new)} = n_v^{(old)} + 1$, and update the statistic information of C_v based on $\mathbf{x}_i$.
    4. Update λ_v and λ_r using Eqs. (32) and (33), respectively.
    if $y_i^{(new)} \neq y_i^{(old)}$ **then**
      noChange = *false*
    end if
  end for
until *noChange* is *true*

After the clustering learning using Algorithm 3 (also called *PCL-OC algorithm* hereinafter), if there exists a cluster to which no objects belong, it will be regarded as a redundant one and simply neglected. In Fig. 2, we have visualized the learning process of PCL-OC algorithm with the same synthetic data set that was utilized in Section 4. Initially, the cluster number was set at three and an equal weight 1.0 was assigned to each cluster. Since this data set is very simple, the optional value range of learning rate η is relatively large. In our illustration, we have set $\eta = 0.01$ to get a fast convergence speed. After one learning epoch, we can find that the weight of the second cluster has decreased because it has

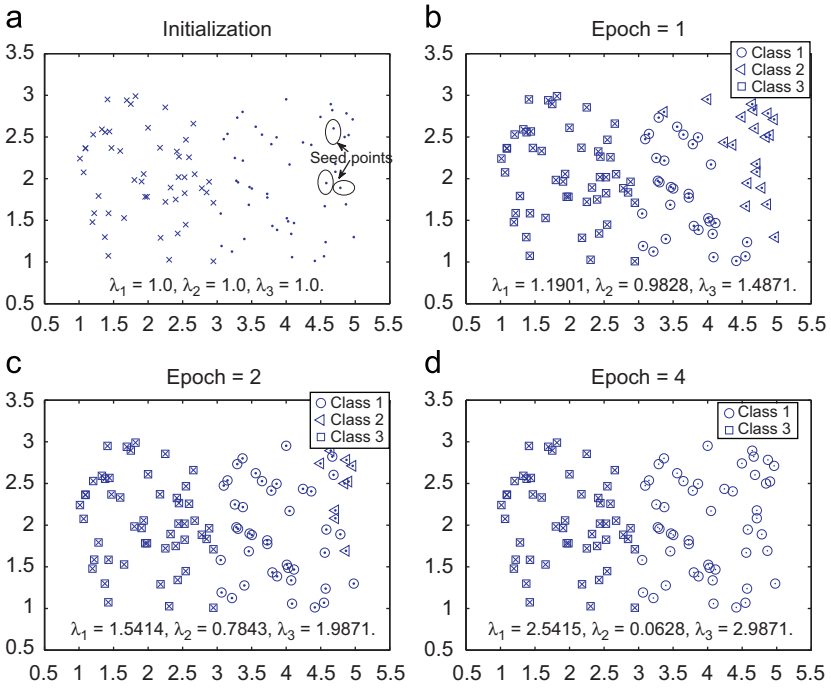

Fig. 2. Illustration of PCL-OC algorithm on synthetic data.

suffered the most penalization and gained the least cluster members. This penalization has been transmitted and strengthened during the following iterations and after the fourth epoch, no data points were assigned to the second cluster due to its low weight. If the iteration continues, one more epoch later we can get $\lambda_2 = 0$, which indicates that this cluster has been totally eliminated from the hypothetic cluster model.

6. Experiments

This section investigates the effectiveness of the proposed approaches for data clustering. We applied them to various data sets obtained from UCI Machine Learning Data Repository (URL: http://archive.ics.uci.edu/ml/) and compared their performance with the existing counterparts. In the experiments, the clustering accuracy [37] was estimated by

$$\mathrm{ACC} = \frac{\sum_{i=1}^{N} \delta(c_i, \mathrm{map}(l_i))}{N},$$

where N is the number of instances in the data set, c_i stands for the provided label, $\mathrm{map}(l_i)$ is a mapping function which maps the obtained cluster label l_i to the equivalent label from the data corpus, and the delta function $\delta(c_i, \mathrm{map}(l_i)) = 1$ only if $c_i = \mathrm{map}(l_i)$, otherwise 0. Correspondingly, the clustering error rate is computed as $e = 1 - \mathrm{ACC}$. The algorithms were coded with MATLAB and all the experiments were implemented by a desktop PC computer with Intel(R) Core(TM)2 Quad CPU, 2.40 GHz main frequency, and 4 GB DDR2 667 RAM.

6.1. Performance evaluation of OCIL algorithm

In this part, we experimentally investigated the performance of proposed OCIL algorithm. For comparative studies, the results of OCIL algorithm have been compared with k-means [4], k-prototype [12], and k-modes [8,27] algorithms because of two reasons: on the one hand, these algorithms all have the same time complexity: i.e., $O(dN)$. On the other hand, they have similar framework and procedure so that the effectiveness of the proposed similarity metric can be well evaluated. Please note that the OCIL algorithm is equivalent to the k-means algorithm as shown in Section 4 if the data have the numerical attributes only. Under the circumstances, the effectiveness of OCIL algorithm on numerical data set therefore becomes transparent and will not be investigated any more. In the following sub-sections, we shall focus on investigating the clustering performance of OCIL on mixed and categorical data sets, respectively.

6.1.1. Experiments on mixed data sets

Firstly, we investigated the performance of OCIL algorithm on mixed data. The information of the selected data sets is shown in Table 1. The performance of OCIL method on mixed data has been compared with k-prototype algorithm and k-means algorithm. Each algorithm has been executed 100 times on each data set and the clustering results are statistically summarized in Table 2. In k-prototype method, the distance regulation parameter γ was set at 0.5σ, where σ is the average standard deviation of numerical attributes. As for k-means, we utilized the single number representation method to covert categorical attributes into numerical ones. That is, each categorical value was represented by an arbitrarily chosen integer. Additionally, Euclidean distance metric was adopted to estimate the distances between numerical vectors in each method.

From Table 2, it can be observed that, with random initializations, the proposed parameter-free algorithm OCIL outperforms the k-prototype and k-means methods in terms of clustering accuracy. Further, as shown in Table 1, the ratios of categorical attributes to numerical attributes in the utilized data sets are different from each other, especially the Dermatology data, which has only one numerical feature but 33 categorical ones. Nevertheless, the OCIL has achieved a satisfactory clustering result. This indicates that the proposed object-cluster similarity metric is applicable to data in variant compound styles without using any parameter to adjust between categorical and numerical attributes. Additionally, for the last three data sets (i.e., German, Adult, and Dermatology) which have very uneven class distributions, the OCIL algorithm can give much improved accuracies compared to the other two methods. This result shows that, in comparison with numerically representing the distance between categorical values, the presented similarity metric in this paper is a more reasonable measurement for cluster analysis on mixed data and can well reveal the inherent cluster membership for either heterogeneous or homogeneous clusters. Moreover, comparing the average running time of OCIL and k-prototype

Table 1
Statistics of the mixed data sets.

Data set	Instance	Attribute (d_c+d_u)	Class	Class probabilities
Statlog heart	270	7+6	2	55.56% 44.44%
Heart disease	303	7+6	2	54.13% 45.87%
Credit approval	653	9+6	2	54.67% 45.33%
German credit	1000	13+7	2	70.0% 30.0%
Adult	30,162	8+6	2	75.11% 24.89%
Dermatology	366	33+1	6	30.6% 16.67% 19.67% 13.39% 14.21% 5.46%

Table 2
Clustering errors of OCIL on mixed data sets in comparison with k-prototype and k-means.

Data set	k-Means	k-Prototype	OCIL
Statlog	0.4047 ± 0.0071	0.2306 ± 0.0821	$\mathbf{0.1761} \pm \mathbf{0.0059}$
Heart	0.4224 ± 0.0131	0.2280 ± 0.0903	$\mathbf{0.1687} \pm \mathbf{0.0033}$
Credit	$0.4487 \pm \mathbf{0.0016}$	0.2619 ± 0.0976	$\mathbf{0.2437} \pm 0.0866$
German	0.3290 ± 0.0014	$0.3289 \pm \mathbf{0.0006}$	$\mathbf{0.3057} \pm 0.0009$
Adult	0.3869 ± 0.0067	0.3855 ± 0.0143	$\mathbf{0.2490} \pm \mathbf{0.0001}$
Dermatology	$0.7006 \pm \mathbf{0.0216}$	0.6903 ± 0.0255	$\mathbf{0.3026} \pm 0.0973$

Table 3
Comparison of the average convergence time between k-prototype and OCIL.

Data set	k-Prototype (s)	OCIL (s)
Statlog	0.0519	**0.0498**
Heart	0.0639	**0.0491**
Credit	0.1323	**0.1282**
German	**0.2999**	0.3342
Adult	15.2795	**3.5447**
Dermatology	0.3674	**0.1811**

Table 4
Statistics of the categorical data sets.

Data set	Instance	Attribute	Class	Class probabilities
Soybean	47	35	4	21.28% 21.28% 21.28% 36.16%
Breast	699	9	2	65.52% 34.48%
Vote	435	16	2	61.38% 38.62%
Zoo	101	16	7	40.59% 19.8% 4.95% 12.87% 3.97% 7.92% 9.9%

algorithms listed in Table 3, we can find that the total running time of OCIL is no more than the one of k-prototype although OCIL needs additional time to calculate the weight of each categorical attribute. That is because OCIL converges faster than k-prototype in most cases.

6.1.2. Experiments on categorical data sets

Next, we further investigated the performance of OCIL algorithm on purely categorical data. The information of utilized four different benchmark data sets has been summarized in Table 4. To conduct comparison study, we have also implemented the other two existing categorical data clustering algorithms: original k-modes (H's k-modes) [8] and k-modes with Ng's dissimilarity metric (N's k-modes) [27].

In the experiment, each algorithm was conducted with random initializations. Table 5 lists the average value and standard deviation in error obtained by OCIL and the other two algorithms, respectively. It can be seen that, for categorical data learning, the proposed clustering method has competitive advantage in terms of clustering accuracy and robustness compared with the other two methods. This superiority of OCIL method mainly owes to two merits of the object-cluster similarity metric. The first one is that, in the proposed metric, the similarity between given categorical attribute value and a cluster depends on the distribution of this value within the cluster, but not the numerical distance between this value and the corresponding attribute value of the cluster mode. In N's k-modes, when calculating the distance between an object and a cluster mode, the frequencies of attribute values within the cluster are considered if the object and cluster mode have the same values. Hence, the performance of N's k-modes is better than H's k-modes on all the data sets we have tried so far. However, when the object and cluster mode have different attribute values, N's k-modes also simply assumes the distance is 1. The other merit is that we do not utilize mode to represent each cluster but calculate the similarity based on the cluster's statistic information in this new metric. In k-modes algorithms, a cluster mode is represented by the most frequent attribute values within the cluster. That is, only one value is selected as the representation for each attribute even though there may be some value with proximate frequency. Hence, the information of a cluster actually cannot be completely presented by the defined mode for categorical data.

Additionally, we further evaluated the convergence speed of the proposed method on categorical data clustering. Table 6 lists the average convergence time over 100 runs cost by each method.

Table 5
Comparison of the clustering errors obtained by three different methods on categorical data sets.

Data set	H's k-modes	N's k-modes	OCIL
Soybean	0.1691 ± 0.1521	$\mathbf{0.0964} \pm 0.1404$	$0.1017 \pm \mathbf{0.1380}$
Breast	0.1655 ± 0.1528	0.1356 ± 0.0016	$\mathbf{0.0934} \pm \mathbf{0.0009}$
Vote	0.1387 ± 0.0066	0.1345 ± 0.0031	$\mathbf{0.1213} \pm \mathbf{0.0010}$
Zoo	0.2873 ± 0.1083	$0.2730 \pm \mathbf{0.0818}$	$\mathbf{0.2681} \pm 0.0906$

Table 6
Comparison of the average convergence time between k-modes and OCIL.

Data set	H's k-modes (s)	N's k-modes (s)	OCIL (s)
Soybean	0.0176	0.0189	**0.0058**
Breast	0.1044	0.1515	**0.0540**
Vote	0.0733	0.0862	**0.0354**
Zoo	0.0418	0.0514	**0.0098**

It can be observed that the convergence time of the proposed method is much faster than the k-modes with the improvement of 60% on average in all cases we have tried so far. Based on the analysis of experimental results, the significant advantage of running time with OCIL algorithm on categorical data can be owed to the following two aspects: on the one hand, the convergence speed of OCIL is faster than k-modes as the number of learning epoches needed by OCIL is smaller than that needed by k-modes on average; on the other hand, the computational cost of OCIL in each learning epoch is less than the k-modes because k-modes needs to update the cluster modes in each learning step while the OCIL need not.

6.2. Performance evaluation of PCL-OC algorithm

To investigate the effectiveness of the proposed penalized competitive learning method, we have applied it to different real data sets, including purely categorical data and mixed data. Moreover, to the best of our knowledge, clustering algorithm with automatic cluster number selection for mixed data has not been studied yet in the literature. Therefore, in our experiments, we only take the k-prototype algorithm as an example to comparatively show the outstanding performance of PCL-OC that is capable of determining the number of clusters automatically.

As a rule of thumb, the learning rate η in the penalization mechanism can be set as $\rho(k/N)$, where ρ is a small coefficient and N is the number of objects in the given data set. That is, the optimal learning rate increases with the initial cluster number k but decreases with N. The value of ρ also has small variation for different size of data set. Generally, a too small value of ρ will lead to an insufficient penalization process and the redundant clusters cannot be completely driven out from the input space. Conversely, a too large value of ρ will cause an excessive penalization, whereby the initial clusters will be over-eliminated. By the rule of thumb, it is appropriate to set the value of ρ between 0.001 and 0.003 for Soybean data that contains 47 instances only. For other data sets with hundreds of instances, ρ can be set between 0.003 and 0.006. In the following two experiments, the value of ρ is set at 0.001 and 0.005, respectively.

In the first experiment, we took the Vote data set for instance to show PCL-OC algorithm's ability of automatical cluster number selection on real data set. To show the details of learning process, we utilized $\alpha_j(t)$ to record the proportion of objects among the whole data set that has been assigned to the jth cluster during the tth

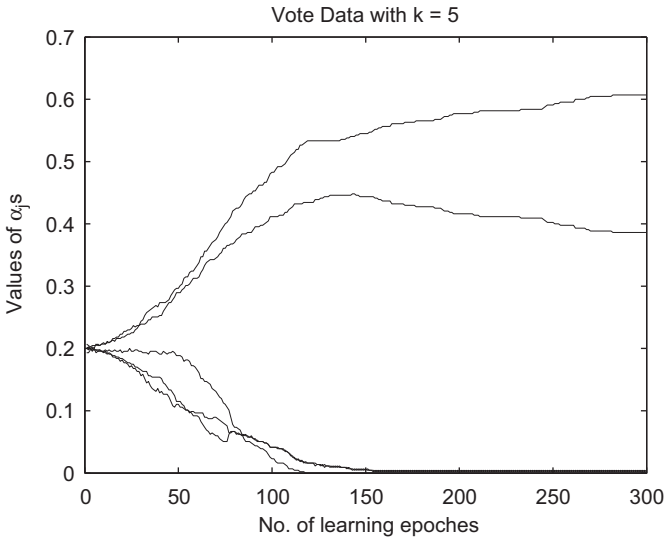

Fig. 3. Learning curves of α_js obtained by PCL-OC on Vote data with k=5.

Table 7
Clustering results of PCL-OC on different data sets with variant settings of k.

Data set	k^*	k	Cluster no. (Mean ± std)		Error rate (mean ± std)
			PCL-OC	k-Prototype	PCL-OC
Heart	2	3	**1.70** ± 0.47	3.0 ± 0.0	0.2315 ± 0.1073
		4	**1.80** ± 0.41	4.0 ± 0.0	0.2507 ± 0.0752
		5	**2.20** ± 0.61	5.0 ± 0.0	0.2458 ± 0.1191
Credit	2	3	**1.86** ± 0.36	3.0 ± 0.0	0.2645 ± 0.1063
		4	**2.16** ± 0.67	4.0 ± 0.0	0.2609 ± 0.1089
		5	**2.34** ± 0.82	5.0 ± 0.0	0.2734 ± 0.1048
Soybean	4	5	**4.42** ± 0.50	5.0 ± 0.0	0.0853 ± 0.0790
		6	**4.18** ± 0.77	6.0 ± 0.0	0.1106 ± 0.1025
		7	**4.04** ± 1.14	7.0 ± 0.0	0.1021 ± 0.0932
Vote	2	3	**2.0** ± 0.0	3.0 ± 0.0	0.1196 ± 0.0001
		4	**2.0** ± 0.0	4.0 ± 0.0	0.1196 ± 0.0002
		5	**2.0** ± 0.0	5.0 ± 0.0	0.1198 ± 0.0005

learning epoch. Initially, five clusters were generated and the seed points were randomly selected in the input space. The learning curves of α_js over the epoches obtained by the PCL-OC algorithm are shown in Fig. 3. It can be seen that the values of three α_js have converged to around zero after about 150 learning epoches. It means that these three redundant clusters have been eliminated from the whole clustering structure because few objects will be assigned to them. Meanwhile, the obtained values of the other two α_js are 0.6069 and 0.3862, which are approximate to the proportions of the two true clusters in the data set. Hence, the PCL-OC algorithm has successfully identified the true cluster number during the learning process.

In the second experiment, we investigated the performance of PCL-OC on different data sets with variant settings of k. In total, four data sets were utilized: two mixed data sets with numerical and categorical attributes and two with purely categorical attributes. These data sets have different cluster numbers and class distributions. For each data set, the PCL-OC has been executed 50 times and the learning results are summarized in Table 7. It can be seen that the PCL-OC algorithm can give a good estimation of the cluster number in each setting of k. For comparison, we have also implemented k-prototype algorithm [12] under the same environment as PCL-OC. Evidently, the k-prototype algorithm needs to preassign the number of clusters exactly without the capability of selecting the cluster number automatically. As a result, the clustering accuracy of the k-prototype is seriously degraded when the number k of clusters is not selected appropriately in advance.

7. Conclusion

In this paper, we have proposed a general clustering framework based on object-cluster similarity, through which a unified similarity metric for both categorical and numerical attributes has been presented. Under this new metric, the object-cluster similarity for categorical and numerical attributes are with the same scale, which is beneficial to clustering analysis on various data types. Subsequently, an iterative algorithm has been introduced to implement the data clustering. The advantages of the proposed method have been experimentally demonstrated in comparison with the existing counterparts. Additionally, to overcome the cluster number selection problem, a penalized competitive learning algorithm has been presented within the proposed clustering framework. The competition and penalization mechanisms embedded in this method are capable of selecting number of clusters automatically by gradually fading out the redundant clusters during the clustering process. Experiments on different benchmark data sets have shown the effectiveness and efficiency of the proposed approach.

Conflict of interest statement

None declared.

Acknowledgments

The work described in this paper was supported by the Faculty Research Grant of Hong Kong Baptist University with the Project Code: FRG2/11-12/067, and the NSFC under grant 61272366.

References

[1] R.S. Michalski, I. Bratko, M. Kubat, Machine Learning and Data Mining: Methods and Applications, Wiley, New York, 1998.
[2] W. Cai, S. Chen, D. Zhang, Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation, Pattern Recognition 40 (3) (2007) 825–838.
[3] A.W.-C. Liew, H. Yan, M. Yang, Pattern recognition techniques for the emerging field of bioinformatics: a review, Pattern Recognition 38 (11) (2005) 2055–2073.
[4] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 1967, 281–297.
[5] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Series B (Methodological) 39 (1) (1977) 1–38.
[6] C.C. Hsu, Generalizing self-organizing map for categorical data, IEEE Transactions on Neural Networks 17 (2) (2006) 294–304.
[7] C. Li, G. Biswas, Unsupervised learning with mixed numeric and nominal data, IEEE Transactions on Knowledge and Data Engineering 14 (4) (2002) 673–690.
[8] Z. Huang, A fast clustering algorithm to cluster very large categorical data sets in data mining, in: Proceedings of the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, 1997, pp. 1–8.
[9] L. Xu, A. Krzyzak, E. Oja, Rival penalized competitive learning for clustering analysis, RBF net, and curve detection, IEEE Transactions on Neural Networks 4 (4) (1993) 636–648.
[10] H. Liao, M.K. Ng, Categorical data clustering with automatic selection of cluster number, Fuzzy Information and Engineering 1 (1) (2009) 5–25.
[11] Y.M. Cheung, Maximum weighted likelihood via rival penalized EM for density mixture clustering with automatic model selection, IEEE Transactions on Knowledge and Data Engineering 17 (6) (2005) 750–761.
[12] Z. Huang, Clustering large data sets with mixed numeric and categorical values, in: Proceedings of the First Pacific-Asia Conference on Knowledge Discovery and Data Mining, 1997, pp. 21–34.
[13] S. Guha, R. Rastogi, K. Shim, ROCK: a robust clustering algorithm for categorical attributes, Information Systems 25 (5) (2001) 345–366.
[14] 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.
[15] M.J. Zaki, M. Peters, CLICK: mining subspace clusters in categorical data via k-partite maximal cliques, in: Proceedings of the twenty-first International Conference on Data Engineering, 2005, pp. 355–356.
[16] D. Barbara, J. Couto, Y. Li, COOLCAT: an entropy-based algorithm for categorical clustering, in: Proceedings of the 11th ACM Conference on Information and Knowledge Management, 2002, pp. 582–589.
[17] P. Andritsos, P. Tsaparas, R.J. Miller, K.C. Sevcik, LIMBO: scalable clustering of categorical data, in: Proceedings of the 9th International Conference on Extending Database Technology, 2004, pp. 123–146.
[18] N. Tishby, F.C. Pereira, W. Bialek, The information bottleneck method, in: Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing, 1999, pp. 368–377.
[19] D.W. Goodall, A new similarity index based on probability, Biometric 22 (4) (1966) 882–907.
[20] Z. He, X. Xu, S. Deng, Scalable algorithms for clustering large datasets with mixed type attributes, International Journal of Intelligence Systems 20 (2005) 1077–1089.
[21] H. Luo, F. Kong, Y. Li, Clustering mixed data based on evidence accumulation, in: X. Li, O. R. Zaiane, Z. Li (Eds.), Advanced Data Ming and Applications, Lecture Notes in Computer Science, vol. 4093, 2006, pp. 348–355.
[22] P. Cheeseman, J. Stutz, Bayesian classification (AutoClass): theory and results, in: Advances in Knowledge Discovery and Data Mining, 1996.
[23] Z. Huang, Extensions to the k-modes algorithm for clustering large data sets with categorical values, Data Mining and Knowledge Discovery 2 (3) (1998) 283–304.
[24] Z. Huang, M. Ng, A note on k-modes clustering, Journal of Classification 20 (2) (2003) 257–261.
[25] S.S. Khan, S. Kant, Computation of initial modes for k-modes clustering algorithm using evidence accumulation, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07), 2007, pp. 2784–2789.
[26] F. Cao, J. Liang, L. Bai, A new initialization method for categorical data clustering, Expert Systems with Applications 36 (7) (2009) 10223–10228.

[27] M.K. Ng, M.J. Li, J.Z. Huang, Z. He, On the impact of dissimilarity measure in *k*-modes clustering algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (3) (2007) 503–507.
[28] H. Akaike, A new look at the statistical model identification, IEEE Transactions on Automatic Control 19 (6) (1974) 716–723.
[29] G. Schwarz, Estimating the dimension of a model, The Annals of Statistics 6 (2) (1978) 461–464.
[30] Y.M. Cheung, A competitive and cooperative learning approach to robust data clustering, in: Proceedings of IASTED International Conference on Neural Networks and Computational Intelligence, 2004, pp. 131–136.
[31] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A survey of kernel and spectral methods for clustering, Pattern Recognition 41 (1) (2008) 176–190.
[32] A.K. Jain, Data clustering: 50 years beyond *k*-means, Pattern Recognition Letters 31 (8) (2010) 651–666.
[33] J. Basak, R. Krishnapuram, Interpretable hierarchical clustering by constructing an unsupervised decision tree, IEEE Transactions on Knowledge and Data Engineering 17 (1) (2005) 121–132.
[34] J.P. na, J. Lozano, P.L. naga, An empirical comparison of four initialization methods for the *k*-means algorithm, Pattern Recognition Letters 20 (10) (1999) 1027–1040.
[35] S.S. Khan, A. Ahmad, Cluster center initialization algorithm for *k*-means clustering, Pattern Recognition Letters 25 (11) (2004) 1293–1302.
[36] S.C. Ahalt, A.K. Krishnamurthy, P. Chen, D.E. Melton, Competitive learning algorithms for vector quantization, Neural Networks 3 (3) (1990) 277–290.
[37] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Advances in Neural Information Processing Systems, 2005.

Yiu-ming Cheung (SM'06) is a professor at Department of Computer Science in Hong Kong Baptist University. He received the Ph.D. degree from the Department of Computer Science and Engineering, The Chinese University of Hong Kong in 2000, and then joined the Department of Computer Science at Hong Kong Baptist University in 2001. His current research interests are in the fields of machine learning and information security, particularly the topics on clustering analysis, blind source separation, neural networks, nonlinear optimization, watermarking and lip-reading. He is the founding chairman of IEEE (Hong Kong) Computational Intelligence Chapter. Currently, he is also the associate editor of Knowledge and Information Systems, as well as the guest co-editor and editorial board member of the several international journals.

Hong Jia received the B.S. and master degrees from Huazhong University of Science and Technology, China, in 2008 and 2010, respectively. She is currently the Ph.D. student at the Department of Computer Science in Hong Kong Baptist University, Hong Kong. Her research interests include clustering analysis and pattern recognition.