ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240709

Learning Order Forest for Qualitative-Attribute Data Clustering

Mingjie Zhao[1], Sen Feng[1], Yiqun Zhang[1,6,*], Mengke Li[2,3,6], Yang Lu[4,5,6] and Yiu-Ming Cheung[6]

[1]School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China
[2]Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China
[3]School of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
[4]Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, China
[5]Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, China
[6]Department of Computer Science, Hong Kong Baptist University, Hong Kong, China

Abstract. Clustering is a fundamental approach to understanding data patterns, wherein the intuitive Euclidean distance space is commonly adopted. However, this is not the case for implicit cluster distributions reflected by qualitative attribute values, e.g., the nominal values of attributes like symptoms, marital status, etc. This paper, therefore, discovered a tree-like distance structure to flexibly represent the local order relationship among intra-attribute qualitative values. That is, treating a value as the vertex of the tree allows to capture rich order relationships among the vertex value and the others. To obtain the trees in a clustering-friendly form, a joint learning mechanism is proposed to iteratively obtain more appropriate tree structures and clusters. It turns out that the latent distance space of the whole dataset can be well-represented by a forest consisting of the learned trees. Extensive experiments demonstrate that the joint learning adapts the forest to the clustering task to yield accurate results. Comparisons of 10 counterparts on 12 real benchmark datasets with significance tests verify the superiority of the proposed method. Source code of the proposed method is available at [39].

1 Introduction

Datasets composed of multi-valued qualitative attributes (also known as categorical or nominal attributes) are ubiquitous in cluster analysis tasks [13, 26, 29], for instance, the clustering of clients, patients, and so on. Unlike a numerical attribute with all its values distributed on an Euclidean distance axis, the qualitative values of a categorical attribute cannot reflect its distance structure. For example, there are three possible values {driver, lawyer, nurse} for attribute "occupation", but their optimal numerical embedding on a distance axis is unknown. Therefore, most related works are dedicated to mining this implicit distance structure [1, 18, 10], and can be roughly divided into: 1) Distance measures and 2) Distance learning methods [3], according to whether they connect to the downstream clustering tasks.

It is noteworthy that distance measures for qualitative values, e.g., Hamming distance [4], simply perform a boolean distance measurement based on whether two values are the same or not. Although subsequent measures [2, 25, 22] introduce various data statistical information to improve distance discrimination, they still treat each inter-value distance in isolation without considering the overall distance structure of all possible values. To address this issue, information entropy is introduced to effectively couple the possible values of an attribute, and an entropy-based distance metric [27] is formed accordingly to more appropriately quantify the distances. Recently, more distance metrics [37, 35, 36] attempt to orderly embed possible values into a distance axis to obtain a distance structure similar to that of numerical attributes. However, their value order relies on the explicit semantic order of the values, e.g., {strong_accept, clear_accept, weak_accept} for an ordinal attribute "review recommendation", which is often unavailable when dealing with nominal attributes.

Distance learning methods focusing on connecting distance measurements and clustering tasks have received more attention in recent years, as they can often obtain distance structures that are more

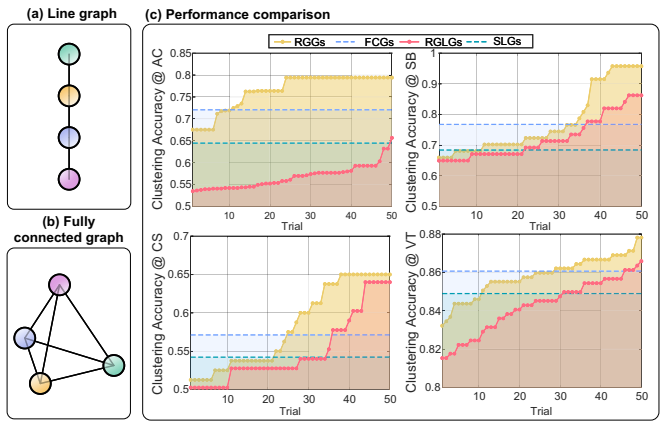


Figure 1. An intuitive comparison of clustering performance by adopting different types of distance structures. (a) and (b) demonstrate the typical line graph and fully connected graph. (c) demonstrates the k-modes [16] clustering performance with the following distance structures: 1) Randomly Generated Graphs (RGGs, not necessarily fully connected but ensure all attribute values are connected), 2) Fully Connected Graphs (FCGs), 3) Randomly Generated Line Graphs (RGLGs), and 4) Semantic Line Graphs (SLGs, arrange possible values in the graph according to their semantic order). The RGGs and RGLGs involving randomization are implemented 50 times, and the clustering accuracy is sorted for better visualization.

* Corresponding Author. Email: yqzhang@gdut.edu.cn

suitable for clustering. An early attempt [9] models sample-cluster similarity as the occurrence probability of possible values in clusters. Later, approaches that directly model the distance space have been proposed, including kernel-based [41] and graph-based [33, 34] distance metric learning. However, they are all based on specific hypotheses, e.g., specific kernels can well-represent the distance metric, or the distance metric follows a graph structure of possible values inspired by specific domain knowledge.

Benefiting from the universality of the graph, the graph-based learning approaches [33, 34, 38, 36, 8] are proven to achieve more competitive clustering performance. More specifically, a graph has been adopted to represent the relationship among possible values of an attribute. For the ordinal attribute values with explicit semantic order, they adopt a line graph shown in Figure 1 (a) whereby the weights of edges are learned to indicate the distances. For the nominal attribute values without semantic order, they use a fully connected graph shown in Figure 1 (b) to facilitate distance learning. Nevertheless, a pair of coupled thorny problems still lies ahead: reasonable prior knowledge is the premise of effective distance learning whilst the data knowledge is usually obtained by observing data distribution under well-defined distance metrics.

The limitations brought by the prior knowledge can be fully verified by Figure 1 (c). It can be seen that clustering under the two types of random graphs, i.e., RGGs and RGLGs, is significantly more promising to obtain higher accuracy compared to FCGs and SLGs. Moreover, RGGs obviously outperform RGLGs probably because RGGs do not overly restrict the relationship among attribute values to follow an order, thus laying the foundation for obtaining the latent optimal relationship through randomization. The above observations provide two hints: 1) A higher degree of topological freedom for the distance structure brings better clustering results, and 2) Explicit semantic order may not be optimal for clustering. Hence, how to obtain the optimal distance structure w.r.t. certain clustering tasks without relying on prior knowledge of the value relationship is crucial for breaking through the current clustering performance.

In this paper, a new qualitative data learning paradigm that performs Clustering with Order Forest learning (COForest) is proposed. The learning process is no longer limited to adjusting the distance between values under the hypothesized value graph, but allows both graph structure and distances to be jointly learned with clustering. It learns by iteratively: 1) Inferring graph structures w.r.t. the current data partition, and 2) Performing clustering using the graph distance structure to more appropriately obtain data partition. Since the inferred graphs are minimal spanning trees, they can concisely and flexibly represent the relationship among possible values. It turns out that the learning processes repeatedly improve the upper bound and approach it, thus bypassing sub-optimal solutions and achieving superior clustering accuracy. Main contributions of this work are summarized into three-fold:

- A new insight is introduced that there exists an optimal latent graph w.r.t. certain clustering tasks in representing the distance structure of a qualitative attribute, and the graph should be flexibly determined without being restricted by prior knowledge.
- COForest is proposed to iteratively optimize the distance structures and clusters to circumvent sub-optimal solutions. Compared with the existing approaches that only tune distances under a given topology, COForest further allows the reconstruction of the topology and thus brings a higher degree of learning freedom.
- Comprehensive experimental evaluations including significance tests, ablation studies, and qualitative visual comparisons, have been conducted to demonstrate the superiority of thoroughly learning distance structures without prior knowledge bias.

2 Propose Method

2.1 Problem Formulation

The problem of categorical data clustering with distance learning is formulated below. Given a categorical dataset $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ with n data samples. Each sample $\mathbf{x}_i$ can be denoted as an l-dimensional row vector $\mathbf{x}_i = [x_{i,1}, x_{i,2}, .., x_{i,l}]^\top$ represented by l attributes $A = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_l\}$. Each attribute $\mathbf{a}_r$ can be denoted as a column vector $\mathbf{a}_r = [x_{1,r}, x_{2,r}, ..., x_{n,r}]$ composed of the r-th values of all the n samples, where the n values can be viewed as sampled from a limited number of possible values $V_r = \{v_{r,1}, v_{r,2}, ..., v_{r,o_r}\}$ with o_r indicating the number of possible values of $\mathbf{a}_r$, and $v_{r,g}$ indicating the g-th possible value of $\mathbf{a}_r$.

Partitional clustering aims to partition X into k non-overlapping sample subsets $C = \{C_1, C_2, ..., C_k\}$ with the objective of minimizing the intra-subset dissimilarity [5, 11, 30], which is conventionally expressed as

$$L(\mathbf{Q}) = \sum_{j=1}^{k} \sum_{i=1}^{n} q_{i,j} \cdot \Gamma(\mathbf{x}_i, C_j), \tag{1}$$

where $\mathbf{Q}$ is an $n \times k$ matrix with its (i,j)-th entry $q_{i,j} \in \{0, 1\}$ indicating the affiliation between sample $\mathbf{x}_i$ and cluster C_j, and each row of $\mathbf{Q}$ (e.g., the i-th row) satisfies $\sum_{j=1}^{k} q_{i,j} = 1$. During clustering, the values of $q_{i,j}$ are determined by

$$q_{i,j} = \begin{cases} 1, & \text{if } j = \arg\min_{y} \Gamma(\mathbf{x}_i, C_y) \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The sample-cluster dissimilarity $\Gamma(\mathbf{x}_i, C_j)$ is collectively reflected by the value-level dissimilarities of different attributes by

$$\Gamma(\mathbf{x}_i, C_j) = \sum_{r=1}^{l} \gamma(x_{i,r}, C_{j,r}), \tag{3}$$

where $\gamma(x_{i,r}, C_{j,r})$ is the dissimilarity between the sample $\mathbf{x}_i$ and the cluster C_j from the perspective of attribute $\mathbf{a}_r$.

Since the relationship among possible values remains to be defined, it is not straightforward to compute the dissimilarity between a sample and a sample set C_j. Therefore, the key to this work is how to reasonably define the distance $\gamma(x_{i,r}, C_{j,r})$ in the attribute aspect. In the following, we first show how to flexibly model the value-level relationship, based on which the corresponding distance metric is defined. Then the joint learning scheme is proposed to make the defined distance structure learnable with clustering.

2.2 Order Forest Construction

As mentioned in Section 1, both the line graph and the fully connected graph have significant shortcomings in representing the distance structure of an attribute. That is, a line graph forces the relationship among all the possible values to be an order, and its effectiveness heavily relies on the prior order knowledge of the possible values. By contrast, a fully connected graph constructs multiple paths for each pair of possible values, and thus cannot concisely and exactly reflect the relationship among possible values.

Therefore, we propose to construct order forest $M = \{M_1, M_2, \ldots, M_l\}$ where M_r is a Minimal Spanning Tree (MST) corresponding to the r-th attribute $\mathbf{a}_r$. Each tree M_r is denoted as

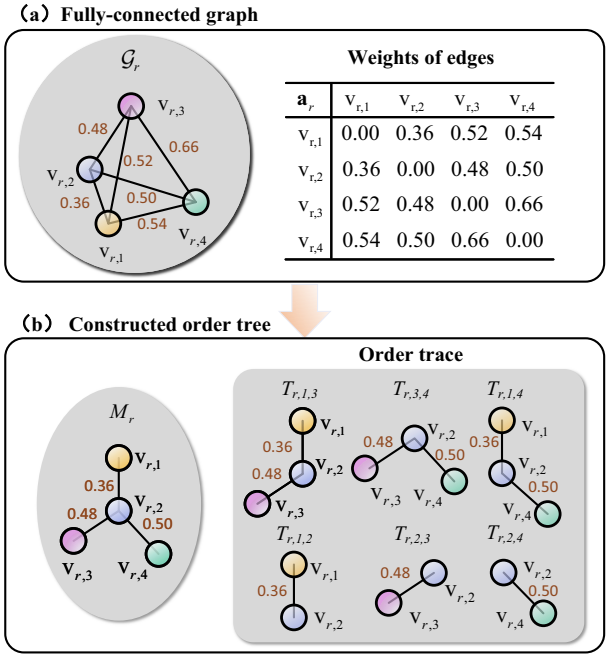


Figure 2. Process of order tree construction. (a) A fully connected graph $\mathcal{G}_r$ is prepared with a distance matrix reflecting the edge weights. (b) Prim or Kruskal algorithm is implemented to generate an order tree with a unique order trace between each pair of nodes, which is defined in Definition 1.

a tuple $M_r =< V_r, B_r, W_r >$ where all the o_r possible values in V_r are treated as nodes. B_r is the set of $o_r - 1$ edges that have the minimum sum of edge lengths and can connect all the o_r nodes. W_r contains the weights reflecting the edge lengths. Such a tree can be searched through the Prim or Kruskal [31, 24] algorithm given a fully connected graph with exact edge weights as shown in Figure 2(a). The weights are actually the distance between any pair of possible values, which can be computed using any existing distance measure defined for qualitative values. From Figure 2(b), it can be seen that the constructed MST concisely and exactly reflects the local order relationship among possible values, and is thus called an order tree.

Remark 1 (Generalization of relationship graph)**.** *Given an attribute $\mathbf{a}_r$, the constructed order tree M_r represents the order relationship among different possible value subsets and provides a unique trace between each pair of possible values as shown in Figure 2(b). Therefore, M_r can be regarded as generalized from: 1) The line graph representing all the values in one order, and 2) The fully connected graph representing redundant relationships among possible values.*

To define dissimilarity between nodes according to the order tree, we first define the order trace between two nodes.

Definition 1 (Order trace)**.** *Given an order tree M_r, order trace $T_{r,s,u}$ is a set containing all the weights between the nodes located on the shortest path from node $v_{r,s}$ to $v_{r,u}$. Since the order tree is undirected, $T_{r,s,u} = T_{r,u,s}$ holds.*

It turns out that each order trace uniquely concatenates a certain number of closely connected nodes, while the other nodes that are further away are excluded. This allows the order tree to flexibly represent local order relationships of possible value subsets, thus yielding a higher degree of freedom in distance structure learning. The dissimilarity between two nodes $v_{r,s}$ and $v_{r,u}$ can be intuitively reflected by their order trace, e.g., by adding all the weights on the trace. So far, the definition of weights plays a key role in constructing order forest and forming the dissimilarity between nodes. Since we focus on the clustering task, the weights and dissimilarity are defined by sufficiently leveraging the cluster information in Section 2.3, and the learning of the dissimilarity is incorporated with clustering in Section 2.4.

2.3 Clustering-Friendly Trace Distance

Given cluster partition $\mathbf{Q}$, weights of a fully connected graph should be first computed and then the order tree extracted from it forms value-level dissimilarities. Specifically, weight between two nodes $v_{r,u}$ and $v_{r,s}$ is defined as the distance between their probability distributions extracted from different clusters by

$$w_{r,u,s} = \left\| \mathbf{p}_{v_{r,u}} - \mathbf{p}_{v_{r,s}} \right\|_p, \tag{4}$$

where $\mathbf{p}_{v_{r,u}} = \left[p_{C_1|v_{r,u}}, p_{C_2|v_{r,u}}, \ldots, p_{C_k|v_{r,u}}\right]$ and $\mathbf{p}_{v_{r,s}} = \left[p_{C_1|v_{r,s}}, p_{C_2|v_{r,s}}, \ldots, p_{C_k|v_{r,s}}\right]$ are k-dimensional vectors representing the probability distributions of $v_{r,u}$ and $v_{r,s}$ across all the k clusters. $p_{C_j|v_{r,u}} = |X_{r,u} \cap C_j|/|X_{r,u}|$ where $|\cdot|$ counts the number of samples in a set and $X_{r,u} = \{\mathbf{x}_i | x_{i,r} = v_{r,u}\}$ is a sample set collecting all the samples in X with their r-th values equal to $v_{r,u}$. The symbol $\|\cdot\|_p$ represents p-norm, which intuitively reflects the difference in the direct probability distribution of nodes, where we adopt a common setting $p = 2$. A distribution $\mathbf{p}_{v_{r,u}}$ describes the distribution pattern of a value across all the k clusters, so that two values with similar patterns are considered to be more similar. We therefore use weights defined in Eq. (4) to construct the order tree that is with the likelihood of producing the current partition $\mathbf{Q}$.

With constructed order tree M_r and the weight defined in Eq. (4), the dissimilarity between two possible values $v_{r,u}$ and $v_{r,s}$ is defined as the length of their order trace as defined in Definition 1, which can be written as

$$d_{r,u,s} = \sum_{w_{r,u,s} \in T_{r,u,s}} w_{r,u,s}. \tag{5}$$

Since this is computed based on the weights defined in a clustering-friendly manner by Eq. (4), we thus call it clustering-friendly trace distance. Accordingly, the sample-cluster distance $\gamma(x_{i,r}, C_{j,r})$ reflected by the order tree structure M_r can be defined based on the value-level trace distance as

$$\gamma(x_{i,r}, C_{j,r}; M_r) = \mathbf{p}_{j,r}^{\top} \mathbf{d}_{r,u}, \tag{6}$$

where we assume that the sample value $x_{i,r}$ equals to the possible value $v_{r,u}$ for simplicity without loss of generality. $\mathbf{d}_{r,u} = [d_{r,u,1}, d_{r,u,2}, ..., d_{r,u,o_r}]$ is an o_r-dimensional vector containing the trace distances between $v_{r,u}$ and each of the o_r possible values in V_r, and $\mathbf{p}_{j,r} = [p_{v_{r,1}|C_j}, p_{v_{r,2}|C_j}, ..., p_{v_{r,o_r}|C_j}]$ is an o_r-dimensional vector describing the probability distribution of possible values in V_r within cluster C_j, where $p_{v_{r,u}|C_j} = |X_{r,u} \cap C_j|/|C_j|$.

Correspondingly, the overall sample-cluster distance $\Gamma(\mathbf{x}_i, C_j)$ defined based on the whole order forest M can be formulated as

$$\Gamma(\mathbf{x}_i, C_j; M) = \sum_{r=1}^{l} \gamma(x_{i,r}, C_{j,r}; M_r). \tag{7}$$

Theorem 1. *The trace distance measure $d_{r,u,s}$ defined in the context of the order tree M_r represents a valid distance metric.*

Proof. $d_{r,u,s}$ follows non-negativity, symmetry, and triangle inequality for any $r \in \{1, 2, ..., l\}$ and $u, s, g \in \{1, 2, ..., o_r\}$ as shown below.

Non-negativity: $d_{r,u,s} \geq 0$. $d_{r,u,s}$ is the length of an order trace, which is always non-negative comprising non-negative weights according to Eqs. (4) and (5) with the norm set at $p = 2$;
Symmetry: $d_{r,u,s} = d_{r,s,u}$. Since the order tree is an undirected graph, the weights on the trace extracted from the undirected graph obey the commutative law for their summation;
Triangle inequality: $d_{r,u,s} \leq d_{r,u,g} + d_{r,g,s}$. The order trace is the unique path between two values with length $d_{r,u,s}$. From $v_{r,u}$ to $v_{r,s}$, detour another node $v_{r,g}$ other than the trace $T_{r,u,s}$ necessarily involves extra weight(s) from the other traces. Given that each weight is non-negative, the result follows. □

Theorem 2. *The sample-cluster distance $\Gamma(\mathbf{x}_i, C_j; M)$ defined in the context of the order forest M represents a valid distance metric.*

Proof. The computation of $\Gamma(\mathbf{x}_i, C_j; M)$ can be viewed as the weighted sum of a series of $d_{r,u,s}$s in $\mathbf{d}_{r,u}$s with non-negative weights represented by the probabilities in $\mathbf{p}_{j,r}$s according to Eqs. (6) and (7). Since trace distance $d_{r,u,s}$ is a metric according to Theorem 1, $\Gamma(\mathbf{x}_i, C_j; M)$ is also a metric following non-negativity, symmetry, and triangle inequality. □

2.4 Joint Learning Algorithm

Joint learning of cluster and order forest is facilitated by integrating the order forest construction mechanism presented in Sections 2.2 and 2.3 into the clustering objective. Accordingly, $L(\mathbf{Q})$ can be refined to $L(\mathbf{Q}, M)$ based on Eqs. (1), (5), (6), and (7):

$$L(\mathbf{Q}, M) = \sum_{j=1}^{k} \sum_{i=1}^{n} q_{i,j} \cdot \sum_{r=1}^{l} \gamma(x_{i,r}, C_{j,r}; M_r). \tag{8}$$

Then the problem becomes how to compute $\mathbf{Q}$ and M to minimize L, which is typically solved by iteratively fixing one and computing another. Specifically, given fixed distance structure $\hat{M}$, $\mathbf{Q}$ can be computed to minimize $L(\mathbf{Q}, \hat{M})$ by

$$q_{i,j} = \begin{cases} 1, & \text{if } j = \arg\min_y \sum_{r=1}^{l} \gamma(x_{i,r}, C_{y,r}; M_r) \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

with $i = \{1, 2, ..., n\}$ and $j = \{1, 2, ..., k\}$. Eq. (9) is strictly derived from Eq. (2) by adopting order forest $\hat{M}$ as its distance structure. After the $\mathbf{Q}$ is computed, we fix it as $\hat{\mathbf{Q}}$ and then reconstruct M according to Figure 2 and Eqs. (4) - (5).

In summary, L is optimized by iteratively solving the two minimization problems: 1) Fix $\hat{M}$, run k-modes [16] to iteratively compute $\mathbf{Q}$ until convergences; 2) Fix $\hat{\mathbf{Q}}$, reconstruct M to update the distance metric. With a finite state space of M, the states will gradually be exhausted during the iterative searching, and thus the convergence of the algorithm can be guaranteed. The whole algorithm is summarized as Algorithm 1.

Theorem 3. *Time complexity of COForest is $O(nlk\mathcal{I}\mathcal{E})$*

Proof. To more intuitively provide the proof, we first define the probability $\mathbf{P}$ and trace distance matrix $\mathbf{D}$, and assume that $\varsigma = \max(o_1, o_2, ..., o_l)$ for worst-case analysis. $\mathbf{P}$ is a $k \times l$ probability matrix with its (j, r)-th entry $\mathbf{p}_{j,r} = [p_{v_{r,1}|C_j}, p_{v_{r,2}|C_j}, ..., p_{v_{r,o_r}|C_j}]$. $\mathbf{D}$ is an $l \times \varsigma$ trace distance matrix, and its (r, u)-th entry is $\mathbf{d}_{r,u} = [d_{r,u,1}, d_{r,u,2}, ..., d_{r,u,o_r}]$.

Assume solving problem $L(\mathbf{Q}, \hat{M})$ involves $\mathcal{I}$ iterations to compute $\mathbf{Q}$ and $\mathbf{P}$, and the whole algorithm involves $\mathcal{E}$ iterations to construct M and update $\mathbf{D}$ for solving $L(\hat{\mathbf{Q}}, M)$.

Algorithm 1: COForest: Clustering with Order Forest Learning

Require: Dataset X, number of sought clusters k
Ensure : Partition $\mathbf{Q}$, order forest M

Initialization: Set outer and inner loop counters by $\mathcal{E} \leftarrow 0$ and $\mathcal{I} \leftarrow 0$; Run k-modes [16] to obtain a relatively stable initial $\mathbf{Q}^{\{\mathcal{E}\}}$; Construct initial $M^{\{\mathcal{E}\}}$ according to $\mathbf{Q}^{\{\mathcal{E}\}}$; Set convergence mark for outer loop by $Conv\_\mathcal{E} \leftarrow False$.
while $Conv\_\mathcal{E} = False$ **do**
  $Conv\_\mathcal{I} \leftarrow False$;
  while $Conv\_\mathcal{I} = False$ **do**
    $\mathcal{I} \leftarrow \mathcal{I} + 1$; Compute $\mathbf{Q}^{\{\mathcal{I}\}}$ by Eq. (9);
    if $\mathbf{Q}^{\{\mathcal{I}\}} = \mathbf{Q}^{\{\mathcal{I}-1\}}$ **then**
      $Conv\_\mathcal{I} \leftarrow True$;
    end
  end
  if $\mathbf{Q}^{\{\mathcal{E}\}} = \mathbf{Q}^{\{\mathcal{I}\}}$ **then**
    $Conv\_\mathcal{E} \leftarrow True$;
  else
    $\mathcal{E} \leftarrow \mathcal{E} + 1$; $\mathbf{Q}^{\{\mathcal{E}\}} \leftarrow \mathbf{Q}^{\{\mathcal{I}\}}$; Reconstruct $M^{\{\mathcal{E}\}}$;
  end
end

For each iteration of $\mathcal{I}$, $\mathbf{P}$ should be prepared by going through all the n data samples once with complexity $O(nkl\varsigma)$, and $\mathbf{D}$ should be prepared by going through all the ς values of l attributes once on n samples with complexity $O(nl\varsigma)$. Then the n samples are clustered to k clusters by considering ς values of l attributes according to Eq. (9), with time complexity $O(nkl\varsigma)$. Therefore, the time complexity of solving $L(\mathbf{Q}, \hat{M})$ in a total of $\mathcal{I}$ iterations is $O(\mathcal{I}nkl\varsigma)$.

For each iteration of $\mathcal{E}$, since $\mathbf{P}$ and $\mathbf{D}$ have been prepared, order tree of ς possible values of each of the l attributes can be searched by constructing M_r with time complexity $O(nl\varsigma^2)$. For $\mathcal{E}$ iterations of the whole COForest algorithm, considering the $\mathcal{I}$ inner iterations, the overall time complexity of COForest is $O(\mathcal{E}(\mathcal{I}nkl\varsigma + nl\varsigma^2))$.

Since ς is a small integer ranging from 2 to 8 in most cases, it can be treated as a constant, and the overall time complexity can be simplified to $O(nlk\mathcal{I}\mathcal{E})$, which is linear to n and l. □

3 Experiments

Five experiments are designed to evaluate the proposed COForest by comparing it with 10 counterparts on 12 real public datasets using three validity metrics. The experiments are summarized below:

- Clustering performance comparisons with significance tests illustrate that COForest significantly outperforms the conventional and state-of-the-art counterparts (Section 3.2).
- Ablation studies comparing five ablated versions of COForest confirm the effectiveness of each of the core components of COForest (Section 3.3).
- Convergence and efficiency of COForest are demonstrated by plotting the objective function values during learning and execution time under different dataset scales, respectively (Section 3.4).
- Reasonableness of the learned distance structure is well confirmed by qualitatively comparing the cluster discrimination ability of different methods using t-SNE (Section 3.5).
- The potential of extending COForest to mixed data with numerical and categorical attributes is validated by comparing its clustering performance with those specifically proposed for mixed data (please refer to the "Supplementary Material" provided by [39]).

Table 1. Information of the 10 counterparts. "Type" indicates whether a method separates or jointly learns the distance definition and clustering.

No.	Counterpart	Year	Type
1	KMD [16]	1998	Separate
2	LSM [27]	1998	Separate
3	JDM [20]	2016	Separate
4	CBDM [17]	2012	Separate
5	OCIL [9]	2013	Joint
6	UDMC [35]	2022	Separate
7	DLC [33]	2020	Joint
8	H2H [38]	2022	Joint
9	HDC [34]	2022	Joint
10	ADC [36]	2023	Separate

Table 2. Statistics of the 12 datasets. l and n are the numbers of attributes and samples, respectively. k^* is the true number of clusters.

No.	Dataset	Abbrev.	l	n	k^*
1	Hayes-Roth	HR	4	132	3
2	Car Evaluation	CE	6	1728	4
3	Australia Credit	AC	8	690	2
4	Congressional Voting	VT	16	435	2
5	Caesarian Section	CS	4	80	2
6	Soybean (small)	SB	35	47	4
7	Nursery School	NS	8	12960	4
8	Zoo	ZO	16	101	7
9	Thoracic Surgery	TS	13	470	2
10	Heart Failure	HF	5	299	2
11	Inflammations Diagnosis	DS	5	120	2
12	Lenses	LS	4	24	3

3.1 Experimental Setup

Experimental settings are briefly described below.

10 Counterparts are sorted out in Table 1. We set their hyper-parameters (if any) to the values recommended by the corresponding papers. Each method is implemented 10 times and the average performance is reported.

12 Datasets from various domains are utilized for the experiments. All the datasets are real public datasets collected from the UCI Machine Learning Repository [23], and the statistical information is shown in Table 2. Before the experiments, we preprocess the datasets by removing samples with missing values. Since we focus on categorical data clustering, numerical attributes in AC, TS, HF, and DS datasets are omitted. For all the compared methods, we set $k = k^*$ as the sought number of clusters.

Three Evaluation Metrics include the clustering accuracy (CA) [15], Adjusted Rand Index (ARI) [32, 14], and Normalized Mutual Information (NMI) [12], are adopted for evaluating clustering performance from different perspectives. Among them, CA is a conventional index, which computes the matching rate based on the best permutation mapping between the obtained clusters and the true classes. In contrast, ARI and NMI are more discriminative, being in value intervals $[-1, 1]$, and $[0, 1]$, respectively. For all the indices, a higher value indicates a better clustering performance. NMI results are provided in the "Supplementary Material" [39].

3.2 Clustering Performance

In this section, we investigate the clustering performance of different algorithms and statistically analyze the superiority of COForest.

Clustering performance of different methods are compared in Tables 3 and 4 w.r.t. CA and ARI, respectively. The best and second-best results on each dataset are highlighted in **bold** and underline, respectively. The observations include the following three aspects: 1) Overall, COForest performs best on almost all datasets, indicating its superiority in clustering. 2) The performance of COForest on the TS and HF datasets is not obviously better than the second-best method. However, the second-best method varies on these datasets, indicating the robustness of COForest. 3) Although COForest does not have the best CA and ARI performance on the VT dataset, it maintains the second-best and is not surpassed by much by the winners. In addition, the results of CBDM on CE, NS, and LS datasets are not reported because the attributes of these datasets are independent of each other, making CBDM fails in measuring distances according to the correlated attributes.

Significance tests are conducted by first implementing Friedman tests on the average performance ranks reported in the last rows in Tables 3 and 4, respectively. The corresponding p-values are 0.00020 and 0.00002, respectively, both passing the test under 99% confidence interval (i.e., p-value = 0.01). On this basis, Bonferroni Dunn (BD) post-hoc tests are implemented. Critical Difference (CD) intervals for the two-tailed BD tests at 95% (α= 0.05) and 90% (α = 0.1) confidence intervals are 3.8048 and 3.5204, respectively, for comparing 11 methods across 12 datasets. As can be seen from the "$\overline{AR}$" rows in Tables 3 and 4 that all compared methods fall outside the right boundary of the CD intervals, except for the DLC method w.r.t. ARI performance under $\alpha = 0.05$. But it is worth mentioning that DLC is very close to the boundary of $\alpha = 0.05$ and stays outside the boundary of $\alpha = 0.1$. In general, the test results indicate that the proposed COForest significantly outperforms the other counterparts.

3.3 Ablation Study

To explicitly demonstrate the effectiveness of the core components of COForest, several ablated versions of it are compared in Figure 3. To evaluate the proposed order forest learning mechanism, we compare COForest with COF^I, which constructs the order forest once without iterative learning. To evaluate the proposed order forest structure, COF^I is modified by replacing the order forest with line graphs and fully connected graphs to form COF^{II} and COF^{III}, respectively. Moreover, to verify the adopted probability distribution-based measure in Eq. (4) for weights computing, we further let COF^{III} adopt the traditional Hamming distance and form the version COF^{IV}.

It can be observed from Figure 3 that COForest outperforms its four variants, which generally illustrates its effectiveness. More specific observations are four-fold: 1) COForest performs not worse than COF^I on all the datasets, validating the necessity of the joint learning of the order forest and clustering. 2) On 10 out of 12 datasets, the performance of COF^I is not worse than COF^{II} and COF^{III}. This indicates that our constructed order forest is more reasonable in reflecting the distance structures, even without learning. The reason would be that the order tree is a generalized distance structure as analyzed in Remark 1, which can more flexibly represent multiple local order relationships. 3) The mutual win and loss of COF^{II} and COF^{III} across the 12 datasets reveals that both line graph and fully connected graph have their own limitations. 4) COF^{III} adopting probability distribution-based measure outperforms COF^{IV} adopting Hamming distance on 10 datasets, indicating that the use of the probability distributions in Eq. (4) is reasonable.

3.4 Convergence and Efficiency Evaluation

To evaluate the convergence of COForest, we plot its objective function values L during the learning on all the 12 datasets in Figure 4. The horizontal and vertical axes represent the number of learning

Table 3. Clustering performance evaluated by CA. "$\overline{AR}$" row reports the average performance rankings.

Data	KMD	LSM	JDM	CBDM	OCIL	UDMC	DLC	H2H	HDC	ADC	COForest (ours)
HR	0.3795±0.02	0.3826±0.03	0.3841±0.03	0.4083±0.06	0.3621±0.05	0.3886±0.01	0.3659±0.03	0.3333±0.00	0.3758±0.02	0.3970±0.05	**0.4530±0.07**
CE	0.3730±0.04	0.3587±0.04	0.3597±0.04	-	0.3659±0.05	0.3505±0.03	0.3746±0.04	0.3354±0.06	0.3730±0.04	0.3730±0.04	**0.4261±0.06**
AC	0.7494±0.05	0.7823±0.04	0.6858±0.12	0.7417±0.08	0.7781±0.10	0.7674±0.08	0.7499±0.14	0.7942±0.00	0.7484±0.09	0.7709±0.09	**0.8307±0.05**
VT	0.8621±0.01	0.8662±0.00	0.8662±0.00	0.8749±0.00	**0.8763±0.00**	0.8639±0.00	0.8540±0.08	0.8736±0.00	0.8736±0.00	0.8713±0.00	0.8761±0.00
CS	0.5475±0.02	0.5425±0.05	0.5475±0.05	0.5787±0.03	0.5037±0.18	0.5788±0.03	0.6013±0.04	0.6050±0.02	0.5862±0.03	0.5875±0.02	**0.6450±0.02**
SB	0.8191±0.18	0.8553±0.19	0.7830±0.16	0.8213±0.15	0.7936±0.33	0.8426±0.17	0.8723±0.17	0.9511±0.10	0.8128±0.13	0.8191±0.16	**0.9723±0.09**
NS	0.3454±0.04	0.3171±0.04	0.3064±0.03	-	0.3454±0.08	0.3235±0.04	0.3301±0.06	0.3441±0.05	0.3454±0.04	0.3454±0.04	**0.3626±0.09**
ZO	0.6564±0.10	0.6594±0.10	0.7149±0.09	0.6921±0.08	0.5663±0.31	0.6564±0.09	0.7020±0.10	0.6980±0.04	0.6713±0.12	0.6812±0.11	**0.7832±0.12**
TS	0.7083±0.08	0.6717±0.08	0.7023±0.10	0.6957±0.09	0.6689±0.08	0.7087±0.09	0.6868±0.08	0.5723±0.03	0.7104±0.08	0.6947±0.10	**0.7232±0.09**
HF	0.5344±0.03	0.5344±0.03	0.5421±0.03	0.5498±0.02	0.4880±0.17	0.5378±0.02	0.5381±0.02	0.5441±0.05	0.5388±0.02	0.5378±0.02	**0.5532±0.03**
DS	0.6833±0.11	0.6833±0.11	0.6975±0.11	0.7142±0.12	0.7242±0.16	0.6725±0.11	0.7242±0.16	0.6267±0.04	0.6725±0.11	0.6975±0.11	**0.7617±0.08**
LS	0.5250±0.07	0.5417±0.10	0.5417±0.10	-	0.5417±0.08	0.5792±0.14	0.5500±0.09	0.5167±0.09	0.5250±0.07	0.5250±0.07	**0.6833±0.14**
$\overline{AR}$	7.2500	7.2083	6.7917	6.4583	7.0833	6.5833	5.3750	6.3750	6.2083	5.5833	1.0833

Table 4. Clustering performance evaluated by ARI. "$\overline{AR}$" row reports the average performance rankings.

Data	KMD	LSM	JDM	CBDM	OCIL	UDMC	DLC	H2H	HDC	ADC	COForest (ours)
HR	-0.0064±0.01	-0.0051±0.01	-0.0048±0.01	0.0127±0.03	-0.0073±0.02	-0.0037±0.00	-0.0068±0.01	-0.0149±0.00	-0.0056±0.01	0.0043±0.03	**0.0429±0.04**
CE	0.0229±0.03	0.0314±0.02	0.0321±0.02	-	0.0501±0.06	0.0289±0.02	0.0676±0.03	0.0140±0.03	0.0229±0.03	0.0229±0.03	**0.1016±0.07**
AC	0.2575±0.10	0.3228±0.07	0.1892±0.17	0.2569±0.11	0.3421±0.15	0.3107±0.11	0.3178±0.22	0.3453±0.00	0.2714±0.12	0.3225±0.12	**0.4462±0.12**
VT	0.5233±0.02	0.5354±0.01	0.5354±0.01	0.5613±0.01	**0.5655±0.01**	0.5287±0.01	0.5208±0.18	0.5572±0.00	0.5572±0.00	0.5503±0.00	0.5647±0.00
CS	-0.0033±0.01	0.0017±0.03	0.0038±0.03	0.0137±0.01	0.0070±0.03	0.0140±0.02	0.0342±0.03	0.0319±0.02	0.0191±0.02	0.0190±0.01	**0.0732±0.02**
SB	0.7657±0.20	0.8164±0.24	0.6826±0.22	0.7652±0.21	0.7902±0.33	0.8232±0.19	0.8500±0.20	0.9271±0.16	0.7595±0.18	0.7863±0.19	**0.9562±0.14**
NS	0.0630±0.02	0.0556±0.02	0.0457±0.02	-	0.1146±0.10	0.0617±0.03	0.0886±0.08	0.0847±0.09	0.0630±0.02	0.0630±0.02	**0.1352±0.13**
ZO	0.5707±0.13	0.5872±0.15	0.6496±0.14	0.6187±0.12	0.5093±0.29	0.5937±0.15	0.6315±0.12	0.6255±0.06	0.6010±0.15	0.6128±0.15	**0.7511±0.18**
TS	0.0054±0.05	0.0123±0.05	0.0188±0.05	0.0171±0.04	-0.0048±0.05	0.0198±0.05	-0.0034±0.04	-0.0249±0.00	0.0084±0.05	0.0031±0.04	**0.0220±0.04**
HF	-0.0067±0.00	-0.0067±0.00	-0.0013±0.01	-0.0009±0.01	-0.0002±0.00	-0.0023±0.00	-0.0005±0.00	-0.0043±0.00	-0.0019±0.00	-0.0023±0.00	**0.0045±0.01**
DS	0.1697±0.18	0.1697±0.18	0.1944±0.19	0.2280±0.20	0.2839±0.32	0.1543±0.18	0.2839±0.32	0.0615±0.04	0.1543±0.18	0.1944±0.19	**0.2901±0.16**
LS	0.0756±0.10	0.1180±0.15	0.1180±0.15	-	0.1287±0.12	0.1919±0.22	0.1379±0.15	0.0786±0.11	0.0756±0.10	0.0756±0.10	**0.3359±0.22**
$\overline{AR}$	8.6667	7.0000	6.5417	6.7500	5.2083	6.0833	4.7083	6.5417	7.0833	6.3333	1.0833

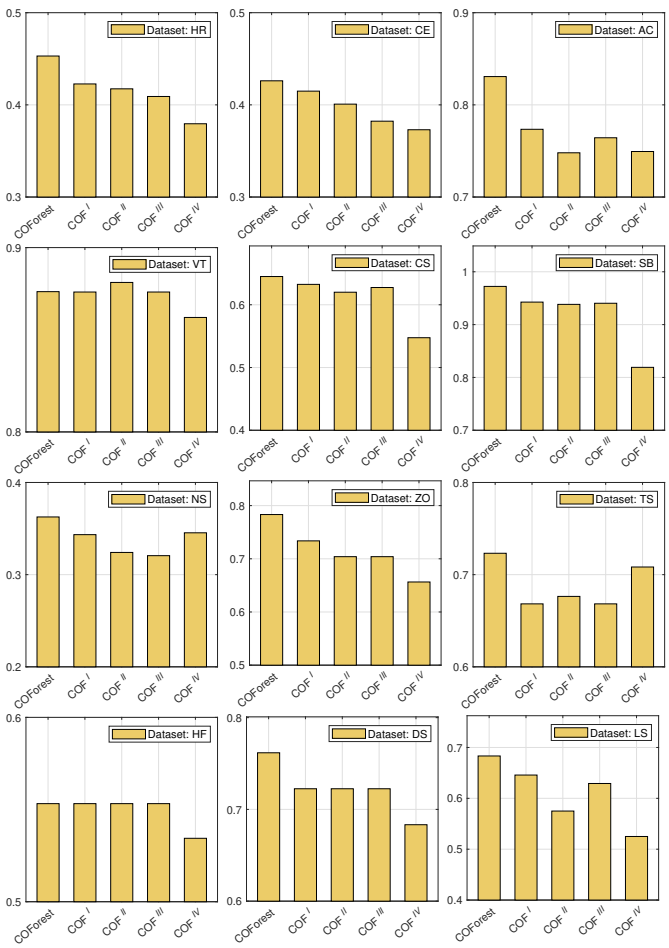


Figure 3. CA performance of different ablated COForest versions.

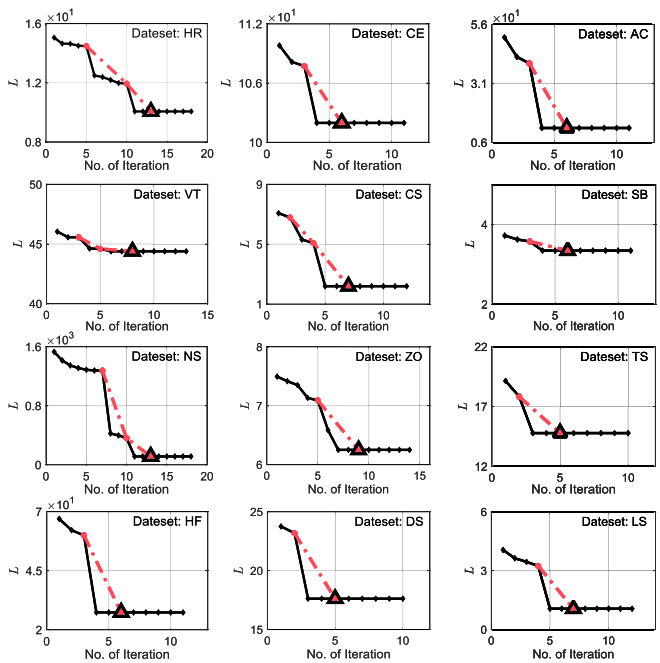


Figure 4. Convergence curves of COForest on different datasets. L represents the value of the objective function.

iterations and the value of L, respectively. The triangle markers on the curve represent the iterations that COForest converges and the red dots mark the iterations of order forest reconstruction. It can be observed that, after each update of the order forest, L decreases, indicating that the forest reconstruction is consistent with the minimization of L. Moreover, COForest converges within 15 iterations in most cases, which is quite efficient for a learning process that iteratively reconstructs the distance structure and learning data partitions.

To evaluate the efficiency of COForest, large synthetic datasets are

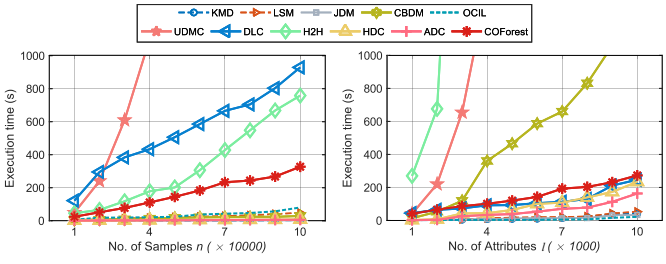
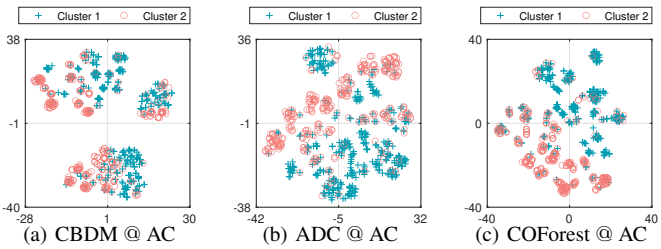


Figure 5. Execution time on synthetic datasets.



(a) CBDM @ AC (b) ADC @ AC (c) COForest @ AC

Figure 6. t-SNE visualization of the AC dataset.

randomly generated with different scales of attributes and samples. Specifically, we generate by: 1) Fixing the number of attributes at $l = 20$ and increasing the number of samples n from 10k to 100k with step-size 10k, and 2) Fixing sample size at $n = 2k$ and increasing the number of attributes l from 1k to 10k with step-size 1k, where 'k' indicates 'kilo'. Note that each attribute has five possible values, and the number of clusters k is consistently set to five. The execution time of all the 11 methods is demonstrated in Figure 5. It can be seen that the execution time of COForest is lower than or similar to the state-of-the-art UDMC, DLC, and H2H. Moreover, the increasing trend of the execution time of COForest is almost linear with n and l, which is consistent with the time complexity analysis of Theorem 3. In summary, COForest is efficient compared to the state-of-the-art methods and does not incur too much additional computational cost compared to the simplest methods.

3.5 Qualitative Evaluation

To illustrate the cluster discrimination capability of COForest and the intuitiveness of the distance structure it obtains, we use the distance between attribute values learned by COForest, CBDM, and ADC to encode the attributes of the AC dataset. The encoded data are then dimensionally reduced into a 2-D space through t-SNE [28] and visualized in Figure 6 by marking the data points with 'true' labels provided by the dataset. If more data points with the same label are gathered in the visualization, then it indicates that the corresponding distance metric is more competent in discriminating different clusters. It can be seen that COForest has significantly better cluster discrimination ability in the comparisons, which indicates the intuitiveness of its obtained clusters upon the tree-like distance structure.

4 Relate Work

This section overviews the existing distance-measure-based and distance-learning-based categorical data clustering methods.

Distance Measures for categorical data including the measures yielded by encoding strategies and the directly defined distance measures. Traditional data encoding techniques, such as one-hot encoding, use Hamming distances to encode each possible value into a new attribute. However, it fails to capture the full spectrum of dissimilarity between possible values due to its boolean nature [10, 6]. To overcome these limitations, statistical-based measures have been introduced that consider the frequency of intra-attribute values, thereby capturing lower information entropy for similar values and suggesting more reasonable distance metrics [27, 9]. Further advancements in this area have developed metrics that account for inter-attribute dependencies, providing a more holistic view of the data relationships [2, 25, 17]. Additionally, consideration of value order differentiates between nominal and ordinal attributes, with specific approaches defining distances by integrating semantic order, thereby obviously improving distance accuracy for ordinal data [37, 35].

Distance Learning methods incorporate the defining of distances into the learning process of clustering. This includes advanced representation learning techniques that dynamically encode categorical data. For instance, some studies use various kernels to untangle attribute couplings more effectively [41], while others develop mixed encoding strategies for both numerical and categorical attributes [21, 40]. These often require meticulous tuning of hyperparameters. A significant step forward in this domain is the introduction of parameter-free approaches that learn the optimal number of clusters and the distances [19, 7]. Another innovative strategy involves transforming nominal attributes into ordinal ones through geometric projections to optimize latent distances, significantly enhancing clustering performance by unifying the treatment of different types of categorical attributes [38]. Furthermore, [33] treats values with order information as line graphs and learns the graph weights. Later, the works [34, 36] further unify the distances of nominal and ordinal attributes and make them learnable with clustering.

These advances significantly improve clustering performance on categorical data. Nevertheless, coupled thorny problems still lie ahead: *reasonable prior knowledge* is the premise of effective *distance learning* whilst the *data knowledge* is usually obtained by observing data distribution under *well-defined distance metrics*.

5 Concluding Remarks

This paper demonstrates and analyzes the key issue that bottlenecks the current qualitative data clustering performance, i.e., distance learning is restricted by prior knowledge of the distance structure. Accordingly, a new learning paradigm called COForest is proposed, which incorporates the construction of distance structure into the learning process and achieves joint optimization with clustering. Given the number of sought clusters k, the learning process of COForest is parameter-free and can be easily applied to various datasets. Moreover, the learned tree-like distance structures are concise and highly interpretable, making them very suitable for representing the implicit distribution of qualitative data. Extensive experiments illustrate the superiority of COForest, as well as the effectiveness of its key technical components.

The proposed method demonstrates outstanding clustering performance on static qualitative data under the given 'true' number of clusters. In the future, it is promising to consider extending it to coping with more complex real situations, e.g., learning from *streaming* data composed of a *mixture* of quantitative and qualitative attributes with an *unknown* number of *imbalanced* clusters.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC) under grants: 62476063, 62102097, 62376233, and 62306181, the NSFC/Research Grants Council

(RGC) Joint Research Scheme under the grant N_HKBU214/21, the Natural Science Foundation of Guangdong Province under grants: 2024A1515010163 and 2023A1515012855, the General Research Fund of RGC under grants: 12201321, 12202622, and 12201323, the RGC Senior Research Fellow Scheme under grant SRFS2324-2S02, the Shenzhen Science and Technology Program under grant RCBS20231211090659101, and the Xiaomi Young Talents Program.

References

[1] A. Agresti. *Analysis of ordinal categorical data*. John Wiley & Sons, 2010.

[2] A. Ahmad and L. Dey. 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, 2007.

[3] M. Alamuri, B. R. Surampudi, and A. Negi. A survey of distance/similarity measures for categorical data. In *Proceedings of the 2014 International Joint Conference on Neural Networks*, pages 1907–1914, 2014.

[4] P. Arabie, N. D. Baier, C. F. Critchley, and M. Keynes. Studies in classification, data analysis, and knowledge organization. *Studies in Classification Data Analysis and Knowledge Organization, Cham, Switzerland: Springer*, 2006.

[5] L. Bai and J. Liang. Sparse subspace clustering with entropy-norm. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 561–568, 2020.

[6] S. Boriah, V. Chandola, and V. Kumar. Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 243–254, 2008.

[7] S. Cai, Y. Zhang, X. Luo, Y.-M. Cheung, H. Jia, and P. Liu. Robust categorical data clustering guided by multi-granular competitive learning. In *Proceedings of the 44th IEEE International Conference on Distributed Computing Systems*, 2024.

[8] J. Chen, Y. Ji, R. Zou, Y. Zhang, and Y.-M. Cheung. Qgrl: Quaternion graph representation learning for heterogeneous feature data clustering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024.

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

[10] T. R. dos Santos and L. E. Zárate. Categorical data clustering: What similarity measure to recommend? *Expert Systems with Applications*, 42(3):1247–1260, 2015.

[11] N. El Malki, R. Cugny, O. Teste, and F. Ravat. Decwa: Density-based clustering using wasserstein distance. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, page 2005–2008, 2020.

[12] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201, 2009.

[13] A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade, L. Abualigah, J. O. Agushaka, C. I. Eke, and A. A. Akinyelu. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, 110:104743, 2022.

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

[15] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, pages 507–514, 2005.

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

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

[18] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[19] H. Jia and Y.-M. Cheung. 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, 2018.

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

[21] S. Jian, L. Hu, L. Cao, and K. Lu. Metric-based auto-instructor for learning mixed data representation. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, pages 3318–3325, 2018.

[22] S. Jian, G. Pang, L. Cao, K. Lu, and H. Gao. Cure: Flexible categorical data representation by hierarchical coupling learning. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):853–866, 2018.

[23] M. Kelly, R. Longjohn, and K. Nottingham. UCI machine learning repository.

[24] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *In Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

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

[26] T. Leng, L. Zhao, X. Xiong, P. Cheng, and J. Zhou. Self-expressive network-based subspace clustering for deep embedding. In *Proceedings of the 26th European Conference on Artificial Intelligence*, pages 1357–1364, 2023.

[27] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304, 1998.

[28] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11):2579–2605, 2008.

[29] J. Makkar, S. Jain, S. Gupta, et al. Mfc: A multishot approach to federated data clustering. In *Proceedings of 26th European Conference on Artificial Intelligence*, pages 1672–1679. 2023.

[30] F. Nie, J. Xue, W. Yu, and X. Li. Fast clustering with anchor guidance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1898–1912, 2024.

[31] R. C. Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957.

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

[33] Y. Zhang and Y.-M. Cheung. An ordinal data clustering algorithm with automated distance learning. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 6869–6876, 2020.

[34] Y. Zhang and Y.-M. Cheung. Learnable weighting of intra-attribute distances for categorical data clustering with nominal and ordinal attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3560–3576, 2022.

[35] Y. Zhang and Y.-M. Cheung. A new distance metric exploiting heterogeneous inter-attribute relationship for ordinal-and-nominal-attribute data clustering. *IEEE Transactions on Cybernetics*, 52(2):758–771, 2022.

[36] Y. Zhang and Y.-M. Cheung. Graph-based dissimilarity measurement for cluster analysis of any-type-attributed data. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9):6530–6544, 2023.

[37] Y. Zhang, Y.-M. Cheung, and K. Tan. A unified entropy-based distance metric for ordinal-and-nominal-attribute data clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 31(1):39–52, 2020.

[38] Y. Zhang, Y.-M. Cheung, and A. Zeng. Het2hom: Representation of heterogeneous attributes into homogeneous concept spaces for categorical-and-numerical-attribute data clustering. In *International Joint Conference on Artificial Intelligence*, pages 3758–3765, 2022.

[39] M. Zhao, S. Feng, Y. Zhang, M. Li, Y. Lu, and Y.-M. Cheung. Code and supplementary material for " learning order forest for qualitative-attribute data clustering". In *Proceedings of 27th European Conference on Artificial Intelligence*, 2024. URL https://github.com/ZMJ-lucky/ECAI-24-COForest.git.

[40] C. Zhu, Q. Zhang, L. Cao, and A. Abrahamyan. Mix2vec: Unsupervised mixed data representation. In *Proceedings of the 7th International Conference on Data Science and Advanced Analytics*, pages 118–127, 2020.

[41] C. Zhu, L. Cao, and J. Yin. Unsupervised heterogeneous coupling learning for categorical representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):553–549, 2022.