Robust Categorical Data Clustering Guided by Multi-Granular Competitive Learning

Shenghong Cai[1], Yiqun Zhang[1*], Xiaopeng Luo[2], Yiu-Ming Cheung[3], Hong Jia[4], Peng Liu[1]
[1]Guangdong University of Technology, Guangzhou, China
[2]Guangzhou Huali College, Guangzhou, China
[3]Hong Kong Baptist University, Hong Kong SAR, China
[4]Guangdong Provincial Key Laboratory of Intelligent Information Processing, Shenzhen, China
3121005074@mail2.gdut.edu.cn, yqzhang@gdut.edu.cn, gordonlok@foxmail.com
ymc@comp.hkbu.edu.hk, hongjia1102@szu.edu.cn, liupeng@gdut.edu.cn
*Corresponding author

2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS) | 979-8-3503-8605-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDCS60910.2024.00035

***Abstract*—Data set composed of categorical features is very common in big data analysis tasks. Since categorical features are usually with a limited number of qualitative possible values, the nested granular cluster effect is prevalent in the implicit discrete distance space of categorical data. That is, data objects frequently overlap in space or subspace to form small compact clusters, and similar small clusters often form larger clusters. However, the distance space cannot be well-defined like the Euclidean distance due to the qualitative categorical data values, which brings great challenges to the cluster analysis of categorical data. In view of this, we design a Multi-Granular Competitive Penalization Learning (MGCPL) algorithm to allow potential clusters to interactively tune themselves and converge in stages with different numbers of naturally compact clusters. To leverage MGCPL, we also propose a Cluster Aggregation strategy based on MGCPL Encoding (CAME) to first encode the data objects according to the learned multi-granular distributions, and then perform final clustering on the embeddings. It turns out that the proposed MGCPL-guided Categorical Data Clustering (MCDC) approach is competent in automatically exploring the nested distribution of multi-granular clusters and highly robust to categorical data sets from various domains. Benefiting from its linear time complexity, MCDC is scalable to large-scale data sets and promising in pre-partitioning data sets or compute nodes for boosting distributed computing. Extensive experiments with statistical evidence demonstrate its superiority compared to state-of-the-art counterparts on various real public data sets.**

***Index Terms*—Cluster analysis, categorical feature, competitive learning, number of clusters, cluster granularity**

I. Introduction

Clustering is one of the most popular unsupervised learning techniques that divides objects into a certain number of clusters where each cluster is usually composed of similar objects [1], [2]. Clustering can be utilized as a learner for recognition tasks, including anomaly detection [3], recommendation [4], risk detection [5], etc. It can also be utilized as a general tool for mining knowledge from data, e.g., latent object distribution [6], potential feature association [7], etc. However, most existing clustering is based on numerical data where all the feature values are quantitative and can directly attend arithmetic calculation [8]–[10]. Another common type of data, i.e., categorical data [11], [12] composed of qualitative feature values as shown in Fig. 1, is usually overlooked.

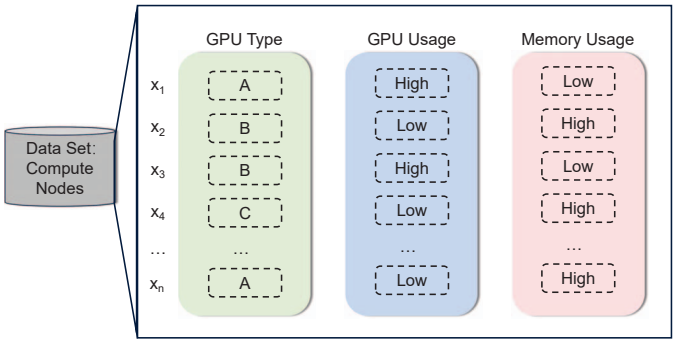


Fig. 1: Three categorical features (i.e., "GPU Type", "GPU Usage" and "Memory Usage") of a data set describing different compute nodes.

A common way for categorical data clustering is to encode the qualitative values into quantitative numerical values. However, the encoding process may easily cause information loss [13] as the implicit distribution of categorical data is difficult to be appropriately mapped into the Euclidean distance space. By contrast, some conventional categorical data clustering approaches [13]–[16] directly perform clustering by adopting distance metrics that are specifically defined for categorical data. Nevertheless, categorical data set is usually composed of features from various domains, which brings great challenges to defining a universal distance metric. Although hierarchical clustering [17] that outputs a dendrogram reflecting nested data object relationship can be utilized to understand the complex distribution of categorical data, the construction of dendrogram is laborious and may even fail due to the lack of powerful categorical data distance metric. Below we discuss the research progress of the above three streams of methods, i.e., 1) encoding-based, 2) distance defining-based, and 3) hierarchical clustering methods, and then refine the specific cutting-edge problems to be solved in this paper.

For the encoding-based stream, besides the most commonly used one-hot encoding [18], more advanced strategies [19], [20] that further consider the value-, object-, and feature-level couplings have been proposed for more informative represen-

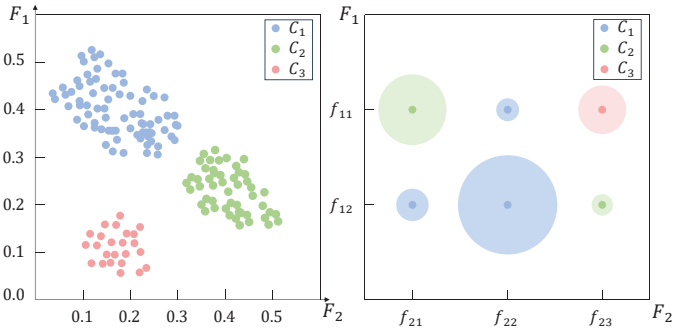
2575-8411/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDCS60910.2024.00035

tation. To make the representation learnable with the downstream clustering tasks, representation learning approaches [21]–[23] have also been proposed and obtained better categorical data clustering performance. Later, the research [24] further presents novel learning strategies to circumvent the non-trivial hyper-parameter selection of representation learning. Most recently, [25] introduces a multi-view projection technique to extract a more comprehensive representation of categorical feature values. Although the above-mentioned approaches have successively refreshed the clustering performance, they all focus on the improvement of encoding and learning techniques rather than the understanding of complex distributions and cluster effects of categorical data.

For the distance defining-based stream, the most popular Hamming distance [26] assigns distances "0" and "1" to pairs of identical and different values, respectively, from the perspective of the most basic value matching perspective. Entropy-based measures [27]–[31] and probability-based metrics [32]–[36], propose to quantify object-cluster affiliation based on more informative data statistics, e.g., occurrence frequency and conditional probability distribution of possible values, and have achieved more satisfactory clustering performance. Noticing the damage to the clustering performance caused by the heterogeneity of numerical and categorical features, some more advanced metrics [37]–[39] further unify the distance definitions of the two types of features from the perspective of probability. However, they mainly focus on the unification issue, and their performance will degrade when processing pure categorical data. The above measures and metrics are often combined with a partitional clustering algorithm for categorical data clustering. Since most of these algorithms require a given number of sought clusters k^*, they are incompetent in exploring and understanding the natural cluster distribution of categorical data.

Hierarchical clustering is considered a promising way for data distribution visualization and understanding, as it produces a tree-like nesting of data objects by recursively linking the current most similar object pairs. Representative link strategies include the conventional average-, complete-, and single-linkage [17], while recent advanced strategies [40]–[42] have also been explored. A link strategy that is specifically designed for categorical data has also been introduced in [43]. However, since the adopted dissimilarity metric acts as the basis for linkage computation, and hierarchical clustering lacks a learning mechanism capable of optimizing the metric, metric inappropriateness will all be unavoidably inherited. Moreover, as objects are treated as basic units during the recursive merging, implementing hierarchical clustering to large categorical data will be laborious. Although the most recent work [44] attempts to utilize statistical tests to guide the detection of significant clusters, unfortunately, the inherent bias of statistical tests makes them incapable of simultaneously detecting the multi-granular clusters that are prevalent in categorical data as shown in Fig. 2.

It can be seen that the limited feature values of categorical data make the objects overlapped at several points (e.g., the six points in Fig. 2(b)) in the distance space. The overlapping objects can be viewed as fine-grained clusters. Several such clusters form a larger cluster at a more coarse granularity, and so on, forming the nesting of multi-granular clusters. Based on the aforementioned analysis, it can be concluded that there is still a lack of cluster analysis methods that can effectively reveal the complex nested multi-granular cluster effect and are universally applicable to categorical data sets composed of qualitative features from various conceptual domains.



(a) Numerical data clusters (b) Categorical data clusters

Fig. 2: Comparison of clusters of numerical and categorical data. Since categorical data objects overlap on six points in (b), spheres with different radii are used to indicate the occurrence frequency of overlapping objects. The natural distance structure of categorical data leads to the nested multi-granular cluster effect (e.g., the green cluster is composed of two clusters with different granularity), which brings difficulties to cluster analysis.

This paper, therefore, proposes a new cluster analysis framework called MGCPL-guided Categorical Data Clustering (MCDC). First, a Multi-Granular Competitive Penalization Learning (MGCPL) mechanism is designed to automatically learn object partitions at different granularities by making the learning converge at different numbers of clusters. As MGCPL treats small clusters as the basic unit, the laborious nested relationship analysis is thus greatly alleviated. Compared with hierarchical clustering, the introduced learning mechanism facilitates intelligent multi-granular cluster detection. To leverage the analysis results of MGCPL, Cluster Aggregation based on MGCPL Encoding (CAME) has also been proposed to obtain partitional clustering results based on a given number of sought clusters. As its name suggests, clusters explored by MGCPL at each granularity are encoded to obtain informative embeddings, where the multi-granular information may complement each other and thus achieve more accurate clustering. It is worth noting that most existing clustering algorithms can be applied to the embeddings to obtain performance improvements. It turns out that the proposed method is robust and accurate on real categorical data sets from various domains, and its linear time complexity makes it highly scalable. Extensive experimental evaluations provide sufficient evidence of its effectiveness and efficiency.

The main contributions can be summarized into three-fold:

TABLE I: Frequently used symbols in the paper.

Symbols	Explanations
X	Data set
F	Features
C	Clusters
n	Number of data objects
d	Number of features
Q	Partition matrix of data objets
C_v	The winning cluster
C_h	The rival nearest winner
η	Learning rate
u	Cluster weights during competitive learning
σ	Number of granularity levels learned by MGCPL
κ	A series of ks learned by MGCPL
Γ	Data representation guided by MGCPL
Θ	Feature weights of representation Γ
k^*	The true number of clusters
Z	Mode of clusters

- To the best of our knowledge, this is the first attempt to reveal the complex but ubiquitous nested multi-granular cluster effect in categorical data, which is promising in inspiring subsequent works on categorical data analysis.
- A new cluster analysis mechanism called MGCPL is proposed to explore the nested multi-granular clusters of categorical data. MGCPL is efficient, and can provide rich data representation information for downstream tasks.
- An aggregation strategy called CAME is designed to fuse the multi-granular results of MGCPL. CAME achieves more accurate clustering, and its representation can also enhance existing categorical data clustering methods.

II. Preliminaries

This section first briefly introduces basic notations and frequently used symbols (see Table I), and then presents categorical data distance measurement and competitive learning mechanism that are highly related to the proposed method.

Given a categorical data set $X = \{x_i | i = 1, 2, ..., n\}$ with n data objects. Each object x_i is represented by d features $\{F_r | r = 1, 2, ..., d\}$. Thus, x_i can be represented as $x_i = [x_{i1}, x_{i2}, ..., x_{id}]^\top$ with $x_{ir} \in dom\,(F_r)$ and $r = 1, 2, ..., d$ where $dom\,(F_r) = \{f_{r1}, f_{r2}, ..., f_{rm_r}\}$ contains all the m_r possible values that can be chosen by feature F_r. In the common partitional clustering tasks, X should be divided into k clusters $C = \{C_l | l = 1, 2, ..., k\}$, i.e., a collection of k disjoint subsets of X, where C_l is the set of objects in the lth cluster and $X = \bigcup_{l=1}^{k} C_l$. Since distance measurement plays a key role in most existing categorical data clustering algorithms, we present an object-cluster distance measure for categorical data in the following.

A. Categorical Data Distance Measure

To achieve better adaptability between distance definition and clustering task, an object-cluster similarity denoted as $s\,(x_i, C_l)$ can be defined as

$$s\,(x_i, C_l) = \frac{1}{d}\left[\sum_{r=1}^{d} s(x_{ir}, C_l)\right] \tag{1}$$

where

$$s(x_{ir}, C_l) = \frac{\Psi_{F_r = x_{ir}}(C_l)}{\Psi_{F_r \neq NULL}(C_l)} \tag{2}$$

is the similarity reflected by the rth feature. Note that $\Psi_{F_r = x_{ir}}(C_l)$ counts the number of objects in cluster C_l that have value x_{ir} in feature F_r, and $\Psi_{F_r \neq NULL}(C_l)$ means the number of objects in cluster C_l that have values in the feature F_r. Intuitively, $s\,(x_i, C_l)$ is the average of similarities reflected by different features. Then we introduce how the competitive learning mechanism works on categorical data based on the above-defined distance to explore clusters.

B. Competitive Learning Algorithm

Most existing clustering methods assume the known true cluster number k^*, which is usually unavailable in real data analysis tasks, especially for data sets with complex distributions like categorical data. Competitive learning [45] mechanism is thus designed to learn the true number of clusters. The core idea of competitive learning is to make the initialized clusters compete with each other to eliminate clusters with less importance, and thus its objects will be carved up by the remaining clusters. In this way, by setting a relatively large initial k, the algorithm can gradually converge to k^* with a more stable and prominent cluster distribution. Such a learning mechanism is to maximize the overall intra-cluster similarity $S(Q)$:

$$S(Q) = \sum_{l=1}^{k}\sum_{i=1}^{n} u_l q_{il} s(x_i, C_l) \tag{3}$$

where q_{il} is the (i, l)th entry of Q, and is computed by

$$q_{il} = \begin{cases} 1, \text{ if } s(x_i, C_l) \geq s(x_i, C_t) \;\; \forall 1 \leq t \leq k \\ 0, \text{ otherwise.} \end{cases} \tag{4}$$

u_l is the weight of cluster C_l satisfying $0 \leq u_l \leq 1$ with $l = 1, 2, ..., k$. It measures the importance of C_j, and a higher weight indicates that the corresponding cluster is more prominent with less possibility to be eliminated.

During clustering, competitive learning is facilitated as follows. For each input x_i, the winning cluster C_v selected from initialized cluster candidates by

$$v = \arg\max_{1 \leq l \leq k} [u_l s(x_i, C_l)] \tag{5}$$

is updated toward x_i by a small step. To avoid the effect that some seed points located in marginal positions will immediately become dead units without learning chance in the subsequent learning process, wining chance of a frequent winning seed point will be gradually reduced. Accordingly, the winning frequency of different clusters can be computed to adjust the selection chance of the winner, and thus Eq. (5) can be re-written as

$$v = \arg\max_{1 \leq l \leq k} [(1 - \rho_l) u_l s(x_i, C_l)] \tag{6}$$

where

$$\rho_l = \frac{g_l}{\sum_{t=1}^{k} g_t} \tag{7}$$

is a winning ratio computed based on g_l, which is the winning times of cluster C_l in the last learning iteration. Accordingly, the weight of cluster C_v is updated by a small step controlled by a small learning rate η, which can be written as

$$u_l^{new} = u_l^{old} + \eta. \tag{8}$$

Note that the value of initial k should be set at a larger value than k^*, i.e., $k \geq k^*$, to ensure that the redundant clusters can be gradually eliminated during the learning process.

III. Proposed Method

This section first presents the MGCPL algorithm for exploring nested multi-granular distribution of clusters, and then introduces the CAME aggregation strategy to combine the multi-granular information provided by MGCPL to obtain the clustering results. The overall pipeline of the cluster analysis method composed of MGCPL and CAME is demonstrated in Fig. 3. Time complexity analysis, discussions on convergence and distributed computing issues, are provided at the end of this section.

A. MGCPL: Multi-granular Competitive Penalization Learning

In most real data cluster analysis tasks, it is not the case that a true number of cluster k^* can be known in advance, especially for categorical data with complex non-Euclidean distance space that are difficult to intuitively understand. Therefore, one of the most important issues in clustering is to estimate the most appropriate number of clusters. However, it is common that there are several ks suitable for the same data set, as the clusters can exist at different granularities, which is called the multi-granular effect. Such an effect is particularly evident in categorical data, because the categorical features are with limited number of possible values, making data objects overlap in the distance space as discussed in the Introduction.

Hence, for a categorical data set, it is necessary to explore a series of suitable numbers of clusters $\kappa = \{k_1, k_2, ..., k_\sigma\}$ where k_σ is the one corresponding to the partition of data objects with the most coarse granularity. As existing competitive learning algorithms aim to find k^* only, we proposed MGCPL to find all the suitable ks in κ. The basic idea is to start the competitive learning with a relatively large initial k_0, and let the k_0 clusters compete with each other to eliminate less important ones to obtain k_1. By inheriting the previously learned k_1 as the initialization, the learning is relunched by clearing the parameters that guide the convergence. Such a process is recursively implemented until the overall MGCPL converges at k_σ where the coarse-grained clusters are prominent enough and no more clusters can be learned to eliminate.

The competitive learning mechanism described by Eq. (3)-Eq. (6) only awards the winning cluster while neglecting the rival cluster, which makes the winners gradually absorb the surrounding seed points and thus not conducive to exploring multi-granular clusters. To avoid this, a rival penalization mechanism is introduced. Specifically, for each input x_i, the winning cluster C_v selected from the initialized cluster candidates is updated toward x_i, while the rival nearest C_h to the winner C_v is determined by

$$h = \arg \max_{1 \leq l \leq k, l \neq v} \left[(1 - \rho_l) u_l s\left(x_i, C_l\right) \right] \tag{9}$$

where ρ_l is defined in Eq. (7). For each data object x_i, when the winning cluster and its rival nearest are determined, x_i will be assigned to the winning cluster C_v, and the corresponding winning time is updated by

$$g_v = g_v + 1. \tag{10}$$

In Eq. (9), u_l is the weight of C_l computed by

$$u_l = \frac{1}{1 + e^{(-10\delta_l + 5)}} \tag{11}$$

with $l \in \{1, 2, ..., k\}$. Such a commonly used Sigmoid function form is to ensure a more sensitive updating of rival weights and make its values in the interval [0,1]. Accordingly, the updating of u_l can be accomplished by changing the value of δ_l instead. Subsequently, the winner C_v is awarded with

$$\delta_v^{new} = \delta_v^{old} + \eta \tag{12}$$

and the rival C_h is penalized with

$$\delta_h^{new} = \delta_h^{old} - \eta s(x_i, C_l) \tag{13}$$

where η is a small learning rate. As a result, the rival is penalized a step away from the winner, and thus the rivals obtain more opportunities to explore the cluster distributions in the distance space.

It is usually assumed that all the categorical features have the same contribution during the object-cluster similarity measurement. But in practice, as features are of different importance in forming clusters of different data sets, we improve Eq. (1) with a weighting mechanism by

$$s\left(x_i, C_l\right) = \frac{1}{d} \left[\sum_{r=1}^{d} \omega_{rl} s(x_{ir}, C_l) \right] \tag{14}$$

where ω_{rl} with $0 \leq \omega_{rl} \leq 1$ is the weight of the rth feature to cluster C_l, and we have $\sum_{r=1}^{d} \omega_{rl} = 1$ with $l \in \{1, 2, ..., k\}$. Since feature-weight ω_{rl} is changing with the change of feature-cluster contribution, we use H_{rl} to indicate the contribution of feature F_r to cluster C_l. To compute H_{rl}, we should first introduce two important terms, i.e., inter-cluster difference α_{rl} and intra-cluster similarity β_{rl}, where α_{rl} measures the ability of feature F_r in distinguishing cluster C_l from the others, while β_{rl} evaluates whether the cluster C_l along the feature F_r has a compact structure. We formulate α_{rl} by

$$\alpha_{rl} = \frac{1}{\sqrt{2}} \sqrt{\sum_{t=1}^{m_r} \left(\frac{\Psi_{F_r = f_{rt}}(C_l)}{\Psi_{F_r \neq NULL}(C_l)} - \frac{\Psi_{F_r = f_{rt}}(X \setminus C_l)}{\Psi_{F_r \neq NULL}(X \setminus C_l)} \right)^2} \tag{15}$$

and calculate β_{rl} by

$$\beta_{rl} = \frac{1}{n_l} \sum_{x_i \in C_l} \frac{\Psi_{F_r = x_{ir}}(C_l)}{\Psi_{F_r \neq NULL}(C_l)} \tag{16}$$

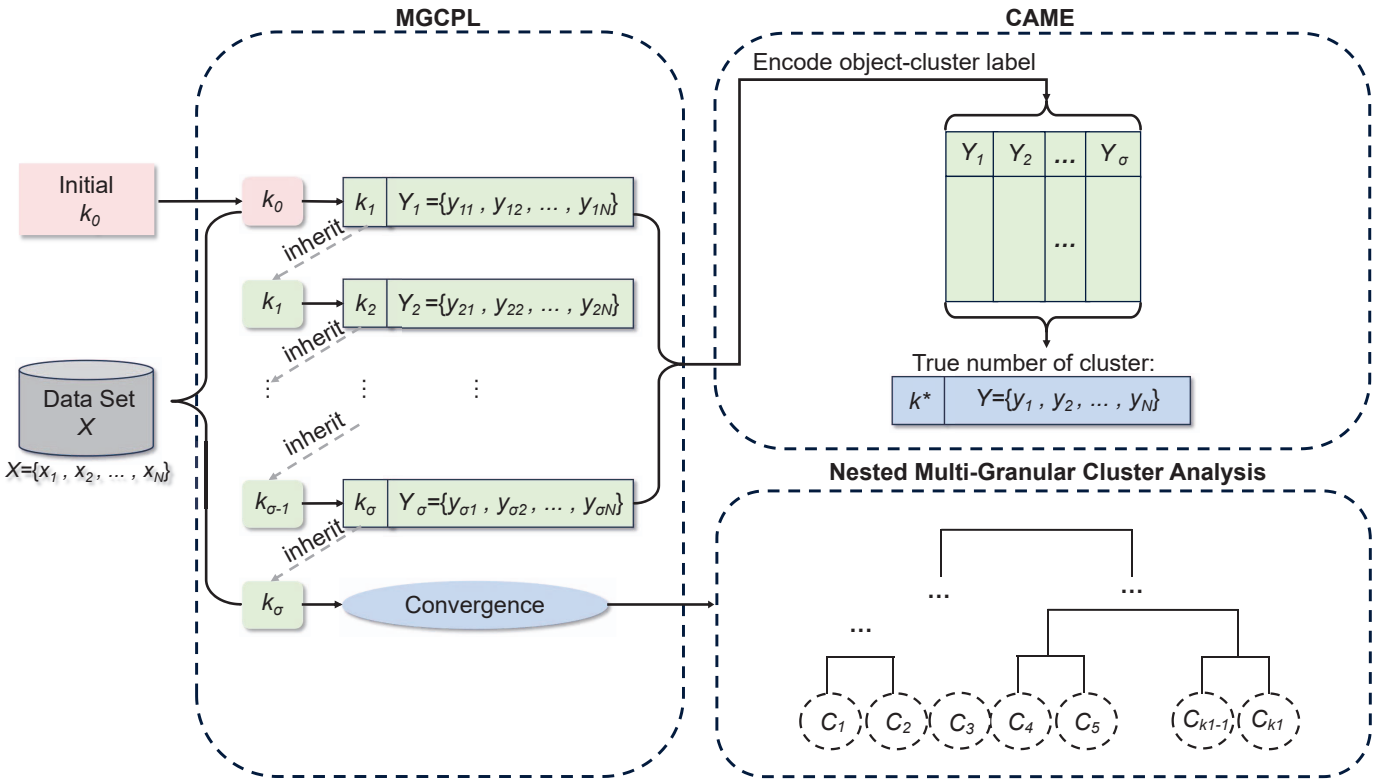


Fig. 3: Pipeline of the proposed method. MGCPL starts its learning with a relatively large initial k_0. The initialized clusters compete with each other to eliminate less important ones and obtain k_1. By inheriting k_1 as the initialization, the learning is re-lunched by clearing the parameters that guide the convergence. Such a process is recursively implemented until converges at k_σ where the k_σ prominent coarse-grained clusters cannot be further eliminated. The multi-granular results can be utilized for nested cluster distribution analysis, and can also be aggregated by CAME to accurately partition X into k clusters.

where n_l is the number of objects in C_l. When both α_{rl} and β_{rl} reach large values, it implies the important contribution of feature F_r in detecting C_l, and thus H_{rl} can be obtained by

$$H_{rl} = \alpha_{rl}\beta_{rl} \tag{17}$$

accordingly. Then the corresponding probabilistic feature weight ω_{rl} can be calculated by

$$\omega_{rl} = \frac{H_{rl}}{\sum_{t=1}^{d} H_{tl}} \tag{18}$$

with $r \in \{1, 2, ..., d\}$ and $l \in \{1, 2, ..., k\}$.

To facilitate the learning of multi-granular clusters by using the above-described learning process, we initialize a larger number of clusters k_0 to launch the learning. When the above-defined competitive penalization learning converges with k_1, i.e., an appropriate number of clusters at a fine granularity corresponding to k_1 have been explored, we let the learning mechanism inherit k_1 and re-launch the learning to explore coarser-grained clusters. To re-launch the learning in a new epoch, all the previous statistics are reset by $g_l = 0$, $u_l = 1/d$, and $\delta_l = 1$ with $l = \{1, 2, ..., k_\sigma\}$. Competitive penalization learning is recursively launched until it obtains the same partition as the previous epoch. In this way, we can obtain a series of numbers of clusters with decreasing values $\kappa = \{k_1, k_2, ..., k_\sigma\}$ where k_σ is the obtained k with the smallest value. At the same time, we obtain a series of partitions, i.e., clustering results, which can be represented as a collection of object labels, i.e., $\Gamma = \{Y_1, Y_2, ..., Y_\sigma\}$ with $Y_\sigma = \{y_{\sigma1}, y_{\sigma2}, ..., y_{\sigma n}\}$. The whole MGCPL algorithm is summarized as Algorithm 1.

B. CAME: Cluster Aggregation based on MGCPL Encoding

The multi-granular cluster distribution information obtained by MGCPL can be utilized to form an informative representation of categorical data. We thus propose a new encoding strategy that uses the object-cluster affiliation Γ as the data representation. Then, we implement categorical data clustering on the new representation. The advantage of Γ encoding is that it can make full use of the information provided by each granularity of the data set and convert the heterogeneous information provided by the features from different domains into the object-cluster affiliation learned by MGCPL. Moreover, since the features in Γ provide object-cluster affiliations at different granularities, their contributions to the final clustering of CAME are usually different in terms of the sought number of clusters k. Therefore, we formulate the cluster aggregation in the form of feature importance learning to minimize the

Algorithm 1 MGCPL: Multi-Granular Competitive Penalization Learning Algorithm

Input: Data set X, learning rate η, initialized k_0.
Output: Multi-granular partitions $\Gamma = \{Y_1, Y_2, ..., Y_\sigma\}$ and corresponding numbers of clusters $\kappa = \{k_1, k_2, ..., k_\sigma\}$.

Initialize convergence = false, $k^{initial} = k_0$.
while convergence = false **do**
  change = true, randomly select $k^{initial}$ objects to represent clusters in C.
  while change = true **do**
    for i = 1 to n **do**
      Compute v and h by Eqs. (6) and (9), update q_{iv} by Eq. (4), update learning variables by Eqs. (7) and (10)-(13).
    end for
    if $Q^{new} = Q^{old}$ **then**
      change = false.
    end if
    Update ω_{rl} by Eq. (15)-(18).
  end while
  Set $k^{initial} = k^{new}$, reset g_l = 0, u_l = 1/d and δ_l = 1 with $l = \{1, 2, ..., k^{new}\}$.
  if $k^{new} = k^{old}$ **then**
    convergence = true.
  end if
end while

objective function $P(Q, \Theta)$ as follows

$$P(Q, \Theta) = \sum_{l=1}^{k} \sum_{i=1}^{n} \sum_{r=1}^{\sigma} q_{il} \theta_r d(x_{ir}, Z_{lr}) \tag{19}$$

where q_{il} is the (i, l)th entry of Q defined as

$$q_{il} = \begin{cases} 1, \text{ if } \sum_{r=1}^{\sigma} \theta_r d(x_{ir}, Z_{lr}) \leq \sum_{r=1}^{\sigma} \theta_r d(x_{ir}, Z_{tr}) \\ \qquad\qquad\qquad\qquad \text{for } \forall\ t \in \{1, 2, ..., k\} \\ 0, \text{ otherwise.} \end{cases} \tag{20}$$

In Eq. (19), the variable $\Theta = \{\theta_1, \theta_2, ..., \theta_\sigma\}$ is a set of feature weights to be updated during learning. Z_l represents the mode of lth cluster with $l = \{k_1, k_2, ..., k_\sigma\}$. $s(x_{ir}, Z_{lr})$ is the Hamming distance between feature value of object x_{ir} and feature value of cluster Z_{lr}. Note that we use Z_{lr} here to indicate that this is the cluster mode value from representation Γ rather than the original data set X. In this subsection, all the data values are from Γ.

The weight θ_r that reflects the importance of F_r in Γ is updated by

$$\theta_r = \frac{I_r}{\sum_{t=1}^{\sigma} I_r} \tag{21}$$

where I_r is the overall intra-cluster similarity contributed by F_r, which can be written as

$$I_r = \sum_{l=1}^{k} \sum_{i=1}^{n} \sum_{r=1}^{\sigma} \left[1 - d(x_{ir}, Z_{lr})\right]. \tag{22}$$

Algorithm 2 CAME: Cluster Aggregation based on MGCPL Encoding

Input: Data representation Γ, number of clusters k.
Output: Partition Q, features importance Θ.

Initialize convergence = false and $\theta_r = 1/\sigma$ with $r = \{1, 2, ..., \sigma\}$.
Compute Q according to Eq. (20).
while convergence = false **do**
  Set $Q = \tilde{Q}$, compute $\tilde{\Theta}$ by Eqs. (21) and (22).
  Set $\Theta = \tilde{\Theta}$, compute $\tilde{Q}$ by Eq. (20).
  if $Q = \tilde{Q}$ **then**
    convergence = true.
  end if
end while

A higher intra-cluster similarity of a feature indicates that this feature contributes more to forming clusters with more similar data objects.

Clustering with the above feature weighting can be treated as an optimization problem to minimize Eq. (19). More specifically, we can iteratively solve the following two minimization problems:

1) Fix object partition $Q = \tilde{Q}$, update feature weights $\tilde{\Theta}$;
2) Fix feature weights $\Theta = \tilde{\Theta}$, compute object partition $\tilde{Q}$.

Such a learning process will converge to a minimal solution in a finite number of iterations, and the final clustering result Q can be obtained. We summarize CAME as Algorithm 2.

C. Time Complexity Analysis

Theorem 1. *The time complexity of MGCPL is* $O(dnk_0)$.

Proof. *To analyze the complexity in the worst case, we adopt* k_0 *as the initial* k, I *is the maximum number of iterations to make the competitive penalization learning converge. During the object-cluster similarity computation, as* $n \times k_0$ *pairs of distances should be computed on* d *features, the time complexity is thus* $O(Idnk_0)$ *for similarity computation. Similarly, there are* $d \times k_0$ *weights in total that should be updated based on the statistics obtained by going through all the* n *data objects, and thus the time complexity for weights updating is* $O(dnk_0)$. *As for the updating of* g_l, u_l, *and* δ_l, *their time complexity can be omitted compared to that of similarity computation and weights updating. Since the above parts will be implemented by* σ *times in Algorithm 1, the overall time complexity of MGCPL is* $O(\sigma Idnk_0)$. *As* σ *and* I *are both much smaller than* n, d, *and* k_0 *in practice, the overall time complexity of MGCPL is thus* $O(dnk_0)$. □

Theorem 2. *The time complexity of CAME is* $O(dnk)$.

Proof. *Assume that the clustering process of CAME needs* T *iterations to converge. In each iteration, weights of* σ *features are updated by considering the* n *data objects in* k *clusters, and thus the time complexity of feature weighing is* $O(dnk)$. *In each iteration, all the* n *data objects should also be partitioned into* k *clusters by computing the object-cluster*

distances reflected by σ features, and thus the time complexity is also $O(dnk)$. For T iterations in total, the overall time complexity is $O(Tdnk)$. Since T can be viewed as a small constant in most cases, the overall time complexity of CAME is thus $O(dnk)$. □

D. Discussions on Convergence and Distributed Computing

The proposed whole clustering approach MCDC is composed of MGCPL in Algorithm 1 and CAME in Algorithm 2. The MGCPL component can be viewed as repeatedly implementing competitive penalization learning [38], which is a strict gradient decent process that is guaranteed to converge. For the CAME component, it is actually a process of features weighted k-modes clustering, which has also been proven to converge in [21]. Although we adopt an approximation to more intuitively update the weights by Eq. (21), such an update strategy is still consistent with the minimization of the objective function in Eq. (19), as features that contribute less on the minimization of the objective function are assigned with smaller importance in the next iteration. Therefore, MCDC well converges on all the data sets we used for the experiments. If strict convergence is required in some scenarios, the weights updating mechanism described by Eqs. (21) and (22) can be simply replaced by the updating strategy described in [21] derived via Lagrange multiplier.

The potential contributions of the proposed multi-granular clustering algorithm to distributed computing systems are mainly two-fold:

1) It can be utilized to pre-partition data points into compact subsets to more reasonably allocate them to distributed computing nodes. Specifically, data points described by categorical features are automatically divided into relatively independent and compact micro-clusters, which are automatically merged into larger-scale clusters of different granularities. The multi-granular information obtained in this process can well guide the central server to allocate data sample subsets of different granularities to suitable nodes, flexibly realizing parallel computing without causing significant loss of local correlation information of the data objects.
2) It can be utilized to pre-divide compute nodes described as the data set shown in Fig. 1 to form performance-consistent node networks that are more suitable for certain computing tasks. That is, computing nodes are automatically grouped into multi-granular clusters according to their categorical features. The nodes in the same cluster have relatively consistent computing performance and features, and can thus collaborate more efficiently to complete distributed computing tasks. Therefore, the obtained multi-granular computing node clusters can flexibly guide the selection of uniform nodes according to computing task requirements.

IV. Experiment

This section introduces the experimental design and the selection of counterparts, validity indices, and data sets. Then five parts of experimental results are demonstrated with in-depth discussions for performance evaluation of the proposed MGCPL-guided Categorical Data Clustering (MCDC).

A. Experimental Settings

Five Experiments are conducted to evaluate the proposed method from different perspectives, which are summarized below.

- Clustering performance evaluation: The proposed MCDC method is compared with existing representative clustering approaches by quantifying their clustering performance using different mainstream validity indices.
- Significance test: Wilcoxon signed rank test is conducted on the performance of the compared approaches. A rejection of the null hypothesis indicates a significant outperforming of our proposed method against the counterparts.
- Ablation Study: MCDC is ablated into different versions by successively removing its main technical components, and the performance of these versions is compared to illustrate the effectiveness of the MCDC components.
- Learning process evaluation: MGCPL will converge to different ks during its learning. We visualize the changing of k with the optimal k^* to illustrate the effectiveness of the multi-granular cluster learning mechanism.
- Computational efficiency evaluation: MGCPL can be viewed as an efficient alternative to hierarchical clustering. We thus plotting and comparing its execution time under different ns, ds, and ks with the counterparts.

Nine Counterparts are compared in the comparative experiments, including six representative clustering methods, two variants of MCDC that adopt and enhance two existing categorical data clustering algorithms, and MCDC itself. The six representative counterparts are the conventional k-modes [14] proposed for partitional clustering and ROCK [43] for hierarchically clustering categorical data. Four recent advanced clustering methods, i.e., WOCIL [38], GUDMM [30], FKMAWCW [36], and ADC [39] are also chosen for more convincing comparison. WOCIL is proposed for automatically learning the clusters of mixed data, i.e., data composed of both categorical and numerical features. GUDMM introduces a generalized multi-aspect distance measure based on mutual information. FKMAWCW is a fuzzy k-modes-based approach that learns weights of features to clusters during clustering. ADC utilizes a graph-based dissimilarity measurement for cluster analysis of data composed of any type of features. GUDMM and FKMAWCW are also applied to the multi-granular encoding output of our proposed MCDC. The formed clustering approaches are named MCDC+GUDMM and MCDC+FKMAWCW, respectively. For simplicity, they are abbreviated as MCDC+G. and MCDC+F. hereinafter. Hyper-parameters of the compared methods (if any) are set according to the corresponding source paper. Learning rate η and k_0 of MCDC is set at $\eta = 0.03$ and $k_0 = \sqrt{n}$, respectively. For all the compared methods, the sought number of clusters is set at k^* corresponding to each data set as shown in Table II.

TABLE II: Statistics of the 8 data sets.d, n, and k^* indicate the number of features, the number of objects, and the true number of clusters, respectively.

No.	Data Set	Abbrev.	d	n	k^*
1	Car Evaluation	Car.	6	1728	4
2	Congressional	Con.	16	435	2
3	Chess	Che.	36	3196	2
4	Mushroom	Mus.	22	8124	2
5	Tic Tac Toe	Tic.	9	958	2
6	Vote	Vot.	16	232	2
7	Balance	Bal.	4	625	3
8	Nursery	Nur.	8	12960	5
9	Synthetic (with large n)	Syn_n	10	200000	3
10	Synthetic (with large d)	Syn_d	1000	20000	3

Four Validity Indices[1] are utilized to measure the clustering performance. Clustering Accuracy (ACC) is a commonly used index that ranges from 0 to 1. It computes the ratio of the number of correctly clustered objects to the total number of objects. Adjusted Rand Index (ARI) calculates the consistency of obtained clustering results and true labels by comparing their pairwise matching. Its values range from -1 to 1. Adjusted Mutual Information (AMI) is based on mutual information, which quantifies the matching between obtained clustering results and true labels from the perspective of information theory. Its values also range from -1 to 1. The Fowlkes-Mallows (FM) score is defined as the geometric mean of the pairwise precision and recall, and ranges from 0 to 1. All the adopted indices reflect a better clustering performance with a higher value.

Ten Data Sets are utilized to conduct a comprehensive evaluation. Among them, eight data sets are representative ones downloaded from the UCI Machine Learning Repository[2]. Two synthetic data sets with large n and d are generated with well-separated clusters for the efficiency evaluation. All the data sets are categorical ones, and data objects with missing values are omitted before conducting experiments. Detailed statistics of the data sets are shown in Table II.

B. Clustering Performance Evaluation

The clustering performance of the proposed MCDC is compared with the counterparts on all eight categorical data sets in Table III. Each result in the table is obtained by executing the corresponding method by 50 times and taking the average performance with standard deviation. Please note that MCDC+G. and MCDC+F. are the two variants of MCDC adopting GUDMM and FKMAWCW as the clustering algorithms, respectively. Comparing their performance with the original GUDMM and FKMAWCW can reflect the effectiveness of the proposed MCDC in enhancing the clustering performance of existing clustering methods. In the table, the best and the second-best results on each data set are highlighted using **boldface** and underline, respectively.

[1]https://scikit-learn.org/stable/modules/clustering.html#clustering-evaluation

[2]https://archive.ics.uci.edu/

It can be observed from Table III that MCDC outperforms its counterparts on most data sets. Although MCDC does not perform the best on some data sets, it still ranks second or with a performance that is very close to the best performer. Thus, it can be concluded that, in general, MCDC demonstrates its superiority in terms of both accuracy and robustness. From the table, we can also see that ROCK, FKMAWCW, and GUDMM have unsatisfactory performance on some data sets. This is because they sometimes cannot obtain the preset number of clusters and are judged as failed. Moreover, ROCK, WOCIL, and three MCDC variants perform very stable because Rock is a hierarchical clustering approach without random initialization, and WOCIL adopts a very stable initialization mechanism. The performance of MCDC is also very stable because the learned multi-granular information complements each other to form a comprehensive and stable representation of different data sets.

As for MCDC+G. and MCDC+F., it can be seen that the performance of the corresponding GUDMM and FKMAWCW is obviously boosted in most cases by MCDC. This indicates that the proposed MCDC is effective in enhancing different categorical data clustering methods. It can be seen that MCDC, MCDC+G., and MCDC+F. achieve obviously better clustering performance than the other counterparts. Moreover, MCDC+F. performs the best in general. This may be because the corresponding FKMAWCW is a fuzzy clustering algorithm that suits categorical data better. More specifically, fuzzy algorithms can more appropriately describe the object-cluster similarity based on the statistics during clustering, and can thus better exploit the multi-granular information in the embeddings provided by MCDC.

C. Significance Test

To provide statistical evidence of the superiority of MCDC, we conduct a significance test using the Wilcoxon signed-rank test based on the clustering performance shown in Table III, and demonstrate the test results in Table IV. The best-performing version of MCDC, i.e., MCDC+F., is compared with each of the counterparts at a 90% confidence interval. We use "+" to indicate a rejection of the null hypothesis, which means MCDC+F. significantly outperforms the corresponding counterpart w.r.t. certain validity index.

It can be seen from Table IV that MCDC significantly outperforms its counterparts under almost all the indices, which obviously illustrates the superiority of MCDC. Although the test does not show a significant advantage of MCDC in comparison with K-MODES and ROCK in terms of AMI, MCDC+F. still outperforms them on most data sets as shown in Table III.

D. Ablation Study

To further verify the effectiveness of different main components of the proposed MCDC method, we ablate it into the following four versions: 1) MCDC[4] is the version that replaces the feature weighting mechanism (described by Eqs. (21)-(22) in Section III-B) of CAME with fixed identical weights, 2)

TABLE III: Clustering performance w.r.t., ACC, ARI, AMI, and FM on categorical data sets. MCDC+G. and MCDC+F. are the variants of MCDC adopting GUDMM and FKMAWCW, respectively. The best and second-best results on each data set are highlighted using **boldface** and underline, respectively.

Index	Data	K-MODES	ROCK	WOCIL	FKMAWCW	GUDMM	ADC	MCDC	MCDC+G.	MCDC+F.
ACC	Car.	0.372±0.00	0.326±0.00	0.270±0.00	0.371±0.00	0.372±0.00	0.361±0.00	0.373±0.00	0.270±0.00	**0.414±0.00**
	Con.	0.866±0.00	0.506±0.00	**0.874±0.00**	0.796±0.01	0.818±0.00	**0.874±0.00**	**0.874±0.00**	**0.874±0.00**	**0.874±0.00**
	Che.	0.551±0.00	0.505±0.00	0.531±0.00	0.561±0.00	0.554±0.00	0.548±0.00	0.578±0.00	0.547±0.00	**0.585±0.00**
	Mus.	0.740±0.02	0.509±0.00	0.678±0.00	0.000±0.00	0.501±0.00	0.752±0.02	0.710±0.00	0.613±0.00	**0.784±0.00**
	Tic.	0.557±0.00	**0.674±0.00**	0.526±0.00	0.538±0.00	0.507±0.00	0.535±0.00	0.602±0.00	0.642±0.00	0.646±0.00
	Vot.	0.869±0.00	0.500±0.00	0.888±0.00	0.778±0.01	0.828±0.00	0.888±0.00	**0.905±0.00**	**0.905±0.00**	**0.905±0.00**
	Bal.	0.448±0.00	0.496±0.00	0.419±0.00	0.463±0.00	0.000±0.00	0.442±0.00	0.464±0.00	0.453±0.00	**0.506±0.00**
	Nur.	0.332±0.00	0.000±0.00	0.239±0.00	0.315±0.00	0.000±0.00	0.337±0.00	0.340±0.00	0.305±0.00	**0.432±0.00**
ARI	Car.	0.027±0.00	0.023±0.00	0.001±0.00	-0.002±0.00	**0.054±0.00**	0.017±0.00	0.051±0.00	0.001±0.00	0.027±0.00
	Con.	0.536±0.00	-0.004±0.00	**0.557±0.00**	0.385±0.05	0.394±0.00	**0.557±0.00**	**0.557±0.00**	**0.557±0.00**	**0.557±0.00**
	Che.	0.014±0.00	-0.001±0.00	0.003±0.00	0.020±0.00	0.012±0.00	0.015±0.00	0.024±0.00	0.008±0.00	**0.028±0.00**
	Mus.	0.303±0.07	-0.001±0.00	0.125±0.00	0.000±0.00	-0.003±0.00	0.321±0.06	0.186±0.01	0.051±0.00	**0.323±0.00**
	Tic.	0.017±0.00	**0.120±0.00**	0.000±0.00	-0.002±0.00	-0.001±0.00	0.007±0.00	0.038±0.00	0.079±0.00	0.062±0.00
	Vot.	0.543±0.00	-0.004±0.00	0.600±0.00	0.349±0.05	0.427±0.00	0.600±0.00	**0.655±0.00**	**0.655±0.00**	**0.655±0.00**
	Bal.	0.027±0.00	**0.080±0.00**	0.005±0.00	0.055±0.00	0.000±0.00	0.025±0.00	0.052±0.00	0.016±0.00	0.079±0.00
	Nur.	0.049±0.00	0.000±0.00	0.002±0.00	0.028±0.00	0.000±0.00	0.052±0.00	0.051±0.00	0.004±0.00	**0.166±0.00**
AMI	Car.	0.049±0.00	0.050±0.00	0.003±0.00	0.082±0.00	0.117±0.00	0.047±0.00	**0.123±0.00**	0.003±0.00	0.015±0.00
	Con.	0.473±0.00	0.001±0.00	**0.484±0.00**	0.337±0.03	0.380±0.00	**0.484±0.00**	**0.484±0.00**	**0.484±0.00**	**0.484±0.00**
	Che.	0.012±0.00	0.000±0.00	0.003±0.00	**0.021±0.00**	0.011±0.00	0.015±0.00	0.020±0.00	0.005±0.00	0.020±0.00
	Mus.	0.280±0.05	0.000±0.00	0.235±0.00	0.000±0.00	0.044±0.00	**0.347±0.04**	0.209±0.01	0.036±0.00	0.248±0.00
	Tic.	0.012±0.00	**0.120±0.00**	0.007±0.00	0.005±0.00	0.000±0.00	0.006±0.00	0.020±0.00	0.058±0.00	0.023±0.00
	Vot.	0.457±0.00	0.000±0.00	0.522±0.00	0.301±0.03	0.417±0.00	0.522±0.00	**0.566±0.00**	**0.566±0.00**	**0.566±0.00**
	Bal.	0.026±0.00	0.071±0.00	0.008±0.00	0.048±0.00	0.000±0.00	0.026±0.00	0.083±0.00	0.017±0.00	**0.089±0.00**
	Nur.	0.060±0.00	0.000±0.00	0.004±0.00	0.043±0.00	0.000±0.00	0.061±0.00	0.077±0.00	0.022±0.00	**0.208±0.00**
FM	Car.	0.409±0.00	0.394±0.00	0.369±0.00	0.406±0.00	0.413±0.00	0.401±0.00	0.407±0.00	0.369±0.00	**0.434±0.00**
	Con.	0.774±0.00	0.518±0.00	**0.784±0.00**	0.711±0.01	0.754±0.00	**0.784±0.00**	**0.784±0.00**	**0.784±0.00**	**0.784±0.00**
	Che.	0.544±0.00	0.525±0.00	0.507±0.00	**0.578±0.00**	0.554±0.00	0.555±0.00	0.573±0.00	0.519±0.00	0.532±0.00
	Mus.	0.667±0.02	0.525±0.00	0.657±0.00	0.000±0.00	0.687±0.00	**0.721±0.01**	0.640±0.00	0.544±0.00	0.662±0.00
	Tic.	0.538±0.00	0.581±0.00	0.527±0.00	0.547±0.00	0.524±0.00	0.526±0.00	0.548±0.00	0.562±0.00	**0.612±0.00**
	Vot.	0.772±0.00	0.500±0.00	0.800±0.00	0.696±0.01	0.734±0.00	0.800±0.00	**0.827±0.00**	**0.827±0.00**	**0.827±0.00**
	Bal.	0.426±0.00	0.441±0.00	0.437±0.00	0.424±0.00	0.000±0.00	0.425±0.00	**0.464±0.00**	0.460±0.00	0.452±0.00
	Nur.	0.303±0.00	0.000±0.00	0.260±0.00	0.306±0.00	0.000±0.00	0.305±0.00	0.309±0.00	0.321±0.00	**0.396±0.00**

TABLE IV: Results of two-tailed Wilcoxon signed-rank test conducted with confidence interval 90% (i.e. α = 0.1). The symbol "+" indicates that MCDC+F. performs significantly better than a certain counterpart, while "-" indicates that there is no significant difference between the two methods.

Method	ACC	ARI	AMI	FM
K-MODES	+	+	-	+
ROCK	+	+	-	+
WOCIL	+	+	+	+
FKMAWCW	+	+	+	+
GUDMM	+	+	+	+
ADC	+	+	-	-

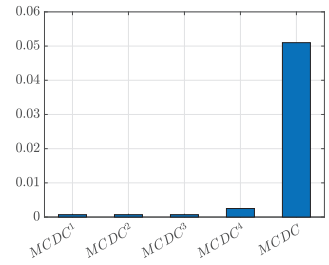
MCDC^3 is obtained by removing the whole CAME module from MCDC and use the k_σ learned by MGCPL for clustering, 3) MCDC^2 is the version obtained by replacing MGCPL of MCDC^3 with the conventional competitive learning with $k^* + 2$ as the initialization described in Section II-B, and 4) MCDC^1 is the version formed by further removing the competitive learning mechanism from MCDC^2 and only adopt the object-cluster distance described in Section II-A. Since the version of MCDC^1 that replaces object-cluster distance with the conventional Hamming distance metric is equivalent to k-modes, which has been compared in Table III, We do not further ablate MCDC^1 to avoid duplicated results.

It can be seen from the results shown in Fig. 4 that the ARI performance of MCDC, MCDC^4, MCDC^3, MCDC^2, MCDC^1 sequentially decreases in general, which intuitively illustrate the effectiveness of all the proposed main technical components of MCDC.

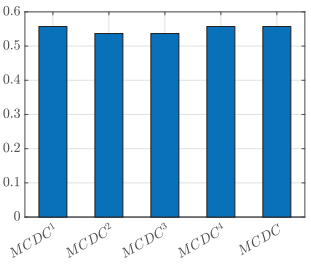
More specifically, it can be observed that MCDC always outperforms MCDC^4. This indicates that the feature weighting mechanism in CAME (i.e., Eqs. (21)-(22) in Section III-B) is effect in learning the importance of features in the embeddings output by CAME, and also illustrates that the encoding strategy of CAME is effective in fusing the multi-granular information provided by MGCPL.

It can also be observed that MCDC^4 performs not worse than MCDC^3 on five data sets, but outperformed by MCDC^3 on three data sets, i.e., Mus., Vot., and Bal. This is because identical weights of MCDC^4 cannot appropriately reflect the importance of the encoded features of these three data sets, which again highlights the necessity of weights learning.

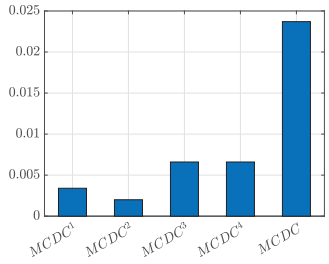
The effectiveness of the proposed MGCPL is illustrated by the fact that MCDC^3 outperforms MCDC^2 on almost all the data sets. The reason is that MGCPL learns the cluster distributions from different ks. Although the result of MCDC^3



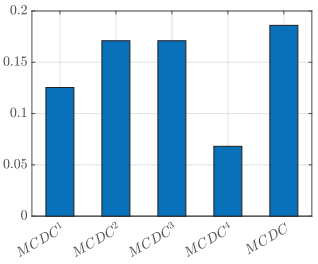
(a) ARI on Car.



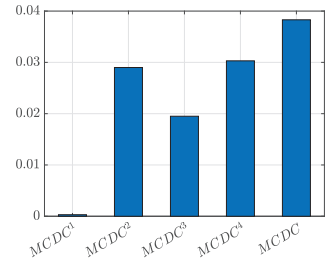
(b) ARI on Con.



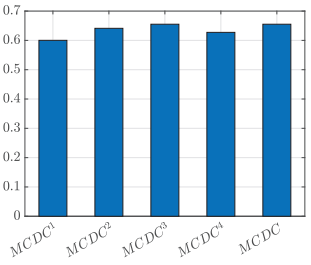
(c) ARI on Che.



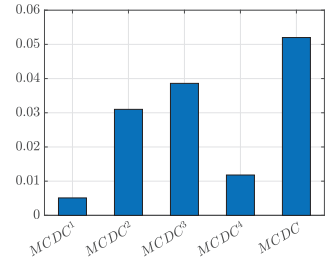
(d) ARI on Mus.



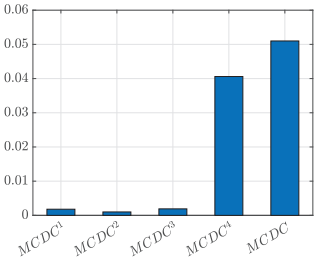
(e) ARI on Tic.



(f) ARI on Vot.

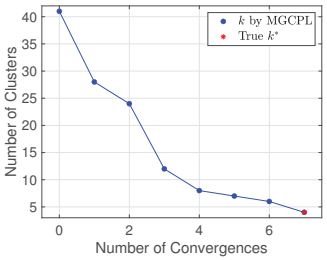


(g) ARI on Bal.

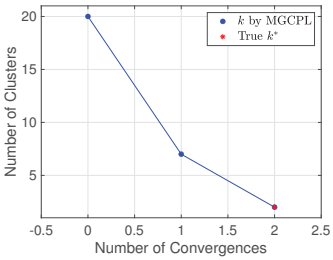


(h) ARI on Nur.

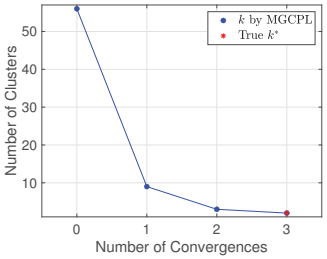
Fig. 4: Comparison of MCDC and its four ablated versions, i.e., $MCDC^4$, $MCDC^3$, $MCDC^2$, and $MCDC^1$, which are obtained by removing the weighting mechanism of CAME, the whole CAME, multi-granular learning mechanism of MGCPL, and the whole MGCPL from MCDC in turn.



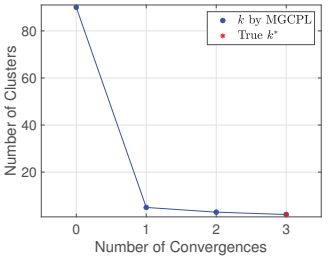
(a) ks learned for Car.



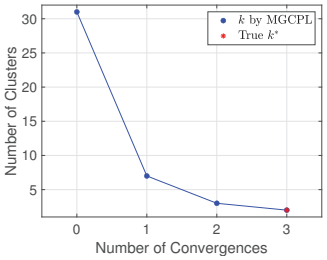
(b) ks learned for Con.



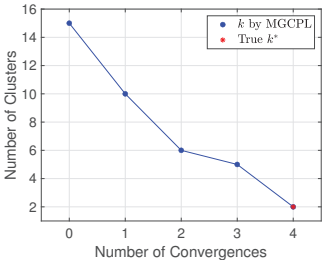
(c) ks learned for Che



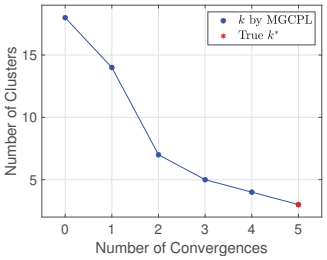
(d) ks learned for Mus.



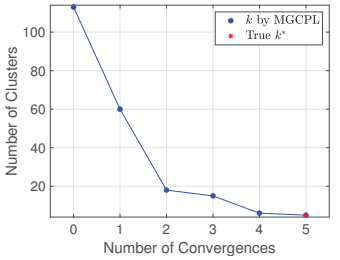
(e) ks learned for Tic.



(f) ks learned for Vot.



(g) ks learned for Bal.



(h) ks learned for Nur.

Fig. 5: Different numbers of clusters learned by MGCPL. Blue dots indicate the number of clusters when MGCPL temporarily converges under the current cluster granularity. Red stars indicate the true number of clusters k^*.

is obtained at the final k_σ, the previous $k_{\sigma-1}$ learned by MGCPL provides a more reasonable initialization for the last round learning compared to the initialized k of $MCDC^2$ where $k = k^* + 2$.

By comparing $MCDC^2$ and $MCDC^1$, it can be found that $MCDC^2$ has no significant advantage over $MCDC^1$. The reason is that $MCDC^1$ requires k^* to be given in advance for clustering, while $MCDC^2$ automatically learns to find k^*. In other words, although $MCDC^2$ adopting a competitive learning mechanism is more powerful, its advantage is obscured because k^* is unfairly leaked to $MCDC^1$.

E. Learning Process Evaluation

Numbers of clusters, i.e., $\kappa = \{k_1, k_2, ..., k_\sigma\}$, learned by MGCPL are demonstrated in Fig. 5 where blue dots indicate the number of clusters when MGCPL temporarily converges

under the current cluster granularity, and red stars indicate the true number of clusters k^* as shown in Table II. Please note that the number of clusters corresponding to "0" on the x-axis indicates the initialized k.

It can be observed that MGCPL converges in stages during its learning, which reflects that MGCPL can automatically learn clusters with different granularities. It can also be observed that almost all the final k_σ learned by MGCPL equal to the true number of clusters k^*, which indicates that MGCPL is competent in searching for the optimal number of clusters k^* without prior clustering knowledge.

F. Computational Efficiency Evaluation

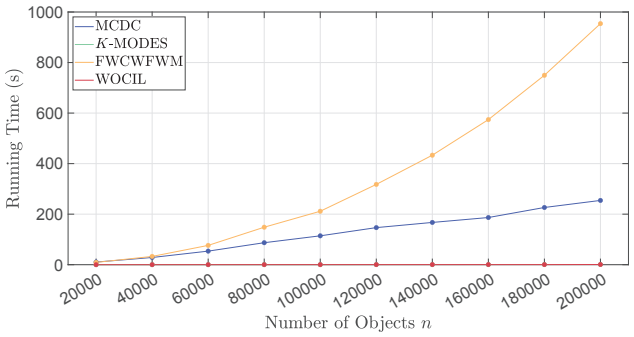
The execution time of MCDC on three synthetic data sets is shown in Fig. 6. We implement several representative counterparts on each synthetic data set with different ns, ks, and ds to verify the time complexity of MCDC. Note that k in this experiment is the number of sought clusters k in Algorithm 2. The execution time is averaged on ten runs of the corresponding method.

Intuitively, the execution time of MCDC increases linearly with the increasing of data size n and feature scale d, which confirms our analysis at the end of Section III-B that MCDC is with linear time complexity w.r.t. n and d. Moreover, it can also be observed that MCDC has linear time complexity w.r.t. k, which indicates that MCDC can be easily applied to different clustering tasks of customized k. In general, MCDC is scalable to large-scale categorical data.

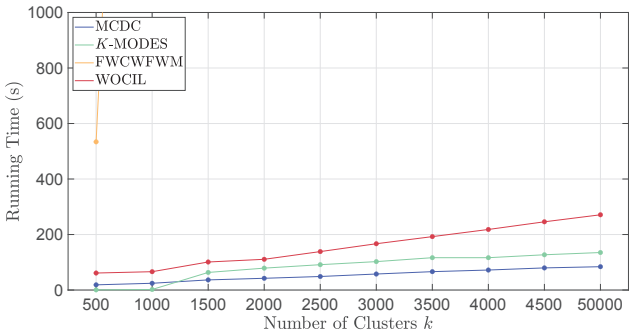
V. Concluding Remarks

This paper proposes a new method called MCDC for cluster analysis of categorical data. MCDC is composed of MGCPL for learning nested multi-granular cluster distribution, and CAME for aggregating the learned nested distribution to obtain partitional clustering results. As the learning process of MGCPL is fully automatic and highly interpretable, the complex cluster distribution of categorical data can be intuitively revealed. Accordingly, CAME encodes the multi-granular distribution learned by MGCPL to obtain informative embeddings of data objects for clustering. Since the two main components of MCDC, i.e., MGCPL and CAME, are both with linear time complexity, MCDC is scalable to large-scale categorical data. Extensive experiments illustrate the superiority of MCDC in terms of clustering accuracy, robustness to data sets in different fields, and computational efficiency.

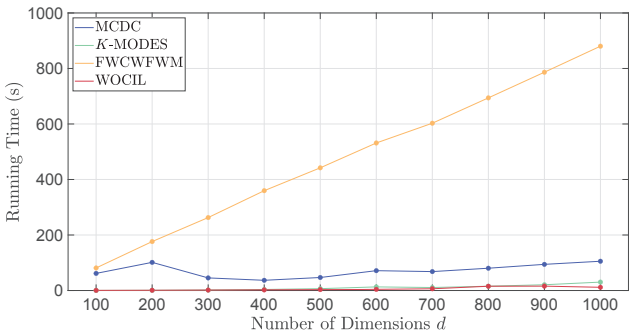
Some limitations of this work include: 1) the proposed method has not been extended to more complex heterogeneous feature data and multi-modal data and 2) we assumed that the data is static and has not yet considered more complex dynamically distributed data clustering. Building on this research, some promising future research orientations are: 1) applying MGCPL to discover implicit cluster distributions in different fields, 2) extending the whole MCDC to process streaming and dynamic data, and 3) leveraging the advantages of MGCPL to active learning for reducing the workload of human experts in manually labeling large-scale categorical data sets.



(a) Time on Syn_n w.r.t. n



(b) Time on Syn_n w.r.t. k



(c) Time on Syn_d w.r.t. d

Fig. 6: Execution time of different methods on (a) Syn_n, (b) Syn_n, and (c) Syn_d with increasing n, k, and d, respectively.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC) under grants: 62102097, 62374047, and 62174038, the NSFC/Research Grants Council (RGC) Joint Research Scheme under the grant N_HKBU214/21, the Natural Science Foundation of Guangdong Province under grants: 2023A1515012855 and 2022A1515011592, the Guangdong Provincial Key Laboratory under grant 2023B1212060076, the General Research Fund of RGC under grants: 12201321, 12202622, and 12201323, the RGC Senior Research Fellow Scheme under grant SRFS2324-2S02, and the Science and Technology Program of Guangzhou under grant 202201010548.

REFERENCES

[1] T. Li, A. Rezaeipanah, and E. M. T. El Din, "An ensemble agglomerative hierarchical clustering algorithm based on clusters clustering technique and the novel similarity measurement," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp. 3828–3842, 2022.

[2] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, pp. 165–193, 2015.

[3] J. Li, H. Izakian, W. Pedrycz, and I. Jamal, "Clustering-based anomaly detection in multivariate time series data," *Applied Soft Computing*, vol. 100, p. 106919, 2021.

[4] Z. Abbasi-Moud, H. Vahdat-Nejad, and J. Sadri, "Tourism recommendation system based on semantic clustering and sentiment analysis," *Expert Systems with Applications*, vol. 167, p. 114324, 2021.

[5] G. Caruso, S. Gattone, F. Fortuna, and T. Di Battista, "Cluster analysis for mixed data: An application to credit risk evaluation," *Socio-Economic Planning Sciences*, vol. 73, p. 100850, 2021.

[6] F. Chang, S. Yasmin, H. Huang, A. H. Chan, and M. M. Haque, "Injury severity analysis of motorcycle crashes: A comparison of latent class clustering and latent segmentation based models with unobserved heterogeneity," *Analytic Methods in Accident Research*, vol. 32, p. 100188, 2021.

[7] X.-F. Song, Y. Zhang, D.-W. Gong, and X.-Z. Gao, "A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9573–9586, 2021.

[8] Y. Yang, J. Cai, H. Yang, and X. Zhao, "Density clustering with divergence distance and automatic center selection," *Information Sciences*, vol. 596, pp. 414–438, 2022.

[9] S. Chawla and A. Gionis, "K-means–: A unified approach to clustering and outlier detection," in *Proceedings of the 2013 SIAM International Conference on Data mining*. SIAM, 2013, pp. 189–197.

[10] N. Liu, Z. Xu, X.-J. Zeng, and P. Ren, "An agglomerative hierarchical clustering algorithm for linear ordinal rankings," *Information Sciences*, vol. 557, pp. 170–193, 2021.

[11] A. Agresti, *Categorical data analysis*. John Wiley & Sons, 2012, vol. 792.

[12] R. Azen and C. M. Walker, *Categorical data analysis for the behavioral and social sciences*. Routledge, 2021.

[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*, vol. 110, p. 104743, 2022.

[14] Z. Huang, "A fast clustering algorithm to cluster very large categorical data sets in data mining," *Dmkd*, vol. 3, no. 8, pp. 34–39, 1997.

[15] R.-J. Kuo, Y. Zheng, and T. P. Q. Nguyen, "Metaheuristic-based possibilistic fuzzy k-modes algorithms for categorical data clustering," *Information Sciences*, vol. 557, pp. 1–15, 2021.

[16] F. Yuan, Y. Yang, and T. Yuan, "A dissimilarity measure for mixed nominal and ordinal attribute data in k-modes algorithm," *Applied Intelligence*, vol. 50, pp. 1498–1509, 2020.

[17] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[18] M. Alamuri, B. R. Surampudi, and A. Negi, "A survey of distance/similarity measures for categorical data," in *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2014, pp. 1907–1914.

[19] Y. Qian, F. Li, J. Liang, B. Liu, and C. Dang, "Space structure and clustering of categorical data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 10, pp. 2047–2059, 2015.

[20] 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*, vol. 31, no. 5, pp. 853–866, 2018.

[21] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, "Automated variable weighting in k-means type clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 657–668, 2005.

[22] C. Zhu, L. Cao, and J. Yin, "Unsupervised heterogeneous coupling learning for categorical representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 533–549, 2020.

[23] L. Bai and J. Liang, "A categorical data clustering framework on graph representation," *Pattern Recognition*, vol. 128, p. 108694, 2022.

[24] 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*, vol. 44, no. 7, pp. 3560–3576, 2021.

[25] 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 *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, 2022, pp. 1–8.

[26] P. Arabie, N. D. Baier, C. F. Critchley, and M. Keynes, "Studies in classification, data analysis, and knowledge organization," 2006.

[27] D. Barbará, Y. Li, and J. Couto, "Coolcat: An entropy-based algorithm for categorical clustering," in *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, 2002, pp. 582–589.

[28] T. Li, S. Ma, and M. Ogihara, "Entropy-based criterion in categorical clustering," in *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004, p. 68.

[29] Y. Zhang, Y.-M. Cheung, and K. C. Tan, "A unified entropy-based distance metric for ordinal-and-nominal-attribute data clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 1, pp. 39–52, 2019.

[30] E. Mousavi and M. Sehhati, "A generalized multi-aspect distance metric for mixed-type data clustering," *Pattern Recognition*, vol. 138, p. 109353, 2023.

[31] Y. Zhang and Y.-M. Cheung, "A new distance metric exploiting heterogeneous interattribute relationship for ordinal-and-nominal-attribute data clustering," *IEEE Transactions on Cybernetics*, vol. 52, no. 2, pp. 758–771, 2020.

[32] S. Q. Le and T. B. Ho, "An association-based dissimilarity measure for categorical data," *Pattern Recognition Letters*, vol. 26, no. 16, pp. 2549–2557, 2005.

[33] 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*, vol. 28, no. 1, pp. 110–118, 2007.

[34] 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 (TKDD)*, vol. 6, no. 1, pp. 1–25, 2012.

[35] 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*, vol. 27, no. 5, pp. 1065–1079, 2015.

[36] A. G. Oskouei, M. A. Balafar, and C. Motamed, "Fkmawcw: Categorical fuzzy k-modes clustering with automated attribute-weight and cluster-weight learning," *Chaos, Solitons & Fractals*, vol. 153, p. 111494, 2021.

[37] Y.-m. Cheung and H. Jia, "Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number," *Pattern Recognition*, vol. 46, no. 8, pp. 2228–2238, 2013.

[38] 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*, vol. 29, no. 8, pp. 3308–3325, 2017.

[39] 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*, vol. 34, no. 9, 2022.

[40] Y. Jeon, J. Yoo, J. Lee, and S. Yoon, "Nc-link: A new linkage method for efficient hierarchical clustering of large-scale data," *IEEE Access*, vol. 5, pp. 5594–5608, 2017.

[41] Y.-m. Cheung and Y. Zhang, "Fast and accurate hierarchical clustering based on growing multilayer topology training," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 876–890, 2018.

[42] A. Dogan and D. Birant, "K-centroid link: A novel hierarchical clustering linkage method," *Applied Intelligence*, pp. 1–24, 2022.

[43] S. Guha, R. Rastogi, and K. Shim, "Rock: A robust clustering algorithm for categorical attributes," *Information Systems*, vol. 25, no. 5, pp. 345–366, 2000.

[44] L. Hu, M. Jiang, Y. Liu, and Z. He, "Significance-based categorical data clustering," *ArXiv Preprint ArXiv:2211.03956*, 2022.

[45] S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton, "Competitive learning algorithms for vector quantization," *Neural Networks*, vol. 3, no. 3, pp. 277–290, 1990.