An Ordinal Data Clustering Algorithm with Automated Distance Learning

Yiqun Zhang, Yiu-ming Cheung*
Department of Computer Science, Hong Kong Baptist University,
Kowloon Tong, Hong Kong SAR, China,
{yqzhang, ymc}@comp.hkbu.edu.hk

Abstract

Clustering ordinal data is a common task in data mining and machine learning fields. As a major type of categorical data, ordinal data is composed of attributes with naturally ordered possible values (also called categories interchangeably in this paper). However, due to the lack of dedicated distance metric, ordinal categories are usually treated as nominal ones, or coded as consecutive integers and treated as numerical ones. Both these two common ways will roughly define the distances between ordinal categories because the former way ignores the order relationship and the latter way simply assigns identical distances to different pairs of adjacent categories that may have intrinsically unequal distances. As a result, they may produce unsatisfactory ordinal data clustering results. This paper, therefore, proposes a novel ordinal data clustering algorithm, which iteratively learns: 1) The partition of ordinal dataset, and 2) the inter-category distances. To the best of our knowledge, this is the first attempt to dynamically adjust inter-category distances during the clustering process to search for a better partition of ordinal data. The proposed algorithm features superior clustering accuracy, low time complexity, fast convergence, and is parameter-free. Extensive experiments show its efficacy.

Introduction

Ordinal data is usually collected from questionnaires, evaluation systems, etc. As a major type of categorical data, possible values of an ordinal attribute are a limited number of naturally ordered categories (Agresti 1996; Allen and Seaman 2007), e.g., {accept, neutral, reject}. In many data mining and machine learning tasks, it is common to analyze ordinal data by clustering, in which distance measurement plays a main role (Ng et al. 2007). Since the values of ordinal data are not quantitative, the distances of ordinal data are not well-defined. Therefore, ordinal data is usually treated in either of the following two ways: 1) Directly define the distances between ordinal categories, and treat ordinal data as nominal one in the clustering; 2) Code ordinal categories as consecutive integers, and treat the coded data as numerical one in the clustering.

In the former way, all the existing distance metrics and measures that are designed for categorical data can be directly used to define the distances for ordinal data. Hamming Distance Metric (HDM) is the most popular one. However, its distance is simply binary (i.e., 0 for identical categories and 1 for different categories), which is too simple to suitable for ordinal data. By contrast, Entropy-Based Measure (EBM) (Barbará, Li, and Couto 2002; Li, Ma, and Ogihara 2004), Association-Based Distance Metric (ABDM) (Le and Ho 2005), Ahmad's Distance Metric (ADM) (Ahmad and Dey 2007), Context-Based Distance Metric (CBDM) (Ienco, Pensa, and Meo 2012), Object-Cluster Similarity Measure (OCSM) (Cheung and Jia 2013), and Jia's Distance Metric (JDM) (Jia, Cheung, and Liu 2016) attempt to more reasonably define the distances by exploiting statistic information of categories, including frequency probabilities, conditional probability distributions, and so on. However, they are originally designed for nominal data only, and the distances defined by them may violate the order relationship among ordinal categories. Most recently, Entropy-Based Distance Metric (EBDM) (Zhang and Cheung 2018) has been proposed, which adopts cumulative entropy as a measure to simultaneously take into account the statistic information and order relationship for ordinal data distance measurement. Unfortunately, it assumes that all the attributes are equally important and independent of each other, which may not always be true from the practical viewpoint. By adopting the above-mentioned metrics and measures, existing categorical data clustering algorithms, including the conventional K-Modes (KMD), the representative Weighted K-Modes (WKMD) (Chan et al. 2004), the state-of-the-art Weighted OCIL (WOCIL) (Jia and Cheung 2018), etc., will produce unsatisfactory ordinal data clustering results due to the unreasonably defined distances.

In the latter way, categories within the same attribute are usually coded as consecutive integers according to their ranking, for example, ordinal categories {accept, neutral, reject} are coded as $\{3, 2, 1\}$. In this way, ordinal data is converted into numerical one, and the existing numerical data clustering algorithms, including the conventional K-Means (KMS), the representative Weighted K-Means (WKMS) (Huang et al. 2005), the state-of-the-art WOCIL (Jia and Cheung 2018), etc., are applicable for the clustering of coded ordinal data. Although the coding effectively preserves the order information of ordinal data, it unreasonably assigns identical distances to different pairs of ad-

*Corresponding author

jacent categories that may have intrinsically unequal distances. For instance, given an ordinal attribute with five categories {excellent, very-good, good, fair, poor}, the difference between "excellent" and "very-good" is usually smaller than the difference between "fair" and "poor", because the former two are different in quantity whilst the latter two are different in quality. Evidently, the simple coding may twist the natural distances of ordinal attributes, and may thus lead to unsatisfactory clustering results.

In this paper, we propose a novel k-mode-type Distance Learning-based Clustering (DLC) algorithm, which dynamically adjusts the inter-category distances during the clustering process to search for a better partition of ordinal data. However, a difficulty lies in how to efficiently describe all the inter-category distances with preserving their order relationship. To tackle this, we only assign weights between the adjacent categories, and the distance between any two non-adjacent categories is jointly described by the weights between them. In this way, for an attribute A_r with v_r categories, only $v_r - 1$ weights are utilized to indicate the $\frac{v_r(v_r-1)}{2}$ inter-category distances with preserving their order relationship. To tackle another difficulty, i.e., how to reasonably adjust the weights, we design a novel measure that computes the expected effectiveness of adjusting a weight for forming more compact clusters (i.e., the clusters with more similar intra-cluster data objects). All the weights are jointly adjusted according to their expected effectiveness measured in different clusters. According to our design, the nontrivial ordinal data distance learning problem becomes achievable. Because DLC integrates the distance learning and the clustering process to automatically learn more suitable distances for the clustering task, it has superior clustering performance. Moreover, it features low time complexity, fast convergence, and is parameter-free. Extensive experiments on different real and benchmark datasets show its efficacy.

Related Work

For categorical data distance measurement, HDM is a commonly used one. Since the categories of an ordinal attribute are with different orders, the binary distances produced by HDM cannot well reflect the difference degrees between different ordinal categories. ADM and ABDM (Ahmad and Dey 2007; Le and Ho 2005) have been proposed to more finely define the inter-category distances. They adopt a common basic idea that two similar categories also have similar corresponding values on the other attributes. However, they assume all the attributes are interdependent, which may not always be true from the practical viewpoint. Thus, CBDM (Ienco, Pensa, and Meo 2012) is proposed to select relevant attributes for defining the inter-category distances. Later, JDM (Jia, Cheung, and Liu 2016) has been proposed, which further considers the case that all the attributes are independent of each other. In the literature, several entropy-based measures have also been proposed to compute the suitability of inserting an object into a cluster (Barbará, Li, and Couto 2002; Li, Ma, and Ogihara 2004). The basic idea is that if a new object is very similar to the objects in a cluster, the entropy of this cluster will not increase a lot after adopting the new object. Another measure called OCSM (Cheung and Jia 2013) has been proposed to indicate the object-cluster similarity by using the occurrence probabilities of the object values in the target cluster.

Because all the above-mentioned categorical data metrics and measures are actually designed for nominal data only, they will ignore the order information for distance/similarity measurement of ordinal data. Therefore, a distance metric called EBDM (Zhang and Cheung 2018) has recently been proposed for ordinal data clustering. It takes into account the order information for quantifying the inter-category distances from the perspective of information theory. Although its clustering performance is quite good, its effectiveness is still limited because it assumes that all the attributes are independent of each other, which is usually unreasonable.

Clustering algorithms can be categorized according to their suited data type. For categorical data clustering, the conventional KMD (Huang 1998) is easy to use, but it treats each attribute equally. Thus, its attribute weighting version WKMD (Chan et al. 2004) has been proposed to achieve better clustering performance. However, since they use the 'modes' to represent clusters, order relationship among ordinal categories will be ignored when computing the object-cluster distance. The state-of-the-art WOCIL (Jia and Cheung 2018) has been recently proposed to further detect potential subspace clusters. Since it adopts OCSM (Cheung and Jia 2013) as similarity measure for categorical data clustering, it still ignores the order information of ordinal data. Numerical data clustering algorithms, including the conventional KMS (MacQueen 1967), the representative WKMS (Huang et al. 2005), the state-of-the-art WOCIL (Jia and Cheung 2018), etc., are applicable to the coded ordinal data. Although the order information can be preserved by the coding, inter-category distances are twisted due to the identical distances assigned to different pairs of adjacent categories.

Proposed Method

Learning attribute weights has achieved huge success by the existing clustering algorithms (Chan et al. 2004; Huang et al. 2005; Jia and Cheung 2018). Assigning a weight to an attribute is equivalent to assigning identical weights to all the inter-category distances of this attribute. When the distances of a dataset are well-defined like the distances of numerical data, weighting each attribute as a whole is reasonable. However, the inter-category distances of ordinal data are not well-defined in general (Zhang and Cheung 2018). In this paper, we assume that the inter-category distances may have different contributions in forming compact clusters, and we propose a novel Distance Learning-based Clustering (DLC) algorithm, which decomposes the task of weighting attributes into sub-tasks of weighting the inter-category distances of attributes, to achieve more accurate partition of ordinal data.

Problem Description: Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ be an ordinal dataset with n data objects represented by m attributes A_1, A_2, ..., A_m with $v_1, v_2, ..., v_m$ categories, respectively. The v_r categories of an attribute A_r are ordered as $o_{r,1} \prec o_{r,2} \prec ... \prec o_{r,v_r}$, where the symbol "$\prec$" represents that the categories on its right ranked higher (have

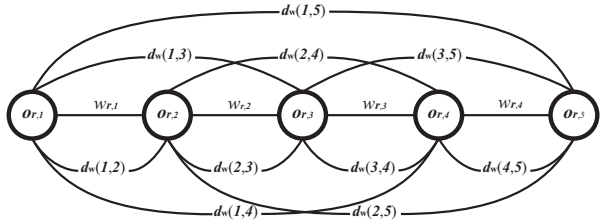

Figure 1: Distances of Attribute A_r ($v_r = 5$).

larger order values) than the categories on its left. The subscript of a category $o_{r,s}$ represents that it belongs to the r^{th} attribute and ranked s^{th} among the categories of the r^{th} attribute. The goal of ordinal data clustering is defined as the problem of minimizing the difference among intra-cluster data objects, and the objective function Z can be written as:

$$Z(\mathbf{Q}, \mathbf{W}) = \sum_{i=1}^{n}\sum_{j=1}^{k} q_{i,j} D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{C}_j), \quad (1)$$

where $\mathbf{Q}$ is an $n \times k$ partition matrix of $\mathbf{X}$. Since we assume crisp partition-based clustering, values of $\mathbf{Q}$ satisfy $\sum_{j=1}^{k} q_{i,j} = 1$ and $q_{i,j} \in \{0, 1\}$, $i \in \{1, 2, ..., n\}$. $D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{C}_j)$ denotes the weighted object-cluster distance between an object $\mathbf{x}_i$ and a cluster $\mathbf{C}_j$. $\mathbf{W}$ contains a set of m vectors $\{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m\}$, each of which contains a set of weights describing the inter-category distances of an attribute. We only assign weights between adjacent categories as shown in Figure. 1, and describe the distance between any two categories by adding all the weights between them. In this way, only $v_r - 1$ weights, i.e., $\mathbf{w}_r = \{w_{r,1}, w_{r,2}, ..., w_{r,v_r-1}\}$, are needed to describe all the $\frac{v_r(v_r-1)}{2}$ inter-category distances of A_r. To avoid the collapse of order relationship caused by negative weights and ensure meaningful optimization, the weights satisfy $\sum_{r=1}^{m}\sum_{s=1}^{v_r-1} w_{r,s} = 1$ and $0 \leq w_{r,s} \leq 1$. To minimize Z, we iteratively solve the following two problems:

- $\boldsymbol{P}_1$: Fix $\mathbf{W}$, minimize Z by adjusting $\mathbf{Q}$;
- $\boldsymbol{P}_2$: Fix $\mathbf{Q}$, reduce Z by adjusting $\mathbf{W}$.

$\boldsymbol{P}_1$ is solved by:

$$q_{i,j} = \begin{cases} 1 & , \quad if\ j = \arg\min_y D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{C}_y) \\ 0 & , \quad else, \end{cases} \quad (2)$$

$i \in \{1, 2, ..., n\}$ and $y \in \{1, 2, ..., k\}$. The weighted object-cluster distance $D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{C}_y)$ is defined as:

$$D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{C}_y) = \sum_{r=1}^{m}\sum_{s=1}^{v_r} d_{\mathbf{w}}(\kappa(x_{i,r}), s) \cdot u_{y,r,s}, \quad (3)$$

where $\kappa(x_{i,r})$ fetches the order value of $x_{i,r}$, for example, if $x_{i,r} = o_{r,t}$, $\kappa(x_{i,r}) = t$. $u_{y,r,s} = \frac{\sigma_{o_{r,s}}(\mathbf{C}_y)}{\sigma(\mathbf{C}_y)}$ is the occurrence probability of $o_{r,s}$ in $\mathbf{C}_y$, where $\sigma(\mathbf{C}_y)$ and $\sigma_{o_{r,s}}(\mathbf{C}_y)$ count the number of objects in $\mathbf{C}_y$, and the number of objects with their r^{th} values equal to $o_{r,s}$ in $\mathbf{C}_y$, respectively. $d_{\mathbf{w}}(\kappa(x_{i,r}), s)$ is the distance between $x_{i,r}$ and $o_{r,s}$, which is defined as:

$$d_{\mathbf{w}}(\kappa(x_{i,r}), s) = \begin{cases} \sum_{h=\min(\kappa(x_{i,r}),s)}^{\max(\kappa(x_{i,r}),s)-1} w_{r,h} & , \quad if\ \kappa(x_{i,r}) \neq s \\ 0 & , \quad if\ \kappa(x_{i,r}) = s. \end{cases} \quad (4)$$

Remarks: The defined distance satisfy $d_{\mathbf{w}}(s, t) \leq d_{\mathbf{w}}(g, h)$, if $\max(g, h) \geq \max(s, t)$, $\min(g, h) \leq \min(s, t)$, $g, s, t, h \in \{1, 2, ..., v_r\}$. That is, $d_{\mathbf{w}}(\cdot, \cdot)$ guarantees that the distance between two categories $o_{r,s}$ and $o_{r,t}$ is not larger than the distance between another two categories $o_{r,g}$ and $o_{r,h}$ that are not ordered between $o_{r,s}$ and $o_{r,t}$, which is consistent with the order relationship among categories.

The weighted object-cluster distance defined in Eq. (3) can finely detect the order differences between $x_{i,r}$ and the r^{th} values of the objects in $\mathbf{C}_y$ based on the present $\mathbf{W}$. The reasons why we do not use the *object-center distance* (MacQueen 1967), *object-mode distance* (Huang 1998), and *object-cluster similarity* (Cheung and Jia 2013) for solving $\boldsymbol{P}_1$ are discussed below:

- *Object-center distance* is suitable for data with well-defined distances. Computing object-center distances for coded ordinal data will produce incorrect clustering results as we discussed in the Introduction.
- Both *object-mode distance* and *object-cluster similarity* ignore the order relationship among ordinal categories, and may thus produce incorrect clustering results.

By solving $\boldsymbol{P}_1$, all the n objects are assigned to their closest clusters based on the present $\mathbf{W}$, and we obtain new $\mathbf{Q}$. Then, $\boldsymbol{P}_2$ is solved by:

$$w_{r,s} = \frac{\Phi_r}{\sum_{g=1}^{m}\Phi_g} \cdot \sum_{j=1}^{k}\left(\frac{b_{j,r,s}}{\sum_{t=1}^{v_r-1} b_{j,r,t}} \cdot \frac{B_{j,r}}{\sum_{h=1}^{k} B_{h,r}}\right), \quad (5)$$

$r \in \{1, 2, ..., m\}$ and $s \in \{1, 2, ..., v_r - 1\}$. $\frac{b_{j,r,s}}{\sum_{t=1}^{v_r-1} b_{j,r,t}}$ is the new $w_{r,s}$ computed according to $\mathbf{C}_j$, $\frac{B_{j,r}}{\sum_{h=1}^{k} B_{h,r}}$ weights the contribution of $\mathbf{C}_j$ in deciding all the weights in $\mathbf{w}_r$, and $\frac{\Phi_r}{\sum_{g=1}^{m}\Phi_g}$ weights the overall importance of $\mathbf{w}_r$. According to Eq. (5), all the weights in $\mathbf{W}$ are updated based on the present $\mathbf{Q}$. In the following, we present the definitions of $b_{j,r,s}$, $B_{j,r}$, and Φ_r, and discuss their roles in the learning.

If adjusting $w_{r,s}$ is expected to achieve better reduction on Z, then $w_{r,s}$ should be adjusted with a greater strength. Therefore, the core of Eq. (5), i.e., $b_{j,r,s}$, is defined as:

$$b_{j,r,s} = \frac{1}{v_r\left(\sum_{t=s+1}^{v_r}\frac{\sigma_{o_{r,t}}(\mathbf{C}_j)}{t-s} + \sum_{g=1}^{s}\frac{\sigma_{o_{r,g}}(\mathbf{C}_j)}{s+1-g}\right)}, \quad (6)$$

where the denominator measures the *expected effectiveness* of shortening the distance between $o_{r,s}$ and $o_{r,s+1}$ for reducing Z. More specifically, $\sum_{t=s+1}^{v_r}\frac{\sigma_{o_{r,t}}(\mathbf{C}_j)}{t-s}$ measures the *expected effectiveness* of shortening $w_{r,s}$ by moving $o_{r,s}$ towards the categories with larger order values. Similarly, $\sum_{g=1}^{s}\frac{\sigma_{o_{r,g}}(\mathbf{C}_j)}{s+1-g}$ measures the *expected effectiveness* of shortening $w_{r,s}$ by moving $o_{r,s+1}$ towards the categories with smaller order values.

Remarks: Larger $\sigma_{o_{r,t}}(\mathbf{C}_j)$ results in larger $\sum_{t=s+1}^{v_r} \frac{\sigma_{o_{r,t}}(\mathbf{C}_j)}{t-s}$ in Eq. (6). This describes that moving $o_{r,s}$ towards more values in $\mathbf{C}_j$ is surely expected to achieve a better reduction on Z.

Remarks: Larger $\frac{1}{t-s}$ results in larger $\sum_{t=s+1}^{v_r} \frac{\sigma_{o_{r,t}}(\mathbf{C}_j)}{t-s}$ in Eq. (6). This describes that moving $o_{r,s}$ towards the values with similar order values to $o_{r,s}$ in $\mathbf{C}_j$ is expected to achieve a better reduction on Z. In contrast, two objects with dissimilar order values are more likely to be partitioned into different clusters, and shortening the distances between them is not surely expected to achieve a better reduction on Z.

Remarks: v_r in the denominator of Eq. (6) ensures that the distances from attributes with different numbers of categories are fairly updated. Otherwise, for a too large (small) v_r, $\mathbf{w}_r$ will be insufficiently (excessively) updated due to the large (small) order differences between categories.

We should shorten the distance between $o_{r,s}$ and $o_{r,s+1}$ (i.e., reducing $w_{r,s}$) according to the *expected effectiveness* to reduce Z. Therefore, reciprocal of the *expected effectiveness* is adopted by Eq. (6). To weight the contribution of $\mathbf{C}_j$ in deciding $\mathbf{w}_r$, $B_{j,r}$ in Eq. (5) is defined as:

$$B_{j,r} = \sum_{t=1}^{v_r-1} \frac{1}{b_{j,r,t}}, \tag{7}$$

which is the cluster level *expected effectiveness* for reducing Z. Specifically, a higher $B_{j,r}$ indicates that updating $\mathbf{w}_r$ according to $\mathbf{C}_j$ can more effectively reduce Z. To weight the importance of the whole $\mathbf{w}_r$, Φ_r in Eq. (5) is defined as:

$$\Phi_r = \sum_{h=1}^{k} B_{h,r}, \tag{8}$$

which is the total *expected effectiveness* of $\mathbf{w}_r$. Evidently, $\mathbf{w}_r$ with larger Φ_r should be assigned with a greater importance to better reduce Z. $B_{j,r}$ and Φ_r are both defined based on the *expected effectiveness*, and they have the consistent goal for reducing Z in Eq. (5). Therefore, integrating them in Eq. (5) is reasonable for our learning task. According to Eq. (8), we simplify Eq. (5) to:

$$w_{r,s} = \frac{\sum_{j=1}^{k} \frac{b_{j,r,s} \cdot B_{j,r}}{\sum_{t=1}^{v_r-1} b_{j,r,t}}}{\sum_{g=1}^{m} \Phi_g}. \tag{9}$$

The training process of DLC is shown in Algorithm 1. The most common way for treating ordinal data in clustering analysis is to assign consecutive integers to the ordinal categories according to their order, which is equivalent to assigning identical distance "1" to each pair of adjacent ordinal categories. Our initialization formula $w_{r,s} = \frac{1}{m(v_r-1)}$ just adopts this common way for the initialization. The denominator $m(v_r - 1)$ ensures that the sum of the initialized distances is equal to the sum of the learned ones. Fig. 2 shows a typical distance learning process of DLC on an ordinal dataset with four attributes. The black spots represent categories, the link between two black spots represents the distance between two categories, and the vertical axis represents the number of learning epochs. Distance between any pair of adjacent categories of an attribute is the same before the learning (Epoch 0). In the learning process, the data is partitioned according to the present distances, and then the distances are updated according to the present partition. After the learning in Epoch 1 and 2, the DLC algorithm converges, and the final distances are obtained. In practice, matrices $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k\}$ recording the occurrence frequencies of categories in each cluster, and matrices $\mathbf{L} = \{\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_m\}$ recording the inter-category distances of each attribute, can be maintained to save the computation cost of DLC. $\mathbf{F}$ and $\mathbf{L}$ are determined by $\mathbf{Q}$ and $\mathbf{W}$, respectively, and should be updated when $\mathbf{Q}$ and $\mathbf{W}$ are updated. By maintaining $\mathbf{F}$ and $\mathbf{L}$, results of Eq. (2) and (9) can be directly read off for saving computation cost.

Algorithm 1 DLC Algorithm

Input: Dataset $\mathbf{X}$, number of clusters k.
Step 1: Set the timestamp by $\tau = 0$; Initialize each row of $\mathbf{Q}^\tau$ by randomly setting one element at 1 and the remainders at 0; Initialize $\mathbf{W}^\tau$ by $w_{r,s} = \frac{1}{m(v_r-1)}$, $r = \{1, 2, ..., m\}$, $s = \{1, 2, ..., v_r - 1\}$;
Step 2: Fix $\mathbf{W}^\tau$, iteratively update $\mathbf{Q}^\tau$ according to Eq. (2) until all the values of $\mathbf{Q}^\tau$ remain unchanged. Then we obtain $\mathbf{Q}^{\tau+1}$. If $\mathbf{Q}^{\tau+1} \neq \mathbf{Q}^\tau$, go to **Step 3**; Otherwise, stop and **Output** $\mathbf{Q}^\tau$ and $\mathbf{W}^\tau$;
Step 3: Fix $\mathbf{Q}^{\tau+1}$, update $\mathbf{W}^\tau$ according to Eq. (9). Then we obtain $\mathbf{W}^{\tau+1}$. Set $\tau = \tau + 1$, and go to **Step 2**;
Output: $\mathbf{Q}^\tau$ and $\mathbf{W}^\tau$.

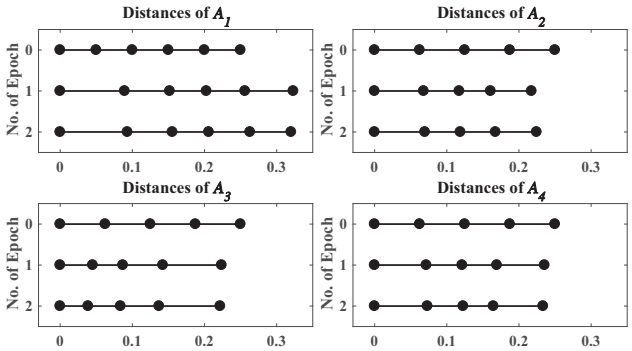

Figure 2: Typical Distance Learning Process of DLC.

Time Complexity: Time complexity for obtaining $\mathbf{Q}^{\tau+1}$ and updating $\mathbf{F}$ is $O(InmkV)$, where I is the number of iterations for updating $\mathbf{Q}$, and $V = max(v_1, v_2, ..., v_m)$ is only adopted to simplify the analysis because the attributes may have different numbers of categories. Time complexity for obtaining $\mathbf{W}^{\tau+1}$ and updating $\mathbf{L}$ is $O(mkV^2)$. Suppose **Step 3** of Algorithm 1 is repeated E times, the overall time complexity of DLC is $O(E(InmkV + mkV^2))$. Since I, E, and V are small constants ($I \times E \leq 20$ according to our experiments, $V \ll n$ and $V^2 < n$ for real datasets), the time complexity of DLC is $O(nmk)$, which is the same as that of the simplest clustering algorithms.

Space Complexity: During clustering, an $n \times m$ matrix $\mathbf{X}$, an $n \times k$ matrix $\mathbf{Q}$, k matrices $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k\}$,

each of which with size $m \times V$, m vectors $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m\}$, each of which with size V, and m matrices $\mathbf{L} = \{\mathbf{l}_1, \mathbf{l}_2, ...\mathbf{l}_m\}$, each of which with size $V \times V$, should be maintained. Since V is a small constant, the overall space complexity of DLC is $O(nm + nk + km)$.

Discussions: Several issues are discussed below:

- Overlapping Clusters: In general, the overlapping region has a relatively low density. According to Eq. (9), DLC will assign larger weights to the low-dense region. Hence, DLC has the ability to distinguish overlapping clusters.
- High Dimensional Data: Since DLC learns weights according to the full-space clusters, reasonableness of the learned weights may be influenced when processing high dimensional data composed of subspace clusters.
- Relation with Subspace Clustering: DLC learns weights by combining the contributions of all the clusters and uses the learned weights to partition the data in its full-space. Therefore, DLC is not a subspace clustering algorithm.

Experiments

To evaluate the performance of DLC, we compare it with 11 counterparts, including the state-of-the-art, representative, and conventional approaches, on 10 real datasets. Experimental settings are discussed below.

11 Counterparts: KMD-HDM (KMDH), KMD-CBDM (KMDC), KMD-JDM (KMDJ), KMD-EBDM (KMDE), WKMD-HDM (WKMDH), WKMD-CBDM (WKMDC), WKMD-EBDM (WKMDE), and the Categorical version of WOCIL (CWO) are chosen as Type-1 counterparts (i.e., approaches formed by combining clustering algorithms and distance metrics/measures proposed for categorical data). JDM is not combined with WKMD because they both have attribute weighting mechanisms that may conflict with each other during clustering. KMS, WKMS, and Numerical version of WOCIL (NWO) adopting Euclidean distance metric are chosen as Type-2 counterparts (i.e., approaches treat ordinal data as numerical one after coding).

10 Datasets: Statistics of the experimental datasets are shown in Table 1. CS, HR, CE, and NS datasets are collected from the UCI machine learning repository (Dua and Karra Taniskidou 2017). ES, LE, and SW datasets are collected from the Weka website (Witten et al. 2016). PE and AE datasets are collected from questionnaires of Shenzhen University. IS dataset is collected from questionnaires of the Education University of Hong Kong. Since we focus on ordinal data clustering, non-ordinal attributes in AE, CS, HR, CE, and NS datasets are omitted. Before performing clustering using the Type-2 counterparts, datasets are pre-processed by: 1) Coding ordinal categories as consecutive integers according to their ranking, and 2) Normalize the coded data using Z-score normalization.

Three Validity Indices: Clustering Accuracy (CA) (He, Cai, and Niyogi 2006), Adjusted Rand Index (ARI) (Rand 1971; Gates and Ahn 2017), and Normalized Mutual Information (NMI) (Strehl and Ghosh 2002), are chosen for evaluating the clustering performance. Values of CA, ARI, and NMI are in the intervals [0,1], [-1,1], and [0,1], respectively. Larger values of them indicate better performance.

Table 1: Statistics of the 10 Real Datasets.

Datasets	# Ins.	# Att.	# Class
Photo Evaluation (PE)	66	4	3
Assistant Evaluation (AE)	72	4	3
Caesarian Section (CS)	80	5	2
Internship Survey (IS)	90	3	2
Hayes-Roth (HR)	160	4	3
Employee Selection (ES)	488	4	9
Lecturer Evaluation (LE)	1,000	4	5
Social Works (SW)	1,000	10	4
Car Evaluation (CE)	1,728	6	4
Nursery School (NS)	12,960	8	4

Parameter Settings: The number of clusters k is set according to the labels of the datasets. For the WKMD- and WKMS-based approaches, the parameter β for updating the attribute weights is set at 2 (Chan et al. 2004; Huang et al. 2005). For JDM, the threshold for selecting attributes is calculated according to the suggested formula (Jia, Cheung, and Liu 2016).

Comparative Results[1]: We compare DLC with the Type-1 counterparts in Table 2 and Type-2 counterparts in Table 3. All the results are averaged by 10 runs of the experiments. The best and second-best results are indicated by boldface and underline, respectively. Significance test is conducted between the best and the second-best results by Wilcoxon signed rank test with 95% confidence interval, and significant difference is indicated by symbol "•". Performance of CBDM-based approaches is not reported on CE and NS datasets. It is because that the attributes of these two datasets are independent of each other and CBDM fails to measure distances for such datasets. **Observations: (1)** DLC performs the best or the second best on almost all the datasets, which illustrates the effectiveness of DLC for ordinal data clustering; **(2)** In comparison with the Type-1 counterparts, DLC achieves significantly better CA, ARI, and NMI performance on 5, 8, and 9 datasets, respectively. **(3)** In comparison with the Type-2 counterparts, DLC achieves significantly better CA, ARI, and NMI performance on 6, 6, and 5 datasets, respectively.

Effectiveness of the Core Components of DLC: We evaluate the effectiveness of the two core components of DLC, i.e., the defined Object-Cluster Distance (OCD) for solving $\boldsymbol{P}_1$, and the proposed Weights Updating (WU) scheme for solving $\boldsymbol{P}_2$. Effectiveness of OCD is evaluated by comparing DLC+OCD (i.e., the version of DLC that only performs Step 1-2 in Algorithm 1) with KMDH and KMS that adopt conventional object-mode and object-center distances, respectively. Effectiveness of WU is evaluated by comparing the complete version of DLC with DLC+OCD. Comparison results are shown in Figure 3-5. For simplicity, complete version of DLC and DLC+OCD are denoted as DLC and OCD, respectively. **Observations: (1)** By comparing OCD with KMDH and KMS, it can be found that OCD performs the best on most datasets, which illustrates the ef-

[1]More comparative results are available at https://drive.google.com/file/d/1tzYJ3a03hO4QDAQX9IvwyeiQMuetHhzl/view

Table 2: CA, ARI, and NMI Performance of DLC and Type-1 Counterparts.

Index	Data	KMDH	KMDC	KMDJ	KMDE	WKMDH	WKMDC	WKMDE	CWO	DLC
CA	PE	0.514±0.07	0.552±0.08	0.486±0.07	**0.626**±0.06	0.500±0.09	0.561±0.06	0.567±0.10	0.582±0.09	0.620±0.07
	AE	0.537±0.08	0.557±0.07	0.526±0.04	0.626±0.09	0.553±0.08	0.551±0.09	0.631±0.10	0.565±0.08	**0.676**±0.06 •
	CS	0.620±0.05	0.615±0.02	**0.630**±0.00	0.613±0.05	0.594±0.04	0.613±0.03	0.565±0.02	0.612±0.04	**0.630**±0.04
	IS	0.562±0.06	0.534±0.03	0.558±0.02	0.606±0.06	0.547±0.04	0.514±0.01	0.600±0.07	0.542±0.04	**0.678**±0.07 •
	HR	0.414±0.05	0.435±0.03	0.427±0.02	0.455±0.02	0.464±0.04	0.504±0.07	**0.529**±0.01	0.512±0.04	0.502±0.05
	ES	0.361±0.03	0.406±0.03	0.359±0.04	0.401±0.03	0.394±0.03	0.408±0.03	0.401±0.03	0.420±0.04	**0.455**±0.03 •
	LE	0.345±0.03	0.320±0.03	0.323±0.04	0.361±0.03	0.331±0.04	0.332±0.03	**0.367**±0.04	0.341±0.04	0.362±0.02
	SW	0.370±0.03	0.371±0.02	0.359±0.03	0.379±0.03	0.376±0.02	0.378±0.02	0.385±0.02	0.384±0.03	**0.418**±0.01 •
	CE	0.350±0.04	-	0.388±0.04	0.349±0.04	0.363±0.06	-	**0.418**±0.05	0.334±0.02	0.400±0.06
	NS	0.370±0.05	-	0.373±0.05	0.378±0.05	0.390±0.09	-	0.397±0.08	0.293±0.03	**0.444**±0.08 •
Index	Data	KMDH	KMDC	KMDJ	KMDE	WKMDH	WKMDC	WKMDE	CWO	DLC
ARI	PE	0.096±0.07	0.141±0.08	0.071±0.06	0.240±0.08	0.090±0.08	0.136±0.06	0.148±0.13	0.152±0.10	**0.253**±0.04
	AE	0.113±0.09	0.127±0.06	0.124±0.03	0.236±0.09	0.129±0.07	0.115±0.08	0.247±0.11	0.131±0.08	**0.294**±0.07 •
	CS	0.054±0.04	0.061±0.02	**0.068**±0.00	0.047±0.04	0.024±0.04	0.040±0.04	-0.002±0.00	0.044±0.03	0.061±0.03
	IS	0.004±0.05	-0.004±0.01	0.004±0.01	0.042±0.05	0.001±0.03	-0.014±0.00	0.042±0.06	-0.014±0.02	**0.125**±0.09 •
	HR	0.017±0.03	0.044±0.02	0.035±0.02	0.051±0.02	0.039±0.02	0.068±0.03	0.086±0.00	0.069±0.02	**0.106**±0.03 •
	ES	0.144±0.04	0.216±0.02	0.169±0.02	0.226±0.03	0.189±0.03	0.216±0.02	0.216±0.03	0.176±0.04	**0.261**±0.03 •
	LE	0.042±0.01	0.036±0.02	0.035±0.02	0.060±0.03	0.036±0.02	0.034±0.02	0.067±0.03	0.047±0.03	**0.083**±0.01 •
	SW	0.039±0.02	0.054±0.02	0.038±0.02	0.050±0.02	0.048±0.01	0.057±0.02	0.061±0.01	0.047±0.02	**0.098**±0.01 •
	CE	-0.005±0.01	-	0.040±0.03	0.028±0.03	0.008±0.01	-	0.023±0.02	0.014±0.00	**0.071**±0.05 •
	NS	0.050±0.03	-	0.069±0.04	0.066±0.04	0.084±0.10	-	0.098±0.09	0.003±0.00	**0.147**±0.07 •
Index	Data	KMDH	KMDC	KMDJ	KMDE	WKMDH	WKMDC	WKMDE	CWO	DLC
NMI	PE	0.126±0.06	0.170±0.07	0.099±0.06	0.279±0.08	0.136±0.10	0.170±0.08	0.195±0.13	0.218±0.12	**0.334**±0.03 •
	AE	0.156±0.07	0.150±0.08	0.142±0.04	0.260±0.09	0.153±0.08	0.141±0.10	0.269±0.09	0.181±0.10	**0.327**±0.04 •
	CS	0.076±0.04	0.064±0.03	**0.086**±0.02	0.074±0.03	0.031±0.04	0.035±0.03	0.009±0.01	0.053±0.04	0.085±0.03
	IS	0.015±0.02	0.008±0.01	0.014±0.01	0.039±0.04	0.013±0.01	0.004±0.00	0.041±0.06	0.020±0.01	**0.097**±0.06 •
	HR	0.041±0.04	0.056±0.03	0.054±0.02	0.066±0.03	0.055±0.04	0.071±0.02	0.085±0.00	0.072±0.02	**0.135**±0.05 •
	ES	0.276±0.04	0.350±0.02	0.288±0.02	0.381±0.01	0.317±0.03	0.360±0.01	0.370±0.03	0.299±0.04	**0.417**±0.02 •
	LE	0.064±0.02	0.068±0.02	0.063±0.03	0.081±0.04	0.061±0.03	0.065±0.03	0.094±0.04	0.070±0.04	**0.135**±0.02 •
	SW	0.061±0.02	0.077±0.02	0.056±0.03	0.077±0.02	0.067±0.02	0.081±0.02	0.095±0.01	0.074±0.02	**0.129**±0.01 •
	CE	0.042±0.02	-	0.076±0.04	0.069±0.03	0.020±0.02	-	0.045±0.03	0.049±0.02	**0.149**±0.04 •
	NS	0.053±0.03	-	0.073±0.04	0.074±0.04	0.096±0.17	-	0.116±0.08	0.007±0.00	**0.182**±0.09 •

Table 3: CA, ARI, and NMI Performance of DLC and Type-2 Counterparts.

Index	Data	KMS	WKMS	NWO	DLC
CA	PE	0.583±0.06	0.580±0.06	0.602±0.07	**0.620**±0.07 •
	AE	0.606±0.03	0.601±0.03	0.608±0.04	**0.676**±0.06 •
	CS	0.591±0.05	0.610±0.05	0.600±0.05	**0.630**±0.04 •
	IS	0.643±0.07	0.639±0.09	0.639±0.09	**0.678**±0.07 •
	HR	0.497±0.05	0.496±0.05	0.501±0.05	**0.502**±0.05
	ES	0.431±0.03	0.428±0.04	0.438±0.03	**0.455**±0.03
	LE	0.352±0.01	0.347±0.01	0.355±0.01	**0.362**±0.02
	SW	**0.419**±0.03	0.402±0.03	0.403±0.03	0.418±0.01
	CE	0.332±0.02	0.342±0.02	0.363±0.03	**0.400**±0.06 •
	NS	0.389±0.06	0.381±0.07	0.380±0.06	**0.444**±0.08 •
Index	Data	KMS	WKMS	NWO	DLC
ARI	PE	0.231±0.03	0.216±0.06	**0.253**±0.07	**0.253**±0.04
	AE	0.255±0.05	0.255±0.05	0.264±0.07	**0.294**±0.07 •
	CS	0.028±0.04	0.045±0.04	0.037±0.04	**0.061**±0.03 •
	IS	0.083±0.08	0.090±0.09	0.090±0.09	**0.125**±0.09 •
	HR	0.087±0.03	0.087±0.03	0.090±0.03	**0.106**±0.03 •
	ES	0.259±0.02	0.249±0.03	**0.261**±0.03	**0.261**±0.03
	LE	**0.085**±0.01	0.075±0.02	0.074±0.01	0.083±0.01
	SW	0.096±0.01	0.072±0.03	0.073±0.02	**0.098**±0.01
	CE	0.027±0.02	0.031±0.03	0.038±0.05	**0.071**±0.05 •
	NS	0.096±0.08	0.095±0.08	0.086±0.08	**0.147**±0.07 •
Index	Data	KMS	WKMS	NWO	DLC
NMI	PE	0.322±0.04	0.302±0.05	**0.339**±0.06	0.334±0.03
	AE	0.323±0.04	0.325±0.05	**0.327**±0.05	**0.327**±0.04
	CS	0.049±0.03	0.071±0.04	0.061±0.04	**0.085**±0.03 •
	IS	0.076±0.08	0.075±0.08	0.075±0.08	**0.097**±0.06 •
	HR	0.102±0.03	0.102±0.03	0.105±0.02	**0.135**±0.05 •
	ES	**0.432**±0.01	0.425±0.02	0.431±0.02	0.417±0.02
	LE	**0.141**±0.01	0.125±0.02	0.128±0.01	0.135±0.02
	SW	0.122±0.01	0.097±0.04	0.099±0.02	**0.129**±0.01
	CE	0.083±0.04	0.086±0.05	0.109±0.07	**0.149**±0.04 •
	NS	0.121±0.10	0.122±0.10	0.113±0.10	**0.182**±0.09 •

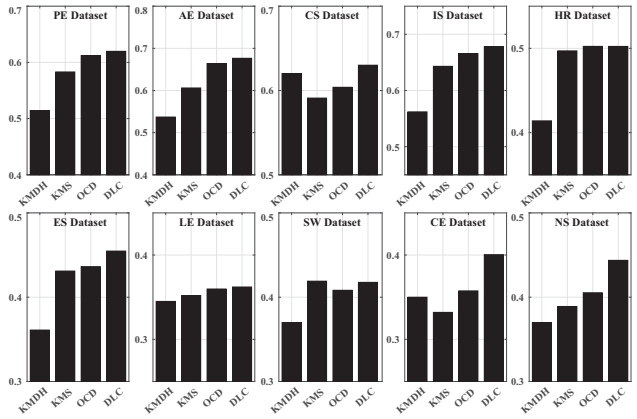

Figure 3: Average CA of KMDH, KMS, OCD, and DLC.

fectiveness of OCD. **(2)** By comparing DLC with OCD, it can be found that DLC obviously outperforms OCD on most datasets, which illustrates the effectiveness of WU. **(3)** The performance of OCD and DLC is almost the same on HR dataset. By checking the final $\mathbf{W}$ obtained by DLC, we find that the obtained $\mathbf{W}$ is very close to the initialized one. This may be because that the 'true' $\mathbf{W}$ of HR dataset is very similar to the initialized one. Therefore, WU cannot learn a better $\mathbf{W}$ in this case. **(4)** In general, this experiment adequately illustrates the effectiveness of the two core components of DLC.

Convergence Study: To evaluate the convergence of DLC, we run it on all the 10 datasets and record the conver-

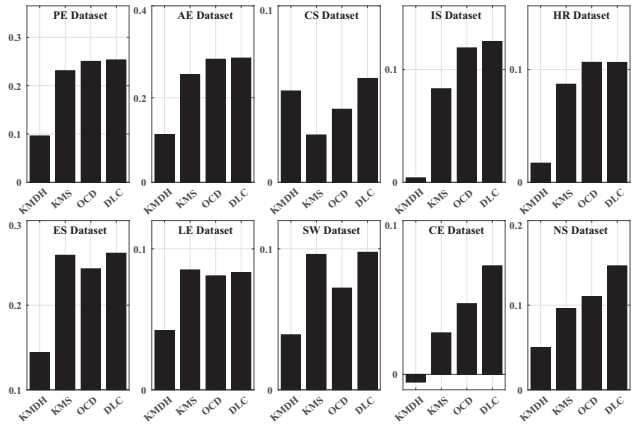

Figure 4: Average ARI of KMDH, KMS, OCD, and DLC.

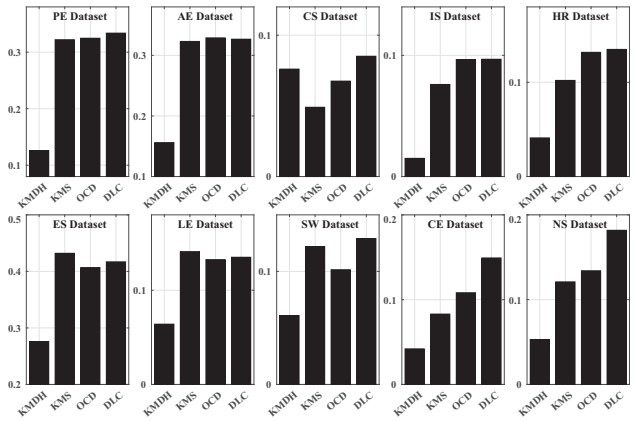

Figure 5: Average NMI of KMDH, KMS, OCD, and DLC.

gence curves in Fig. 6. The horizontal axis and vertical axis represent the total number of iterations and the value of the objective function, respectively. Triangles indicate the start and end iterations of the learning. Circles indicate the iterations that $\mathbf{W}$ has been updated. **Observations: (1)** After the updating of $\mathbf{W}$, Z (i.e., the value of the objective function) is obviously reduced, which indicates the effectiveness of DLC. **(2)** DLC spends less than 20 iterations to converge on all the datasets except on ES. This is because each attribute of ES has around 10 categories, which leads to the updating of more weights. Even though, DLC still converges using not too many iterations (31 in total).

Efficiency Evaluation: We evaluate the efficiency of DLC by comparing its execution time with that of KMDH, WKMDH, CWO, KMS, WKMS, and NWO on different-sized NS datasets (generated by randomly selecting 10%, 20%, ..., 100% instances from the NS dataset) in Figure 7. Since the computation cost of CBDM, JDM, and EBDM is similar to or higher than the others, execution time of the approaches based on them are not reported, which does not influence the efficiency validation. In general, DLC spends a little more time than the simplest KMDH and KMS, and has almost the same execution time as WKMS and NWO.

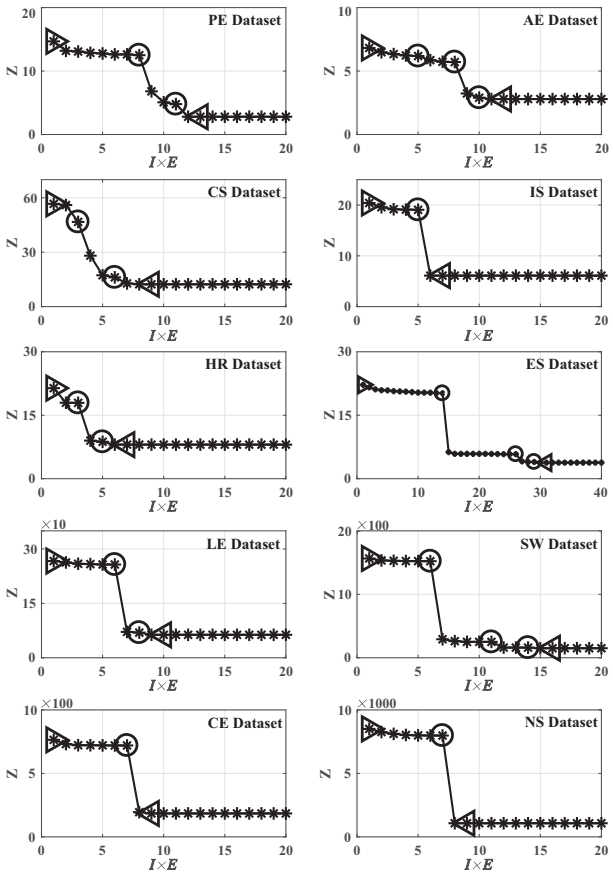

Figure 6: Convergence Curves of DLC.

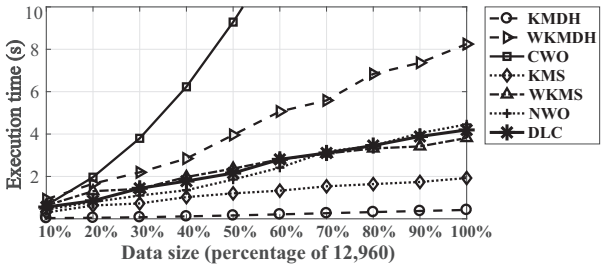

Figure 7: Execution Time on NS Dataset.

Moreover, the increasing rate of DLC's execution time is almost linear over data size, which is consistent with our time complexity analysis. In conclusion, DLC is efficient in comparison with the state-of-the-art approaches, and it will not bring much additional computation cost than the simplest ones.

Conclusion

In this paper, we have proposed an object-cluster distance measure, which finely quantifies the distance between object and cluster by exploiting the order relationship, for ordinal data. Based on this measure, we have then developed a novel DLC algorithm for ordinal data clustering by integrat-

ing the data partitioning with inter-category distance learning. Compared with the existing counterparts, DLC learns more reasonable distances by dynamically adjusting the distances according to the partitions of the dataset. DLC features superior clustering accuracy, low time complexity, fast convergence, and is parameter-free. Extensive experiments on different real and benchmark datasets demonstrate the efficacy of DLC.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grants: 61672444 and 61272366, Hong Kong Baptist University (HKBU), Research Committee, Initiation Grant - Faculty Niche Research Areas (IG-FNRA) 2018/19 with grant code: RC-FNRA-IG/18-19/SCI/03, the Innovation and Technology Fund of Innovation and Technology Commission of the Government of the Hong Kong SAR with project code: ITS/339/18, the Faculty Research Grant of HKBU under Project FRG2/17-18/082, and the SZSTI under Grant JCYJ20160531194006833.

References

Agresti, A. 1996. *An Introduction to Categorical Data Analysis*. Wiley.

Ahmad, A., and Dey, L. 2007. A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set. *Pattern Recognition Letters* 28(1):110–118.

Allen, I., and Seaman, C. 2007. Likert scales and data analyses. *Quality progress* 40(7):64–65.

Barbará, D.; Li, Y.; and Couto, J. 2002. Coolcat: An entropy-based algorithm for categorical clustering. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, 582–589. ACM.

Chan, E. Y.; Ching, W. K.; Ng, M. K.; and Huang, J. Z. 2004. An optimization algorithm for clustering using weighted dissimilarity measures. *Pattern Recognition* 37(5):943–952.

Cheung, Y.-m., and Jia, H. 2013. Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number. *Pattern Recognition* 46(8):2228–2238.

Dua, D., and Karra Taniskidou, E. 2017. UCI machine learning repository.

Gates, A. J., and Ahn, Y.-Y. 2017. The impact of random models on clustering similarity. *The Journal of Machine Learning Research* 18(1):3049–3076.

He, X.; Cai, D.; and Niyogi, P. 2006. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems*, 507–514.

Huang, J. Z.; Ng, M. K.; Rong, H.; and Li, Z. 2005. Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(5):657–668.

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

Ienco, D.; Pensa, R. G.; and Meo, R. 2012. From context to distance: Learning dissimilarity for categorical data clustering. *ACM Transactions on Knowledge Discovery from Data* 6(1):1–22.

Jia, H., and Cheung, Y.-m. 2018. Subspace clustering of categorical and numerical data with an unknown number of clusters. *IEEE Transactions on Neural Networks and Learning Systems* 29(8):3308–3325.

Jia, H.; Cheung, Y.-m.; and Liu, J. 2016. A new distance metric for unsupervised learning of categorical data. *IEEE Transactions on Neural Networks and Learning Systems* 27(5):1065–1079.

Le, S. Q., and Ho, T. B. 2005. An association-based dissimilarity measure for categorical data. *Pattern Recognition Letters* 26(16):2549–2557.

Li, T.; Ma, S.; and Ogihara, M. 2004. Entropy-based criterion in categorical clustering. In *Proceedings of the Twenty-First International Conference on Machine Learning*, 68–75. ACM.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297. University of California.

Ng, M. K.; Li, M. J.; Huang, J. Z.; and He, Z. 2007. On the impact of dissimilarity measure in k-modes clustering algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(3):503–507.

Rand, W. M. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* 66(336):846–850.

Strehl, A., and Ghosh, J. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3:583–617.

Witten, I. H.; Frank, E.; Hall, M. A.; and Pal, C. J. 2016. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Zhang, Y., and Cheung, Y.-m. 2018. Exploiting order information embedded in ordered categories for ordinal data clustering. In *Proceedings of the 24th International Symposium on Methodologies for Intelligent Systems*, 247–257. Springer.