Learning Distance Structure for Ordinal Data Clustering

Anonymous Author(s)
Submission Id: 1053

ABSTRACT

Ordinal data clustering is common in data mining and machine learning tasks. As a major type of categorical data, ordinal data is composed of attributes with naturally possible values (also called categories interchangeably). However, due to the lack of dedicated metric and clustering algorithm, ordinal categories are usually treated as nominal ones, or coded as consecutive integers and treated as numerical ones. Both these two common ways may twist the intrinsic distances between ordinal categories, and will thus produce incorrect clustering results. This paper proposes a novel ordinal data clustering algorithm, which iteratively learns 1) the distances between categories, 2) the weights of attributes, and 3) the optimal partition of ordinal data. To the best of our knowledge, this is the first time that an ordinal data clustering algorithm is proposed. The proposed algorithm features superior clustering accuracy, low time complexity, and fast convergence. Extensive Experiments show its efficacy.

KEYWORDS

Categorical Data; Ordinal Data; Distance Structure; Order Relationship; Clustering Analysis; Iterative Learning

1 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 [2], 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/similarity measurement plays a main role [10]. Since the values of ordinal data are not quantitative, the distances of ordinal data are not well-defined in general. Therefore, ordinal data is usually treated in either of the following two ways in clustering analysis: 1) Treat ordinal data as nominal one; 2) Code ordinal categories as consecutive integers, and treat the coded data as numerical one.

In the former way, all the metrics proposed for categorical data are applicable for ordinal data distance measurement. These metrics include Hamming Distance Metric (HDM), Ahmad's Distance Metric (ADM) [1], Association-Based Distance Metric (ABDM) [9], Context-Based Distance Metric (CBDM) [6], Jia's Distance Metric (JDM) [8], Object-Cluster Similarity Measure (OCSM) [3]. Among them, HDM is the simplest and most popular one. However, its distance is simply binary, i.e., the distance value is zero if two categories are identical, otherwise one. Evidently, it is too simple to suitable for ordinal data. By contrast, the remaining five state-of-the-art ones attempt to more reasonably define the distances by exploiting statistic information of categories. However, they are actually proposed for nominal data, and the distances defined by them may violate the natural order relationship among ordered categories. Most recently, an Entropy-Based Distance Metric (EBDM) [12] has been proposed for ordinal data distance measurement. It adopts cumulative entropy as a measure to simultaneously take the statistic information and order relationship into account for defining the distances between ordered categories. 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, existing categorical data clustering algorithms, including the conventional K-Modes (KMD), the representative Weighted K-Modes (WKMD) [5], the state-of-the-art Weighted OCIL (WOC)[1] [7], will produce unsatisfactory clustering results due to unreasonably defined distances.

In the latter way, all the categories within the same attribute are usually coded by consecutive integers, for example, ordinal categories {accept, neutral, reject} are coded as $\{3, 2, 1\}$. In this way, ordinal data is converted into numerical one with preserving the order relationship among categories. Existing numerical data clustering algorithms, including the conventional K-Means (KMS), the representative Weighted K-Means (WKMS) [5], and the state-of-the-art WOC[1] [7], are applicable for clustering the coded ordinal data. However, since the statistic information of categories is lost after coding, and the identical distance is assigned to different pairs of adjacent categories that may have intrinsically unequal distances, the intrinsic distances of ordinal data are usually twisted, which may lead to unsatisfactory clustering results.

In this paper, we therefore propose a Distance Structure Learning-based Clustering (DSLC) algorithm, which iteratively adjusts the distances between categories to search for a better partition of ordinal data. The basic idea for the DSLC algorithm is to adjust the distances between categories according to their contributions in forming compact clusters. However, a difficulty lies in how to efficiently describe and adjust the distances between each pair of the categories with preserving their order relationship for each attribute. To tackle this, we maintain a unique distance structure for each attribute to efficiently describe the distances between categories and their order relationship. Specifically, we only use the distances between adjacent categories, called basic distances, to describe the distance structure of each attribute. The distances between non-adjacent categories are jointly described by the basic distances according to the order of the categories. In this way, for an attribute A_r with v_r categories, only $v_r - 1$ basic distances are needed to indicate the distances of all the $\frac{v_r(v_r-1)}{2}$ pairs of categories with preserving their order relationship. Thus, this design makes the

[1] WOC belongs to both of the two ways because it is applicable for both nominal and numerical data.

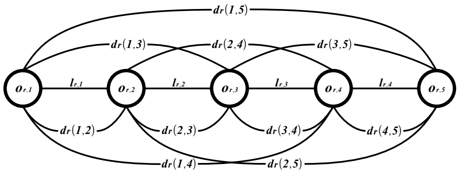


Figure 1: Distance structure of A_r when $v_r = 5$.

nontrivial ordinal data distance learning problem achievable. The main contributions of this paper are three-fold: 1) an ordinal data clustering algorithm DSLC is proposed, 2) an efficient distance structure learning scheme is designed for the distance learning with preserving the order relationship, and 3) a learning process control scheme is provided to ensure fast convergence of DSLC. Extensive experiments on eight real and benchmark datasets show the efficacy of DSLC.

2 PROPOSED METHOD

In this section, we first formalize the problem, and then introduce the proposed DSLC and analyze its time complexity.

2.1 Problem Description

Basic notations. 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. The m attributes may have different numbers of categories, for example A_r has v_r categories that are ordered as $o_{r,1} \prec o_{r,2} \prec ... \prec o_{r,v_r}$, where "$\prec$" indicates that the categories on its right ranked higher (have larger order values) than the categories on its left. Each category (e.g., $o_{r,t}$) has a unique subscript, which indicates that $o_{r,t}$ belongs to A_r, and has order value t among the categories of A_r.

Distance structure. For each attribute, there is a distance structure indicating the distances and the order relationship between its categories. We adopt a set of m vectors $\mathbf{L} = \{\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_m\}$ to describe the distance structures. Specifically, $v_r - 1$ basic distances (i.e., the distances between adjacent categories) $\mathbf{l}_r = \{l_{r,1}, l_{r,2}, ..., l_{r,v_r-1}\}$ describe the distances and order relationship of the $\frac{v_r(v_r-1)}{2}$ pairs of categories of A_r as shown in Fig. 1, and the distance between two categories $o_{r,g}$ and $o_{r,h}$ of A_r is defined as

$$d_r(g,h) = \begin{cases} \sum_{s=min(g,h)}^{max(g,h)-1} l_{r,s} & , \ if\ g \neq h \\ 0 & , \ if\ g = h, \end{cases} \tag{1}$$

which satisfy $d_r(s,t) \leq d_r(g,h)$, 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, the distance between two categories will not be larger than the distance between another two categories that are not ordered between them, which is consistent with the order relationship among the categories of an attribute.

Objective function. Suppose $\mathbf{X}$ is composed of k disjoint object clusters $\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_k$, our goal is to minimize the objective function Z with variables $\mathbf{Q}$ and $\mathbf{L}$:

$$Z(\mathbf{Q}, \mathbf{L}) = \sum_{i=1}^{n} \sum_{j=1}^{k} \sum_{r=1}^{m} q_{i,j} \cdot D_r(\mathbf{x}_i, \mathbf{C}_j), \tag{2}$$

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 $\sum_{j=1}^{k} q_{i,j} = 1$ with $q_{i,j} \in \{0, 1\}$, $i \in \{1, 2, ..., n\}$, $j \in \{1, 2, ..., k\}$. $\mathbf{L}$ is a set of basic distances defined in Section 2.1. $D_r(\mathbf{x}_i, \mathbf{C}_j) = \sum_{s=1}^{v_r} d_r(\kappa(x_{i,r}), s) \cdot u_{j,r,s}$ is the distance between $\mathbf{x}_i$ and $\mathbf{C}_j$ measured according to their values on A_r. $d_r(\cdot, \cdot)$ is defined by Eq. (1). $x_{i,r}$ is the value of $\mathbf{x}_i$ on A_r. The operation $\kappa(x_{i,r})$ fetches the order value of $x_{i,r}$, for example, if $x_{i,r} = o_{r,t}$, $\kappa(x_{i,r}) = t$. $u_{j,r,s} = \frac{\sigma_{o_{r,s}}(\mathbf{C}_j)}{\sigma(\mathbf{C}_j)}$ is the occurrence probability of $o_{r,s}$ in $\mathbf{C}_j$, where $\sigma(\mathbf{C}_j)$ and $\sigma_{o_{r,s}}(\mathbf{C}_j)$ count the number of objects in $\mathbf{C}_j$, and the number of objects with their values on A_r equal to $o_{r,s}$ in $\mathbf{C}_j$, respectively.

2.2 DSLC Algorithm

DSLC minimizes Z by iteratively solving two problems: P_1: Fix $\mathbf{L}$, minimize Z by adjusting $\mathbf{Q}$; and P_2: Fix $\mathbf{Q}$, reduce Z by adjusting $\mathbf{L}$. Solving procedures are discussed below.

Initialization. For each row of $\mathbf{Q}$, we randomly set one element to 1 and the remainders to 0. In this way, the n objects are randomly assigned into k clusters. We set the values of each vector of $\mathbf{L}$ (e.g., $\mathbf{l}_r$, $r \in \{1, 2, ..., m\}$) to a set of identical values $\{\frac{1}{m(v_r-1)}, \frac{1}{m(v_r-1)}, ..., \frac{1}{m(v_r-1)}\}$ to ensure that the object-cluster distance $\sum_{r=1}^{m} D_r(\mathbf{x}_i, \mathbf{C}_j)$ is in the interval [0,1]. Then, P_1 is solved as follows.

Solve P_1. P_1 is solved by

$$q_{i,j} = \begin{cases} 1 & , \ if\ j = \arg\min_y \sum_{r=1}^{m} D_r(\mathbf{x}_i, \mathbf{C}_y) \\ 0 & , \ else, \end{cases} \tag{3}$$

$i \in \{1, 2, ..., n\}$ and $j \in \{1, 2, ..., k\}$. All the n objects are assigned to their closest clusters according to Eq. (3) until the values of $\mathbf{Q}$ no longer change. Then we solve P_2 as follows.

Solve P_2. P_2 is solved by adjusting the distance structures described by $\mathbf{L}$ to make the clusters more compact. More specifically, if adjusting a basic distance in $\mathbf{L}$ can decrease the total distance among the objects in a cluster by more, this basic distance should be adjusted with a greater force to achieve a better reduction of Z. Thus, we define the utility of adjusting a basic distance $l_{r,s}$ as $B_{r,s} = \sum_{j=1}^{k}(b_{j,r,s}^{\prec} + b_{j,r,s+1}^{\succ})$, where $b_{j,r,s}^{\prec} = \sum_{t=s+1}^{v_r} d_r(s,t) \cdot \sigma_{o_{r,t}}(\mathbf{C}_j)$ measures the total distance between $o_{r,s}$ and the values ranked higher than it in $\mathbf{C}_j$. Obviously, a larger $b_{j,r,s}^{\prec}$ indicates that moving $o_{r,s}$ towards $o_{r,s+1}$ can better decrease the total distance among the objects in $\mathbf{C}_j$. Similarly, the value of $b_{j,r,s+1}^{\succ} = \sum_{t=1}^{s} d_r(s+1,t) \cdot \sigma_{o_{r,t}}(\mathbf{C}_j)$ indicates the utility of moving $o_{r,s+1}$ towards $o_{r,s}$ in terms of $\mathbf{C}_j$. Therefore, the larger the utility $B_{r,s}$ is, the more the corresponding $l_{r,s}$ should be shortened to move $o_{r,s}$ and $o_{r,s+1}$ toward each other. Accordingly, each basic distance (e.g., $l_{r,s}$, $r \in \{1, 2, ..., m\}$, $s \in \{1, 2, ..., v_r - 1\}$) is updated to $\frac{\frac{1}{B_{r,s}}}{m \sum_{h=1}^{v_r-1} \frac{1}{B_{r,h}}}$ where the denominator $m\sum_{h=1}^{v_r-1}\frac{1}{B_{r,h}}$ is adopted to ensure that the magnitude of the updated distances is the same as the initialized ones. To remove the restriction that all the attributes are of equal importance, we further weight each attribute (e.g., A_r, $r \in \{1, 2, ..., m\}$) by $W_r = 1/\sum_{i=1}^{n}\sum_{j=1}^{k} q_{i,j} \cdot D_r(\mathbf{x}_i, \mathbf{C}_j)$, because an attribute causing larger total object-cluster distance is usually less important in forming compact clusters. Since an attribute weight W_r actually weights the importance of the whole distance structure of A_r, we concatenate W_r to the updating of $\mathbf{L}$ by

$$l_{r,s} = \frac{\frac{1}{B_{r,s}}}{\sum_{h=1}^{v_r-1}\frac{1}{B_{r,h}}} \cdot \frac{W_r}{\sum_{g=1}^{m} W_g}, \tag{4}$$

where $\frac{1}{m}$ of the original $\frac{\frac{1}{B_{r,s}}}{m\sum_{t=1}^{v_r-1}\frac{1}{B_{r,t}}}$ is replaced by $\frac{W_r}{\sum_{g=1}^{m} W_g}$. According to Eq. (4), $\mathbf{L}$ is updated to reduced Z.

Learning process control. The values of $\mathbf{L}$ are dominated by the present learning iteration. As a result, $\mathbf{L}$ changes too fast and DSLC will spend more iterations to converge. To avoid this, a common practice is to set a maximum number of learning iterations. We also provide a better option by controlling the learning process using a parameter $\omega \in [0, 1]$:

$$\mathbf{L}^{\tau+1} = \omega \cdot \mathbf{L}^{\tau+1} + (1 - \omega) \cdot \mathbf{L}^{\tau}, \tag{5}$$

where τ is a timestamp. In this way, a part of the previously learned information is preserved for more stable updating.

DSLC Algorithm *is summarized as follows.*
Input: $\mathbf{X}$, k, and ω.
Step 1: Set the timestamp by $\tau = 0$, initialize $\mathbf{Q}^{\tau}$ and $\mathbf{L}^{\tau}$ according to the 2nd paragraph of this Section;
Step 2: Fix $\mathbf{L}^{\tau}$, iteratively update $\mathbf{Q}$ according to Eq. (3) until $\mathbf{Q}$ remains unchanged, 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 $\mathbf{L}^{\tau}$.
Step 3: Fix $\mathbf{Q}^{\tau+1}$, update $\mathbf{L}$ according to Eq. (4) and (5), obtain $\mathbf{L}^{\tau+1}$. Set $\tau = \tau + 1$, and go to Step 2;
Output: $\mathbf{Q}^{\tau}$ and $\mathbf{L}^{\tau}$.
A set of matrices $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k\}$ recording the occurrence frequencies of categories in each cluster, and a set of matrices $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_m\}$ recording the distances between categories of each attribute, can be maintained to save computation cost of DSLC. $\mathbf{F}$ and $\mathbf{D}$ are determined by $\mathbf{Q}$ and $\mathbf{L}$, respectively, and should be dynamically updated. According to $\mathbf{F}$ and $\mathbf{D}$, results of Eq. (3) and (4) can be directly read off in Step 2 and 3 without laborious computation.

Time complexity *of DSLC is analyzed as follows.* Time complexity for obtaining $\mathbf{Q}^{\tau+1}$ and updating $\mathbf{F}$ in Step 2 is $O(InmkV)$, where I is the number of iterations for updating $\mathbf{Q}$, and $V = max(v_1, v_2, ..., v_m)$ is adopted for the analysis here because each attribute may have different number of categories. In Step 3, time complexity for obtaining $\mathbf{L}^{\tau+1}$ and updating $\mathbf{D}$ is $O(mkV^2)$. Suppose Step 2 and 3 are repeated E times, the time complexity of DSLC is $O(E(InmkV + mkV^2))$. Since both I and E are very small constants ($I \times E \leq 20$ according to the experiments), and V is also a small constant ($V \ll n$ and $V^2 < n$ for real ordinal datasets), the time complexity of DSLC is $O(nmk)$, which is the same as the time complexity of the simplest KMD and KMS algorithms.

3 EXPERIMENTS

3.1 Experimental Settings

10 counterparts. KMD, KMD-CBDM (KMDC), KMD-JDM (KMDJ), KMD-EBDM (KMDE), WKMD, WKMD-CBDM (WKMDC), and WKMD-EBDM (WKMDE) are chosen as representative counterparts that treat ordinal data as nominal one (called Type-1 counterparts). JDM is not combined with WKMD because their attribute weighting mechanisms conflicts with each other. KMS and WKMS are chosen as the counterparts that treat ordinal data as numerical one (called Type-2 counterparts). WOC applicable for both nominal and numerical data is also chosen as a counterpart. We set $\beta = 7$ according to [5] for WKMD and WKMS, and $\omega = 0.5$ for DSLC (ω is evaluated in Section 3.3).

Eight ordinal datasets. Internship Questionnaire (IQ)[2], Photo Evaluation (PE)[3], Assistant Evaluation (AE)[3], Primary Tumor (PT)[4], Breast Cancer (BC)[4], Car Evaluation (CE)[4], Nursery School (NS)[4], and Lecturer Evaluation (LE)[5]) are collected, and their statistics are: IQ: $(90, 3, 2)$, PE: $(66, 4, 3)$, AE: $(72, 4, 3)$, PT: $(133, 17, 11)$, BC: $(286, 9, 2)$, CE: $(1728, 6, 4)$, NS: $(12960, 8, 4)$, and LE: $(1000, 4, 5)$, where the 1st, 2nd, and 3rd values of each triplet indicate numbers of instances, attributes, and classes, respectively. Attribute values of the datasets should be coded as consecutive integers and processed by Z-score normalization for the Type-2 counterparts.

Two validity indices. The popular Adjusted Rand Index (ARI) [4], and Normalized Mutual Information (NMI) [11], are chosen. Values of ARI and NMI are in the intervals [-1,1] and [0,1], respectively. Larger ARI and NMI values indicate better clustering performance.

3.2 Comparative Results

We compare DSLC with the Type-1 and Type-2 counterparts in Table 1 and 2, respectively. All these results are averaged by 10 runs of the experiments. The best and second-best results are indicated by boldface and underline, respectively. The "↑" columns report the improvements achieved by DSLC in comparison with the second-best results. CBDM fails to measure distances for CE and NS, because they are composed of independence attributes. Thus, performance of KMDC and WKMDC is not reported on CE and NS. It can be observed that DSLC obviously outperforms all the 10 Type-1 and Type-2 counterparts on most of the datasets, which indicates that DSLC is more effective for ordinal data clustering.

3.3 Parameter & Convergence Evaluation

To evaluate ω, we run DSLC 200 times for different ω, and record the results on all the datasets in Fig. 2(a). To evaluate

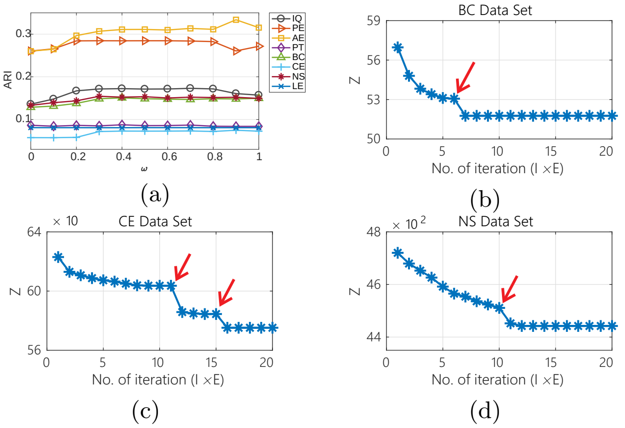
[2] Questionnaires collected from Education University of Hong Kong
[3] Questionnaires collected from Shenzhen University
[4] Collected from http://archive.ics.uci.edu/ml/datasets.php
[5] Collected from https://www.cs.waikato.ac.nz/ml/weka/datasets.html

Table 1: DSLC vs. Type-1 counterparts.

Index	ARI										NMI									
Data	KMD	KMDC	KMDJ	KMDE	WKMD	WKMDC	WKMDE	WOC	DSLC	↑	KMD	KMDC	KMDJ	KMDE	WKMD	WKMDC	WKMDE	WOC	DSLC	↑
IQ	-0.01	-0.01	0.007	0.044	-0.01	-0.02	0.072	-0.02	**0.171**	138%	0.012	0.004	0.014	0.046	0.018	0.003	0.069	0.023	**0.115**	67%
PE	0.105	0.135	0.069	0.222	0.091	0.131	0.166	0.132	**0.285**	28%	0.138	0.166	0.095	0.263	0.133	0.176	0.211	0.194	**0.344**	31%
AE	0.140	0.147	0.115	0.270	0.144	0.139	0.279	0.118	**0.311**	11%	0.173	0.176	0.124	0.304	0.175	0.171	0.310	0.161	**0.371**	20%
PT	0.073	0.068	0.074	0.078	0.055	0.063	0.045	0.084	**0.086**	2%	0.193	0.178	0.184	0.192	0.147	0.159	0.124	0.211	**0.216**	2%
BC	0.015	0.103	0.095	0.042	0.043	0.103	0.057	0.127	**0.149**	17%	0.014	0.051	0.051	0.032	0.025	0.050	0.037	0.061	**0.080**	31%
CE	-0.01	–	0.037	0.031	0.010	–	0.029	0.013	**0.072**	95%	0.043	–	0.075	0.078	0.023	–	0.067	0.051	**0.121**	55%
NS	0.054	–	0.074	0.075	0.084	–	0.082	0.003	**0.150**	79%	0.057	–	0.077	0.080	0.115	–	0.106	0.006	**0.196**	70%
LE	0.039	0.034	0.040	0.069	0.038	0.031	0.072	0.050	**0.081**	13%	0.065	0.064	0.075	0.096	0.066	0.058	0.099	0.073	**0.137**	38%

Table 2: DSLC vs. Type-2 counterparts.

Index	ARI					NMI				
Data	KMS	WKMS	WOC	DSLC	↑	KMS	WKMS	WOC	DSLC	↑
IQ	0.090	0.102	0.102	**0.171**	68%	0.077	0.081	0.081	**0.115**	42%
PE	0.248	0.225	0.248	**0.285**	15%	0.334	0.310	0.336	**0.344**	2%
AE	0.229	0.234	0.251	**0.311**	24%	0.309	0.316	0.329	**0.371**	13%
PT	0.084	0.046	0.084	**0.086**	2%	0.212	0.155	0.209	**0.216**	2%
BC	0.118	0.133	0.136	**0.149**	10%	0.069	0.072	0.069	**0.080**	11%
CE	0.030	0.024	0.019	**0.072**	140%	0.085	0.083	0.099	**0.121**	22%
NS	0.110	0.111	0.127	**0.150**	18%	0.133	0.138	0.159	**0.196**	23%
LE	0.077	0.074	0.067	**0.081**	5%	0.132	0.127	0.119	**0.137**	4%



(a) (b) (c) (d)

Figure 2: ARI - ω curves and Convergence curves.

the convergence, we run DSLC on BC, CE, and NS, and record the results in Fig. 2(b)-(d) (Results on the other datasets are omitted due to space limitation). We have three observations for the evaluations: 1) DSLC is quite robust on different ω, excepting very small and large ω values, i.e., 0, 0.1, 0.2, 0.9, 1, because a small ω results in inadequate learning and a large ω makes the learning unstable. Thus, $\omega = 0.5$ is reasonable for DSLC; 2) The performance on PT and LE does not change on different ω, because the distance structures initialized by DSLC is similar to the intrinsic distance structures of PT and LE; 3) DSLC spend less than 20 iterations to converge, which is fast for updating a large number of basic distances in $\mathbf{L}$; 4) After updating $\mathbf{L}$, Z is obviously reduced (see the iterations pointed out by red arrows), which indicates the effectiveness of DSLC.

4 CONCLUSION

In this paper, we have proposed DSLC for ordinal data clustering. Differing from the existing approaches, DSLC integrates the distance learning and data partitioning into a learning algorithm to obtain accurate clustering results. DSLC is easy to use with only one easy-to-set parameter. More importantly, it features superior clustering accuracy, low time complexity, and fast convergence. Experiments on different real and benchmark data sets illustrate its efficacy.

REFERENCES

[1] Amir Ahmad and Lipika Dey. 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 (2007), 110–118.

[2] Elaine I Allen and Christopher A Seaman. 2007. Likert scales and data analyses. *Quality progress* 40, 7 (2007), 64–65.

[3] Yiu-Ming Cheung and Hong Jia. 2013. Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number. *Pattern Recognition* 46, 8 (2013), 2228–2238.

[4] Alexander J Gates and Yong-Yeol Ahn. 2017. The impact of random models on clustering similarity. *The Journal of Machine Learning Research* 18, 1 (2017), 3049–3076.

[5] Joshua Zhexue Huang, Michael K Ng, Hongqiang Rong, and Zichen Li. 2005. Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 5 (2005), 657–668.

[6] Dino Ienco, Ruggero G Pensa, and Rosa Meo. 2012. From context to distance: Learning dissimilarity for categorical data clustering. *ACM Transactions on Knowledge Discovery from Data* 6, 1 (2012), 1–25.

[7] Hong Jia and Yiu-Ming Cheung. 2018. Subspace clustering of categorical and numerical data with an unknown number of clusters. *IEEE Transactions on Neural Networks and Learning Systems* 29, 8 (2018), 3308–3325.

[8] Hong Jia, Yiu-ming Cheung, and Jiming Liu. 2016. A new distance metric for unsupervised learning of categorical data. *IEEE Transactions on Neural Networks and Learning Systems* 27, 5 (2016), 1065–1079.

[9] Si Quang Le and Tu Bao Ho. 2005. An association-based dissimilarity measure for categorical data. *Pattern Recognition Letters* 26, 16 (2005), 2549–2557.

[10] Michael K Ng, Mark Junjie Li, Joshua Zhexue Huang, and Zengyou He. 2007. On the impact of dissimilarity measure in k-modes clustering algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 3 (2007), 503–507.

[11] Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles–a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3, Dec (2002), 583–617.

[12] Yiqun Zhang and Yiu-ming Cheung. 2018. Exploiting order information embedded in ordered categories for ordinal data clustering. In *Proceedings of the International Symposium on Methodologies for Intelligent Systems*. Springer, 247–257.