Information Sciences 690 (2025) 121588

Contents lists available at ScienceDirect

Information Sciences

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

Significance-based decision tree for interpretable categorical data clustering

Lianyu Hu, Mudi Jiang, Xinying Liu, Zengyou He *

School of Software, Dalian University of Technology, Dalian, China

ARTICLE INFO

Keywords:
Categorical data
Interpretable clustering
Multiple testing correction
Two-sample test
Unsupervised decision trees

ABSTRACT

Numerous clustering algorithms prioritize accuracy, but in high-risk domains, the interpretability of clustering methods is crucial as well. The inherent heterogeneity of categorical data makes it particularly challenging for users to comprehend clustering outcomes. Currently, the majority of interpretable clustering methods are tailored for numerical data and utilize decision tree models, leaving interpretable clustering for categorical data as a less explored domain. Additionally, existing interpretable clustering algorithms often depend on external, potentially non-interpretable algorithms and lack transparency in the decision-making process during tree construction. In this paper, we tackle the problem of interpretable categorical data clustering by growing a decision tree in a statistically meaningful manner. We formulate the evaluation of candidate splits as a multivariate two-sample testing problem, where a single p-value is derived by combining significance evidence from all individual categories. This approach provides a reliable and controllable method for selecting the optimal split while determining its statistical significance. Extensive experimental results on real-world data sets demonstrate that our algorithm achieves comparable performance in terms of cluster quality, running efficiency, and explainability relative to its counterparts.

1. Introduction

Clustering is an exploratory technique aiming to simplify complex data sets by dividing homogeneous objects into distinct groups. Numerous clustering algorithms have been developed from different perspectives, with partitional methods [1], density-based methods [2], model-based methods [3], and hierarchical methods [4] being the most widely used in practice. The output clusters are utilized across a wide range of domains, such as healthcare, management, and industry, many of which are characterized by inherently high-risk scenarios [5]. In such contexts, the involvement of experts in monitoring and ensuring accountability is essential. Consequently, making clustering algorithms interpretable is vital [6], allowing human users to thoroughly understand the decision-making processes and identify risks at any stage of the clustering procedure. Unfortunately, the development of interpretable clustering algorithms has only received limited attention recently.

Since categorical data sets are ubiquitous across different domains and widely accessible, the issue of categorical data clustering [7] has received special attention and investigation over the past decades. Essentially, most existing clustering algorithms for categorical data are developed along similar lines as their counterparts for numeric data. Similarly, the issue of interpretable categorical data

* Corresponding author.
E-mail addresses: hly4ml@gmail.com (L. Hu), zyhe@dlut.edu.cn (Z. He).

https://doi.org/10.1016/j.ins.2024.121588
Received 26 April 2024; Received in revised form 21 October 2024; Accepted 21 October 2024

Available online 26 October 2024

Table 1
SigDT vs existing interpretable clustering algorithms. CUBT is separately mentioned because it is the only interpretable categorical data clustering algorithm in the literature so far.

	Clusterability prediction	Interpretable splitting	Cluster number prediction
SigDT	✓	✓	✓
CUBT and other pre-modeling algorithms	✗	✗	✓
Post-modeling algorithms	✗	✗	✗

clustering is relatively unexplored in the literature as well. To our knowledge, only the method called CUBT in [8] is particularly designed for the purpose of producing interpretable clustering results from categorical data.

The decision tree is one of the most commonly used models for characterizing interpretable clustering results [9]. In this paper, we also adopt the decision tree as the clustering model for generating interpretable clusters, i.e., the rule along the path from the root node to each leaf node explains why samples are assigned to the corresponding cluster. To date, some tree-based interpretable clustering algorithms are already available in the literature [10,11], of which only the algorithm in [8] is directly tailored for categorical data. Regardless of the target data types, these existing interpretable clustering methods can be broadly classified into two categories: (1) In the first category of post-modeling algorithms, decision trees are constructed based on the guidance of clustering result of a third-party clustering algorithm [12,13]. (2) In the second category of pre-modeling algorithms, the clustering decision tree is directly constructed from the data sets by either optimizing a joint objective function [14,15] or utilizing a greedy tree-growth algorithm in a top-down manner [16,17].

Despite the success of existing solutions for solving the interpretable clustering issue, there are still several deficiencies that have not been resolved:

- Clusterability prediction: Some data sets may inherently lack a clustering structure (the data is composed of only one cluster). For such types of data sets, it is meaningless to conduct cluster analysis no matter whether the clustering method is interpretable or not. Hence, interpretable clustering algorithms should be capable of assessing the clusterability of the target data set, i.e., providing an "explanation" of the plausibility of dividing the data into multiple clusters.
- Interpretable splitting: The existing optimization-based algorithms can only ensure that the clustering result is interpretable in terms of decisions trees, failing to guarantee that clustering process is also explainable. That is, decision-making at each split should be trustful and interpretable as well. Such an interpretable splitting capability will certainly enhance our confidence in explaining both the clustering model and its results.
- Cluster number prediction: An interpretable algorithm should minimize the number of input parameters required for its autonomous decision-making process, aiding users in understanding the model. A key parameter often unknown is the number of clusters, which is typically hard to specify in practice. Hence, it is highly desirable to automatically determine the number of clusters in an interpretable manner.

Motivated by above observations, we present a new interpretable clustering algorithm for categorical data, which is named as SigDT (**Sig**nificance-based **D**ecision **T**ree). SigDT tackles above-mentioned challenges by introducing the statistical significance testing technique into the unsupervised decision tree construction process. As summarized in Table 1, SigDT has several advantages over existing interpretable clustering methods.

More precisely, the SigDT algorithm orchestrates the clustering process by constructing a decision tree akin to conventional ones. At each branch node, one candidate split divides current samples into two groups. Under the null hypothesis that these two groups have no difference, the candidate split assessment problem can be casted as a multiple hypothesis testing issue, where each individual test is to compare two success probabilities of each attribute value across two groups. According to the significance testing result in terms of p-values, we can either accept or reject the best candidate split by comparing its p-value with a significance level threshold.

Obviously, SigDT can alleviate those mentioned challenges faced by current interpretable clustering algorithms in a unified and elegant manner. Firstly, at the root node, if p-values of all candidate splits are larger than the significance level, then we can claim that the data set is unclusterable and it is not necessary to conduct the cluster analysis. Secondly, since the split point is chosen based on a rigorous significance testing procedure, the clustering process (tree growth process) and the splitting decision are at least explainable in a statistical sense. Finally, by specifying a significance level threshold, we can stop the tree growth procedure if p-values of all candidate splits for each leaf node cannot pass the threshold. As a result, we automatically determine the number of clusters (i.e. leaf nodes) and this number is statistically explainable.

In summary, the main contributions of our work to the field of interpretable clustering are as follows:

- We introduce the first trustworthy and understandable decision tree construction algorithm for interpretable categorical data clustering from a hypothesis testing perspective.
- Our method ensures statistical interpretability at each branch node, with the corresponding partition being statistically significant. As a by-product, it can automatically assess the clusterability and determine the number of clusters under the same umbrella.
- A simulated study demonstrates that our method can determine not to split at the root node for unclusterable data.
- Extensive experiments on real categorical data sets demonstrate our method's competitive performance compared to both non-interpretable and other interpretable clustering methods.

The structure of this paper is organized as follows: Section 2 reviews methods most relevant to our study. Section 3 presents a detailed description of our proposed method. Section 4 presents the experimental results on both simulated and real data sets. Finally, Section 5 concludes the paper.

2. Related work

Given the limited research efforts on interpretable categorical data clustering, we will review two related fields: interpretable clustering methods and categorical data clustering approaches. Additionally, our method has a distinct characteristic that is also connected to concepts in clusterability evaluation methods.

2.1. Interpretable clustering

Users find high-dimensional data clustering easier to understand when it is based on simple rules for individual features or attributes, rather than clusters relying on all dimensions simultaneously. To meet this need, a variety of interpretable models have been proposed, including logical formulas [18], rules [19], decision trees [20], and geometric boundaries such as prototypes [21], hyper-rectangles [22], hypercubes [23], polytopes [24], and polyhedron [25]. Among these, binary decision trees [12,17,26,27] stand out as the most popular and promising model. They offer a clear pathway to trace how clusters are derived, depending on individual feature values, from the root to leaf nodes.

In post-modeling algorithms, binary decision trees create a fixed number of leaf nodes, each corresponding to an explainable cluster closely aligned with its original cluster from a third-party clustering algorithm. Most algorithms in this category first obtain initial clustering result using k-means and then construct the decision tree based on the splitting criteria that try to approximate the optimal k-means cost [26,27]. In addition, some penalty factors such as the depth per leaf node can be incorporated into the cost function to yield more concise decision tree [12]. Obviously, such post-modeling algorithms have several limitations: the number of clusters has to be specified in advance and each split is evaluated based on a cost function that lacks a statistical interpretability.

In pre-modeling algorithms, the number of clusters can be automatically determined according to the number of leaf nodes of constructed decision tree. These methods encounter two main challenges: identifying the optimal split and determining the right time to stop splitting. Existing strategies commonly choose optimal splits based on inter-cluster separation and intra-cluster compactness of the candidate partition, typically through distance-based [14,28] or heterogeneity [17,8] measures. However, these metrics do not offer statistical interpretability, making it challenging to qualitatively justify whether the optimal split should be adopted for tree growth. As a result, establishing a truly effective stopping condition is a challenging issue, with trivial ones, such as ensuring a minimum number of objects in each leaf node, being routinely applied. To obtain a concise tree with less leaf nodes, a post hoc phase that includes pruning and merging processes is typically indispensable.

Hypothesis testing methods have been employed in some studies to construct decision trees: one facilitates tree growth [29], and the other serves as a stopping condition [16]. However, the former requires some known ground-truth labels in a supervised setting, making it unsuitable for interpretable clustering. The latter, as emphasized in its original paper, does not employ a strict statistical test since its assumptions may not hold in practice. More critically, it cannot provide analytical p-values, marking a fundamental difference from our method.

2.2. Categorical data clustering

In the field of cluster analysis, customized clustering algorithms have to be developed when we are trying to partition a specific type of data samples into different clusters. In particular, clustering categorical data, which involves discrete feature values, often makes standard numerical clustering methods unsuitable. Hence, the development of new algorithms for clustering categorical data has been widely investigated during the past decades.

In the field of categorical data clustering, a wide range of classic algorithms has been proposed, encompassing the most commonly used approaches such as partitional methods [30], density-based methods [31], model-based methods [32], and hierarchical methods [33]. Owing to the inherent discrete nature of categorical data, clusters typically comprise heterogeneous objects, which are not readily understood or interpreted through the geometric spaces applicable to numerical data. This heterogeneity stems from intra-attribute categories that lack quantifiable relationships and from inter-attribute differences where various attributes may exhibit distinct contexts.

Numerous similarity measures have been developed to describe the relationships between pairs of categorical objects, among which Hamming distance [30] and Entropy-based measures [34] are the most commonly used. To improve clustering accuracy, advanced clustering methods for categorical data often focus on analyzing the relationships among categorical values through complex representation learning techniques [35]. A common strategy involves embedding categorical data into a numerical format and then applying k-means [36]. However, these approaches often yield difficult-to-explain clustering outcomes, as the intermediate stages of representation learning are black-box methods and are decoupled from, as well as independent of, the final k-means cost function.

Among the methods for categorical data clustering, two are particularly relevant to our approach: CUBT [8] and DV [37]. The former employs decision trees, while the latter utilizes hypothesis testing techniques; yet, neither combines these two approaches. In detail, DV sequentially extracts statistically significant clusters by initially conducting a significance test to determine if a cluster center reflects a local pattern. It then determines an appropriate radius for identifying the objects that are assigned to that cluster center. Obviously, DV is still a traditional non-interpretable clustering method, which cannot provide an interpretable clustering result in terms of decision trees.

2.3. Clusterability evaluation methods

Despite the clear objective of clustering algorithms, several fundamental research issues in cluster analysis remain overlooked. In particular, for a given data set, is it truly meaningful to perform cluster analysis? If no clustering structure exists within the data set, then it is impossible to obtain meaningful clusters no matter which clustering algorithms are employed.

Clusterability evaluation methods [38] are designed to assess whether a data set exhibits a clustering structure, typically serving as a preliminary step before applying clustering algorithms. If the data set is found to lack such a structure, being indistinguishable from uniform data or random data, any clustering algorithm would not yield meaningful clusters, thus negating the need for further clustering analysis tasks. Likewise, in our method, this aim justifies the branching of nodes during tree growth, with each splitting being controlled and allowed only when the existence of a clustering structure is confirmed.

Several clusterability evaluation methods for numerical data have been proposed [39,40], with the most commonly used methods involving hypothesis testing, often setting the null hypothesis that the data are generated from a single multivariate Gaussian distribution. However, this approach assumes that the data are continuous and is, therefore, not applicable to categorical data. Hence, designing clusterability evaluation methods for categorical data remains an open problem.

3. Methods

3.1. Preliminaries

Consider a categorical data set $DS = \{O_1, \cdots, O_N\}$ of N objects in which each object is characterized by M attributes. For the i-th object O_i, its value on the m-th attribute can be one of Q_m categories in the set $A^m = \{A^m_1, \cdots, A^m_{Q_m}\}$. The aim of interpretable categorical data clustering is to partition DS into different clusters, where each cluster is described by a precise and understandable rule.

In the context of the binary decision tree model, the data set DS is recursively divided into two subsets during the tree growth stage. The final k leaf nodes correspond to k clusters $\{DS^1, \cdots, DS^k\}$. For the b-th branch node R_b (in depth-first order, with the root node as $b = 1$), it encompasses two types of information: the data set $DS^{(R_b)}$ within the current node, and a split point $S^{(R_b)}$, where any category A^m_q can be considered as the candidate one, with $S^{(R_b)} = A^m_q$. The set of objects $DS^{(R_b)}$ in the branch node R_b can be divided into two subgroups: $DS^{(R_b)}_1$ in the left child node (Group 1) and $DS^{(R_b)}_2$ in the right child node (Group 2). Specifically, $DS^{(R_b)}_1$ is composed of objects whose attribute values that match the category specified by $S^{(R_b)}$, whereas $DS^{(R_b)}_2$ does not.

To construct the interpretable clustering tree, branch nodes are sequentially split until they are deemed to be leaf nodes. For each possible split at the b-th branch node R_b, we derive a p-value to assess whether the two groups it induces demonstrate a statistically significant difference, denoted as $pval(DS^{(R_b)}_1, DS^{(R_b)}_2)$. Suppose the split $S^{(R_b)} = \hat{A}^m_q$ can yield the smallest p-value among all candidate splits, it will be used to split the b-th branch node if the p-value is less than the significance level.

3.2. Overview of SigDT

SigDT constructs a significance-based decision tree for clustering categorical data, where each branch node is automatically created and divided throughout a streamlined and greedy process. Our approach aims to identify truly optimal splits for effective tree growth, synonymous with the clustering process itself. The central idea is that not all possible splits in a given data set (or its subsets) are meaningful. For a clusterable data set, we aim at identifying the best split that significantly deviates from random splitting. For a data set that is inherently unclusterable, any attempt to split it becomes futile. That is, the inclusion of branch nodes during tree growth for such unclusterable data sets is unexplainable, even when an interpretable model is used.

As depicted in Fig. 1, SigDT accepts any categorical data set as input in step (a), consistently selecting the best split from all candidate splits based on the smallest p-value in step (c). The significance-based splitting criteria are utilized throughout steps (b) and (d), covering the phases before and after the selection of the best split. From step (a) to (b), all necessary information is directly extracted from the input data set, and upon applying a split, the data set is divided into two groups, resulting in two sets of frequency counts for each category. The p-value of a given split is calculated by assessing these counts, determining whether two groups exhibit statistical differences across a sufficient number of categories. When the best split is obtained in step (d), its p-value is compared with a dynamically adjusted significance level. If the p-value of the best split falls below this threshold, the split is deemed significant and used to form a branch node. Conversely, if the p-value is larger than the threshold, the current node will not be further divided and it is instead treated as a leaf node. Priority is given to the left child node of any branch node, which then serves as the new input data set for recursion back to step (a). This process continues until a leaf node is reached. Notably, if no statistically significant split exists for the root node, the data set is considered unclusterable.

3.3. Significance-based splitting criteria

To derive the p-value for each candidate split $S^{(R_b)}$, the assessment of differences between $DS^{(R_b)}_1$ and $DS^{(R_b)}_2$ can be formulated as a multivariate two-sample testing problem. The null hypothesis is that objects in these two groups are drawn from the same population, suggesting they should not be split. In contrast, the alternative hypothesis posits that the two groups are sufficiently heterogeneous to warrant division.

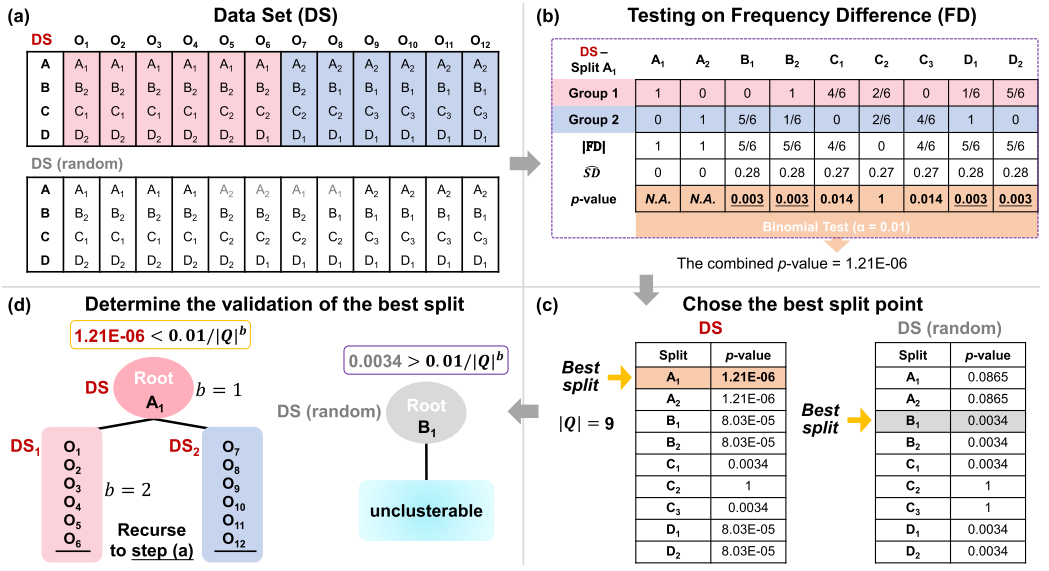

Fig. 1. Illustration of the SigDT framework: **(a)** Two toy example categorical data sets, each comprising 12 objects and 4 attributes {A, B, C, D}, are used for illustration purpose. **(b)** For each candidate split point in a data set, we calculate the corresponding p-value. For instance, the split point $A = A_1$ divides the data set DS into two groups, with the final p-value determined by considering the p-values from testing frequency differences across all categories. **(c)** At this stage, the best split is selected based on the smallest p-value. **(d)** This step verifies whether the p-value of the best split falls below a predetermined threshold; if so, the current branch node R^b is created. It shows that R^1 has been included for DS. In contrast, DS (random) is determined to be unclusterable since all candidate splits cannot yield statistically significant partitions at the root node. Subsequently, the left child node of R^1 is given priority for recursion, assessing the potential inclusion of R^2 by starting again from step (a) with DS_1 as the input.

Two-sample testing approaches applied to multivariate data can be classified as either parametric methods or nonparametric methods. Parametric methods require strong assumptions, such as specific knowledge about the underlying distribution of data. Among nonparametric methods, the commonly employed technique for multivariate data [41] involves first constructing a similarity graph for objects and then generating a minimum spanning tree (MST). However, in our context, such graph-based test has notable drawbacks: (1) The construction of similarity graph and the generation of MST may incur a quadratic time cost with respect to the number of objects. (2) The Euclidean distance adopted by existing graph-based tests is inapplicable for categorical data. Furthermore, using such distances to differentiate objects can be ineffective in high-dimensional scenarios, especially when the data set contains a relatively small number of objects.

Based on above observations, we present a new method for tackling the multivariate two-sample test issue for categorical data, which can achieve a linear time complexity by aggregating all necessary counts to calculate the test statistic through a single traversal of the data set.

3.3.1. *Test statistic and p-value calculation*

Considering the discrete nature of categorical data, the two-sample testing problem can be viewed as a multiple testing issue. Specifically, for each category A_q^m $(1 \le m \le M, 1 \le q \le Q_m)$, the individual null hypothesis posits that its occurrence probabilities [42] for the two groups produced by the split point $S^{(R_b)}$ are identical:

$$H_{0mq}^{S^{(R_b)}} : p_1(A_q^m) = p_2(A_q^m) = p(A_q^m), \tag{1}$$

and this is contrasted against the alternative hypothesis:

$$H_{1mq}^{S^{(R_b)}} : p_1(A_q^m) \neq p_2(A_q^m), \tag{2}$$

where p_1 and p_2 are the population success probabilities for group 1 and group 2, respectively, with the common probability p being unspecified.

It is natural to treat the two groups as two independent sequences of Bernoulli trials: one sequence consisting of n_1 trials with a success rate of p_1, and another sequence consisting of n_2 trials with a success rate of p_2. Given the observed frequency counts $\hat{p}_1$ and $\hat{p}_2$ (which are estimates of p_1 and p_2, respectively) in the data,[1] the frequency difference $\mathrm{FD} = \hat{p}_1 - \hat{p}_2$ serves as the unbiased estimator for $p_1 - p_2$. Thus, the mean of the sampling distribution of FD is equal to that of $p_1 - p_2$, and the standard deviation (SD) of the sampling distribution of FD can be estimated with sample variances as [42]:

[1] For example, as illustrated in Fig. 1(b) for the 'C_1' category, within group 1 there are $n_1 = 6$ trials with 4 successes, yielding $\hat{p}_1 = 4/6$. With no occurrences in group 2 ($\hat{p}_2 = 0$), the frequency difference is $\mathrm{FD} = 4/6 - 0 = 4/6$.

$$\widehat{SD}(\hat{p}_1 - \hat{p}_2) = \sqrt{var(\hat{p}_1 - \hat{p}_2)} = \sqrt{var(\hat{p}_1) + var(\hat{p}_2)} = \sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}. \tag{3}$$

Given the null hypothesis that $p_1 = p_2 = p$, we can replace $\hat{p}_1$, $\hat{p}_2$ with $\hat{p}$ to simplify the Equation (3):

$$\widehat{SD}(\hat{p}_1 - \hat{p}_2) = \sqrt{\frac{\hat{p}(1-\hat{p})}{n_1} + \frac{\hat{p}(1-\hat{p})}{n_2}}, \tag{4}$$

where

$$\hat{p} = \frac{\text{total successes in both groups}}{\text{total trials in both groups}} = \frac{n_1\hat{p}_1 + n_2\hat{p}_2}{n_1 + n_2} \tag{5}$$

is an estimator of the hypothesized common success rate p that pools information from the two groups.

The test statistic, represented as the standardized FD, is given by

$$Z_{FD} = \frac{\hat{p}_1 - \hat{p}_2}{\widehat{SD}(\hat{p}_1 - \hat{p}_2)}. \tag{6}$$

When both n_1 and n_2 are sufficiently large to ensure adequate sample sizes,[2] the null distribution of the test statistic Z_{FD} can be approximated by the standard normal distribution, denoted as $Z_{FD} \sim \mathcal{N}(0,1)$. This allows for the analytical derivation of the p-value for the two-sided test against the $H_{0mq}^{S^{(R_b)}}$ as follows:

$$p\text{-value}_{mq}(Z_{FD}) = 2 \times (1 - \Phi(|Z_{FD}|)), \tag{7}$$

where Φ is the cumulative distribution function of the standard normal distribution, and $|Z_{FD}| = \frac{|FD|}{\widehat{SD}}$ is the absolute value of the test statistic. This p-value quantifies the probability of observing a frequency difference as extreme as, or more extreme than, what is observed under the null hypothesis that the two groups have equal occurrence probabilities for A_q^m.

Now, considering the total number of categories, $|\mathbf{Q}| = \sum_{m=1}^{M} Q_m$, in the data, we will derive the final p-value for a specific $S^{(R_b)}$, which encompasses $|\mathbf{Q}|$ individual hypotheses. The global null hypothesis is expressed as:

$$H_0^{S^{(R_b)}} : \bigcap_{1 \le m \le M, 1 \le q \le Q_m} H_{0mq}^{S^{(R_b)}}, \tag{8}$$

and this is contrasted against the global alternative hypothesis:

$$H_1^{S^{(R_b)}} : \bigcup_{1 \le m \le M, 1 \le q \le Q_m} H_{1mq}^{S^{(R_b)}}, \tag{9}$$

where the global alternative hypothesis contends that at least one of the null hypotheses, $H_{0mq}^{S^{(R_b)}}$, is false. This hypothesis evaluates the collective evidence across all categories to determine whether the candidate split $S^{(R_b)}$ has resulted in two groups exhibiting statistically significant frequency differences. The decision to accept $H_1^{S^{(R_b)}}$ is based on the aggregated results from the individual p-values of $|\mathbf{Q}|$ independent tests.

To combine the evidence from all individual tests, we utilize a meta-analysis technique, specifically the Binomial test [43], to compute the final p-value against $H_0^{S^{(R_b)}}$. This is achieved by aggregating multiple p-values obtained via Equation (7) to form a new test statistic as follows:

$$r = \sum_{m=1}^{M} \sum_{q=1}^{Q_m} \delta(p\text{-value}_{mq}, \alpha), \tag{10}$$

where

$$\delta(p\text{-value}_{mq}, \alpha) = \begin{cases} 0 & \text{if } p\text{-value}_{mq} > \alpha \\ 1 & \text{if } p\text{-value}_{mq} \le \alpha \end{cases} \tag{11}$$

is an indicator function for each p-value according to the threshold α. Under $H_0^{S^{(R_b)}}$ that all $|\mathbf{Q}|$ null hypotheses hold, r, the number of tests that lead to rejection at the α level, follows a Binomial distribution representing its overall rejection probability. The cumulative distribution function of the Binomial distribution, which accounts for more extreme cases than r, is then summed to derive the final p-value for the candidate split $S^{(R_b)}$ as follows:

[2] In our approach, considering the small sample sizes inevitably encountered in practical scenarios, we set a threshold of more than 5 objects per group for further testing. This mirrors common practices in categorical data analysis, where it is typically recommended that expected counts in contingency tables exceed 5.

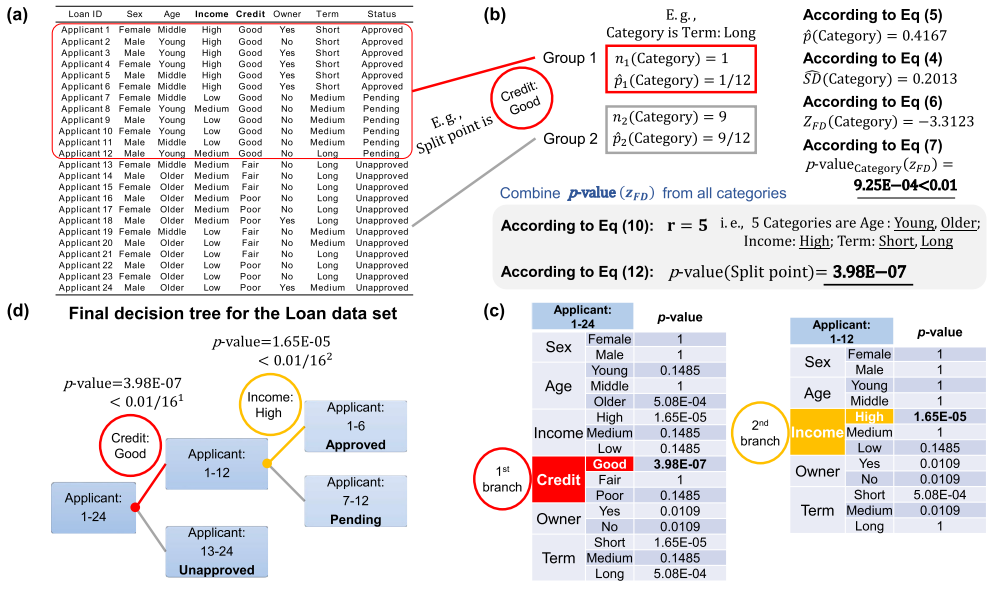

Fig. 2. (a) A Loan data set, comprising 24 records of applicants, each containing 6 attributes ('Sex', 'Age', 'Income', 'Credit', 'Owner', 'Term'). The records can be divided into 3 status clusters ('Approved', 'Pending', 'Unapproved') based on the applicant's information. **(b)** For a given selected split point (e.g., 'Credit = Good'), the Loan data set is divided into two groups based on whether samples contain the split point or not. We calculated the p-value of FD on each category (e.g., 'Term = Long') according to Equations (4)~(7). Then, all those p-values are combined into a single p-value according to Equations (10)~(12). **(c)** The best splits on the first and second branches are 'Credit = Good' and 'Income = High', each with the smallest p-value among all candidate splits. **(d)** The final decision tree for the Loan data set is constructed accordingly. According to the adjusted significance level in Equation (13), no further significant splits can be produced at any leaf node.

$$p\text{-value}(S^{(R_b)}) = \sum_{h=r}^{|\mathbf{Q}|} \binom{|\mathbf{Q}|}{h} \alpha^h (1-\alpha)^{|\mathbf{Q}|-h}, \tag{12}$$

which is the probability of obtaining at least r rejections among $|\mathbf{Q}|$ null hypotheses. Equation (12) combines all testing information among every categorical variable, providing a comprehensive assessment solution to $|\mathbf{Q}|$ two-sample testing problems.

3.3.2. Multiple testing correction

Given that an optimal split $S^{(R_b)} = \hat{A}_q^m$ is selected at the b-th branch node, the statistical significance of the resulting two groups is determined by comparing its p-value to a specified significance level. In scenarios involving only a single hypothesis test, the significance level α is typically set at 0.01. This level tolerates the possibility that up to 1% of more extreme outcomes might occur purely by chance. However, when handling multiple hypotheses (T tests), the risk of wrongly rejecting one or more true null hypotheses (Type I errors) increases as T grows. The probability of making at least one Type I error, i.e., the Family-Wise Error Rate (FWER), across T independent tests is calculated as $\text{FWER} = 1-(1-\alpha)^T$. For instance, with $T = 69$ and $\alpha = 0.01$, this FWER reaches 50%. Therefore, we must conduct a multiple-comparison correction in order to control the Type I error.

Specifically, we employ the Bonferroni correction method [44] to control the FWER using an adjusted significance level $\alpha^* = \frac{\alpha}{T}$. This ensures that $\text{FWER} = 1-(1-\alpha^*)^T \le \alpha$. This adjusted level is used to compare against the p-value to determine the statistical significance of the partition generated by $S^{(R_b)} = \hat{A}_q^m$ at the current b-th node. To implement this correction, we need to count the total number of multiple comparisons, T, which corresponds to the number of null hypotheses $H_0^{S^{(R_b)}}$ tested from the root node to the current node under a fixed tree structure. Each node follows sequentially from previously added branch nodes. When testing the b-th node, it includes the $b-1$ previously added optimal branch nodes and the current node itself, making a total of b nodes under consideration. To form each optimal node, it involves testing $|\mathbf{Q}|$ candidate splits ($|\mathbf{Q}|$ null hypotheses). Therefore, with T calculated in the aforementioned manner, the dynamically adjusted significance level α^* to validate $S^{(R_b)} = \hat{A}_q^m$ is computed as follows:

$$\alpha^*(R_b) = \frac{\alpha}{T} = \frac{\alpha}{|\mathbf{Q}|^b}. \tag{13}$$

3.4. An illustration on example data set

To elaborate on how SigDT works in practice, we use an example data set from the field of financial management to explain the clustering tree construction procedure.

The primary business of the bank is lending. In the vast amount of loan applicants, managers need to determine whether to approve a loan for each applicant. Here, we use a small but comprehensive Loan data set, as shown in Fig. 2(a), to demonstrate the clustering process of SigDT and its final interpretable results.

Besides the detailed calculations displayed in Fig. 2(b), we will show that the clustering process of SigDT is transparent. This transparency mainly stems from the fact that all candidate splits at each branch node are thoroughly evaluated using p-values, as

Table 2
Classification of the nine competing methods. Default parameter settings are utilized for these methods unless otherwise noted. Specifically, for CDCDR, options that can enhance the performance suggested by the authors are employed, including Spectral Embedding as the 'Graph Embedding Method' and the joint operation as the 'Integration Operation'.

Non-interpretable	Hypothesis testing	**DV**
	Classic	*k*-mode [30]; Entropy-based method [34]
	State-of-the-art	CDE [35]; CDCDR [36]
Interpretable	Pre-modeling	**CUBT**
	Post-modeling	IMM [26]; RDM [27]; SHA [12]

listed in Fig. 2(c), which are statistically interpretable values that range from 0 to 1. Specifically, from the list, we can observe that some candidate splits have relatively large *p*-values at all branch nodes. For instance, when using 'Sex' as a decision attribute (whether the split point is male or female), the *p*-value equals 1.

Ultimately, as shown in Fig. 2(d), we utilized two best splits with minimal *p*-values at each branch to obtain three clusters. Each cluster can be explained by an easily understandable decision rule. Specifically, applicants without good credit are unapproved. Among those with good credit, applicants with high income are directly approved. The remaining applicants with good credit but without high income are conservatively placed into the 'Pending' cluster for further manual review.

4. Results

In this section, we perform a detailed evaluation of our SigDT method on eighteen real-world categorical data sets, examining its performance in three aspects: clusterability prediction (Section 4.2), clustering quality (Section 4.3), and explainability (Section 4.4). The primary aim of our experiments is to empirically validate the effectiveness of SigDT compared to existing relevant clustering algorithms. Firstly, two most closely related categorical data clustering methods in the performance comparison are briefly summarized as follows:

- DV [37]: This algorithm,[3] which employs hypothesis testing, iteratively extracts statistically significant cluster centers until no more significant centers can be found, thus automatically determining the number of clusters. If no statistically significant center is identified during the initial extraction and no clusters are output, the data set is deemed unclusterable.
- CUBT [8]: This method begins by constructing a maximal growth tree, denoted as CUBT_{max}. It then applies two different pruning measures after the same joint processing stage, resulting in two algorithmic variants: CUBT^{Ham}, which utilizes Hamming distance, and CUBT^{MI}, which employs mutual information. We adhere to the default parameters provided in the original implementation by the authors,[4] which include trivial thresholds to control the tree growth that a maximum tree depth `lp=7`, a minimum number of objects in each leaf node `minsize=10`, and a minimum number of objects required to split a branch node `minsplit=20`.[5]

In addition, other seven interpretable and non-interpretable clustering algorithms are included in the experiments as well. All nine competing methods are classified into five sub-types, as shown in Table 2. The recently proposed interpretable clustering methods predominantly fall into the post-modeling category, designed primarily for numerical data. To enable comparison, we employ one-hot encoding to convert each categorical object into a numerical vector, allowing methods such as IMM,[6] RDM, and SHA[7] to be applied to the categorical data sets.

To compare with existing algorithms with respect to clustering quality (Section 4.3) and explainability (Section 4.4), SigDT consistently retains an initial optimal split, even with data deemed unclusterable. Unlike SigDT, DV, and CUBT, other methods do not consistently produce the same clustering result in each run. Therefore, to ensure a fair evaluation, we perform 50 independent runs for these algorithms on each data set and use the average result in the comparison. Additionally, these methods require specifying the number of clusters, K, which we set according to the ground-truth number provided for each data set. All experiments are conducted on an Intel i7-10700F @ 2.90 GHz personal computer with 16GB RAM.

4.1. Data sets and performance metrics

Table 3 presents the characteristics of 18 real-world categorical data sets, sorted by the number of objects in ascending order. These data sets, consisting entirely of categorical attributes (including 'Binary' or 'Integer' types, which are treated as categorical attributes after discretization), are publicly available from the UCI Machine Learning Repository.[8]

[3] https://github.com/hetong007/CategoricalClustering.
[4] https://www.i2m.univ-amu.fr/perso/badih.ghattas/cubt.php.
[5] In contrast, our method sets a more relaxed criterion to ensure sufficient sample size for testing, requiring at least 6 objects for each candidate leaf node, hence `minsize=6`.
[6] https://github.com/navefr/ExKMC.
[7] https://github.com/lmurtinho/ShallowTree.
[8] https://archive.ics.uci.edu/datasets?FeatureTypes=Categorical.

Table 3
The characteristics of 18 UCI categorical data sets.

Data set	*Abbr.*	*N*	*M*	|**Q**|	*K*
Lenses	Ls	24	4	9	3
Lung Cancer	Lc	32	56	159	3
Soybean (Small)	So	47	21	58	4
Zoo	Zo	101	16	36	7
Promoter Sequences	Ps	106	57	228	2
Hayes-Roth	Hr	132	4	15	3
Lymphography	Ly	148	18	59	4
Heart Disease	Hd	303	13	57	5
Solar Flare	Sf	323	9	25	6
Primary Tumor	Pt	339	17	42	21
Dermatology	De	366	33	129	6
House Votes	Hv	435	16	48	2
Balance Scale	Bs	625	4	20	3
Credit Approval	Ca	690	9	45	2
Breast Cancer	Bc	699	9	90	2
Mammographic Mass	Mm	824	4	18	2
Tic-Tac-Toe	Tt	958	9	27	2
Car Evaluation	Ce	1728	6	21	4

Table 4
The *p*-value of the initial optimal split by SigDT and its clusterability prediction results on ODS and RDS* for 18 UCI data sets. *p*-values with a gray background indicate that data sets are deemed unclusterable by SigDT, while bold values indicate agreement between DV and SigDT that the data sets are unclusterable.

Data set	Ls	Lc	So	Zo	Ps	Hr	Ly	Hd	Sf	Pt	De	Hv	Bs	Ca	Bc	Mm	Tt	Ce
α^*	0.001	6E-05	0.0002	0.0003	4E-05	0.0007	0.0002	0.0002	0.0004	0.0002	8E-05	0.0002	0.0005	0.0002	0.0001	0.0006	0.0004	0.0005
p-value (ODS)	**1**	3E-16	2E-44	3E-35	3E-10	**0.0096**	4E-22	1E-20	1E-28	3E-25	4E-127	1E-45	**1**	5E-21	1E-118	3E-10	2E-17	**1**
p-value (RDS*)	**1**	**0.0225**	**0.0028**	0.0503	**0.0086**	**1**	0.0029	0.0196	**0.0258**	0.0662	**0.0020**	0.0124	**0.0169**	**0.0104**	**0.0022**	**0.0138**	**0.2377**	**0.1903**

To evaluate the clusterability prediction capability in the absence of prior knowledge about whether a data set is inherently clusterable, we adopt a simulation approach by randomly reassigning categories to generate randomized data. It is reasonable to assume that completely randomized data is unclusterable. Progressively increasing the number of random reassignments for the original data makes the simulated randomness more evident. This simulation procedure serves as a benchmark to assess the capability of our method on clusterability prediction.

For evaluating clustering quality, we utilize two widely-used external validation metrics: Purity and F-score. These metrics assess the clustering outcomes by comparing the predicted cluster labels with the ground-truth labels. Higher values of these metrics indicate superior clustering quality.

For evaluating the explainability of clustering trees, where each leaf node corresponds to an individual cluster, we consider the complexity of decision rules from the root node to multiple leaf nodes. To this end, we employ three metrics to gauge the simplicity of the tree: the number of leaf nodes (nLeaf), the maximal depth of the tree (maxDepth), and the average depth of the leaf nodes (avgDepth). Lower values of these metrics indicate superior explainability, reflecting more intuitive and concise tree-based rules for describing cluster formation.

4.2. Simulation analysis on clusterability prediction

To generate a Randomized Data Set (RDS) from any Original Data Set (ODS), we uniformly and independently apply one of two strategies to all attributes, thereby changing the ODS into a RDS. Here is the process for the m-th attribute:

(1) **Category Exchange Strategy:** Begin by randomly selecting two objects, O_i and O_j, from the ODS, where their attribute values on the m-th attribute are exchanged. This operation only affects these two objects on the m-th attribute, with all other objects' values on this attribute remaining unchanged between the ODS and RDS. This exchange operation is independently repeated for each attribute, and a complete cycle of exchanges across all attributes is defined as a single exchange event. Multiple consecutive exchange events can be executed to incrementally increase the randomness of the RDS. A RDS subjected to more exchange events is presumed to exhibit a higher level of randomness compared to those with fewer exchange events.

(2) **Random Permutation Strategy:** Begin by generating a random permutation of integers from 1 to N (the total number of objects in the ODS). This new sequence is used to generate a new order of categories for the m-th attribute, replacing the original order. This permutation operation is independently repeated for each attribute, affecting multiple objects on each attribute. Similar to executing the Category Exchange Strategy multiple times, this strategy is designed to produce a completely randomized data set (RDS*), which is assumed to be unclusterable due to the high level of randomness introduced.

Table 4 displays the clusterability prediction results of SigDT, which consistently identifies each RDS* as being unclusterable. In contrast, DV fails to recognize RDS* as unclusterable for five data sets: Zo, Ly, Hd, Pt, and Hv. Regarding the ODS of all 18 UCI data sets, SigDT predicts most to be clusterable, with exceptions for Ls, Hr, Bs, and Ce, which are also deemed as being unclusterable by

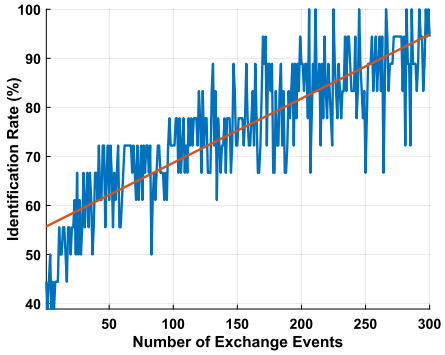

Fig. 3. Percentage of unclusterable randomized data sets identified by SigDT among the RDS versions of 18 UCI data sets, with increasing levels of simulated randomness. Each 'Number of Exchange Events' parameter is independently applied to generate a RDS directly based on the ODS for each simulation.

DV. This demonstrates that SigDT effectively differentiates between the clusterability of ODS and RDS*, under the assumption that UCI data sets are inherently clusterable.

To address the challenge of objectively determining whether a RDS is unclusterable, we track the trend in clusterability predictions against increasing levels of randomness in the RDS, as illustrated in Fig. 3. Initially, at 'Number of Exchange Events' = 1, where each RDS is generated through one exchange operation per attribute of the ODS, SigDT identifies eight data sets as unclusterable, corresponding to an identification rate of 44.4% (8 out of 18). As the 'Number of Exchange Events' increases, introducing more randomness into each RDS, SigDT progressively identifies more of the generated data sets as being unclusterable. Ultimately, after a sufficient number of exchange events, SigDT can achieve an identification rate of 100%, signaling that the RDS has attained a state of complete randomization.

4.3. Performance comparison on clustering quality

Fig. 4 depicts the clustering quality comparison, where algorithms positioned further towards the upper right corner of the coordinate chart for each data set indicate better performance with respect to Purity and F-score. We consider one algorithm to be superior to another if it surpasses in both metrics. The running times, recorded in seconds, are illustrated in Fig. 5. Observations from both Fig. 4 and Fig. 5 yield several key experimental conclusions:

(1) **Overall performance**: SigDT achieves good clustering quality on most data sets while requiring significantly less execution time compared to most algorithms. In terms of both Purity and F-score, SigDT is only surpassed by other algorithms on five specific data sets: Zo, Ly, Hd, Sf, and Bc, with the number of algorithms proving superior to SigDT on these data sets being 8, 2, 1, 2, and 7, respectively. The underperformance on the Zo data set is partly due to the fact that its ground-truth cluster set has two clusters with only 5 and 4 objects each, falling below the minimum candidate leaf node size of 6 required by our method during the hypothesis testing stage. In terms of Purity, SigDT achieves the best performance on 6 data sets: So, Ps, Hv, Ca, Mm, and Tt. As Purity may be sensitive to biases associated with the number of predicted clusters, we place additional emphasis on the performance metrics of the F-score. SigDT achieves the best F-score on 9 data sets: Ls, Lc, So, Hr, Pt, De, Bs, Mm, and Ce. Notably, Ls, Bs, and Ce, which are deemed unclusterable by SigDT, attain modest F-score values around 0.5. Furthermore, as depicted in Fig. 6(a), SigDT is significantly better than three competitors across all data sets. In contrast, the other algorithms (excluding DV) do not exhibit such pronounced superiority relative to each other.

(2) **Comparison with DV**: Although both SigDT and DV employ hypothesis testing procedures, SigDT runs significantly faster, typically requiring less than 0.01 seconds for most data sets, in contrast to DV, which requires at least 1 second for the majority of data sets. DV may also fail to report clustering results if it cannot identify any statistically significant cluster centers, which limits its practical utility. As shown in Fig. 4, DV can provide clustering outcomes for only 7 out of 18 data sets (So, Zo, Ly, Hd, De, Hv, Bc). In contrast, our algorithm identifies significant initial splits on most data sets and can even achieve modest clustering quality on those data sets deemed being unclusterable. Among the data sets where DV does report clusters, it outperforms SigDT on only Zo and Bc. In terms of the number of predicted clusters, DV correctly predicts the ground-truth cluster number on 2 data sets (So and Zo), which does not surpass SigDT since our method also predicts the right cluster number on two data sets (So and Mm), as illustrated in Fig. 7.

(3) **Comparison with CUBT**: Both CUBT^{Ham} and CUBT^{MI} are among the most time-consuming competitors and generally yield clusters of significantly lower quality than SigDT, as shown in Fig. 6. These versions of CUBT are only superior to SigDT on 2 (Hd, Bc) and 4 (Zo, Ly, Sf, Bc) data sets, respectively. Regarding the accuracy on predicting the number of clusters close to the ground-truth number K, as depicted in Fig. 7, CUBT^{Ham} and CUBT^{MI} collectively predict a cluster number that is close to the ground-truth on 9 data sets, while SigDT can achieve this goal on 11 data sets (Ls, Lc, So, Zo, Ps, Hr, Ly, De, Bs, Ca, Mm). Notably, CUBT_{Max} tends to produce more leaf nodes as the number of objects in the data set increases. This trend is visible in Fig. 7, where data sets are arranged in an ascending order of the number of objects. Unlike SigDT, CUBT_{Max} does not automatically halt splitting but instead relies on trivial conditions that may be sensitive to the scale of the data.

(4) **Comparison with non-interpretable clustering algorithms**: SigDT does not significantly outperform any specific non-interpretable clustering algorithms, but it excels over all these methods collectively on 2 data sets and achieves superior performance

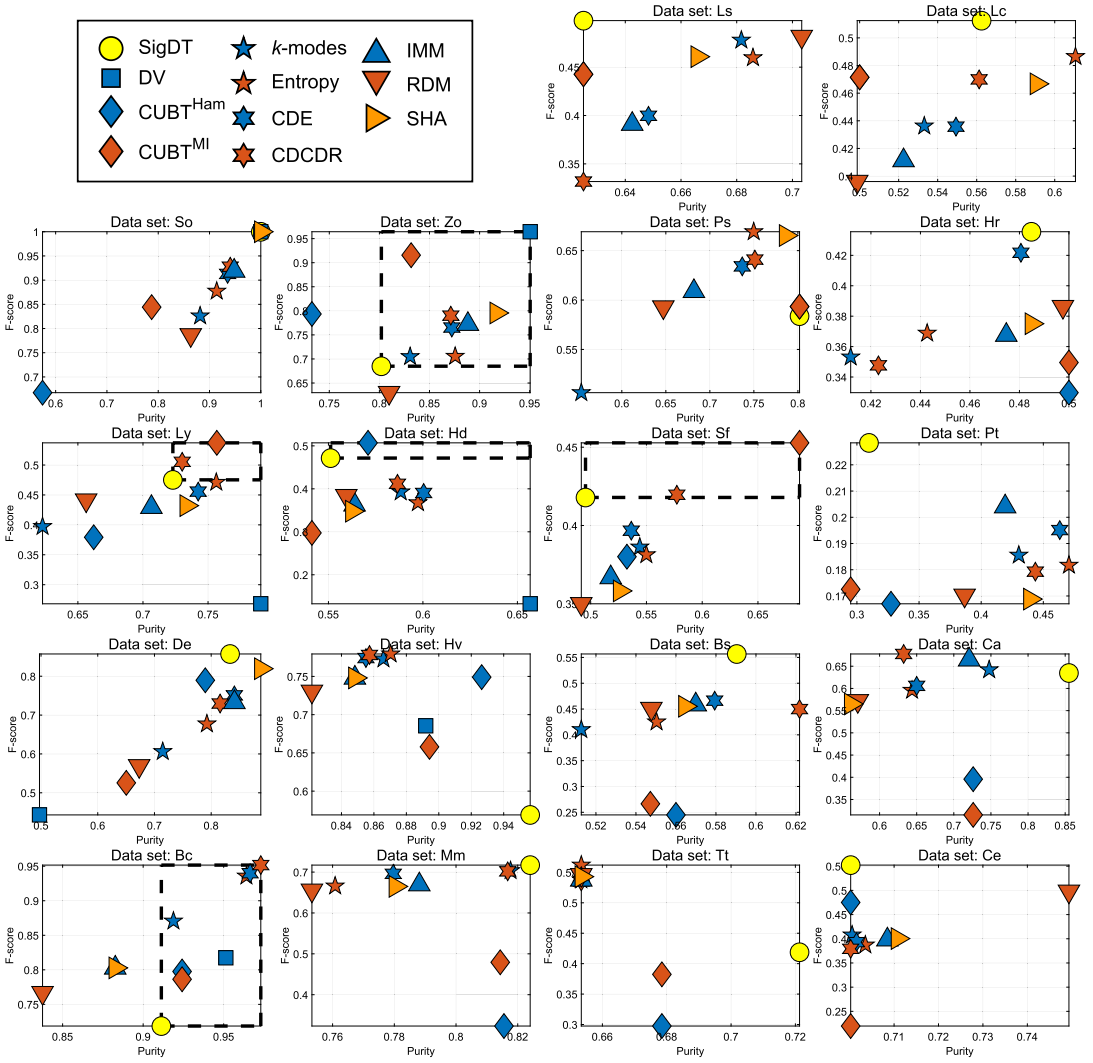

Fig. 4. The clustering quality comparison in terms of both Purity and F-score on 18 UCI data sets. For each data set, the algorithms that can outperform SigDT with respect to both Purity and F-score are highlighted with a dashed box.

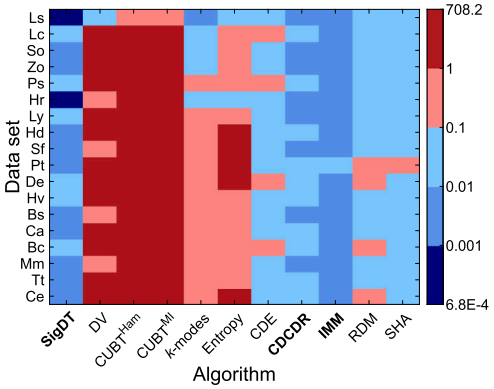

Fig. 5. Comparison of running times (in seconds) for all algorithms. The three fastest algorithms, IMM, CDCDR, and SigDT, are highlighted in bold and listed in ascending order of total execution time across all data sets. These algorithms are significantly faster than the remaining eight algorithms across all data sets, as confirmed by pairwise comparisons using the one-sided Wilcoxon signed-rank test at the 95% confidence interval.

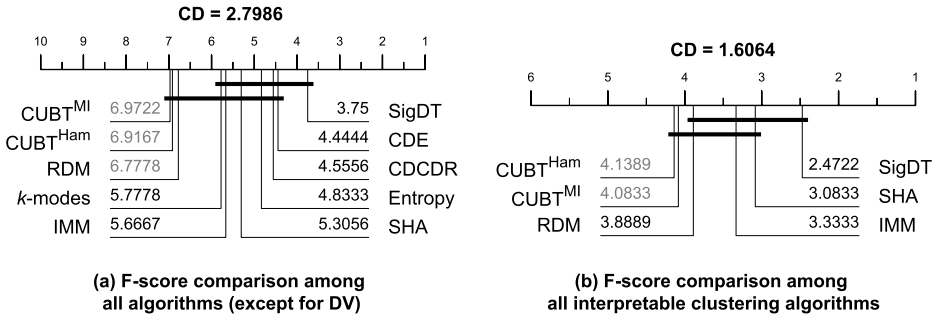

Fig. 6. Comparison of SigDT and other competitors in terms of F-score verified by the two-tailed Bonferroni-Dunn test [45] at the 95% confidence interval. In the Critical Difference (CD) diagram, each algorithm is positioned according to its average rank across all data sets. Algorithms that are not statistically significantly different in performance are connected by a thick line, the length of which represents the CD value. If the distance between two algorithms exceeds this length, the difference in their performance is considered statistically significant.

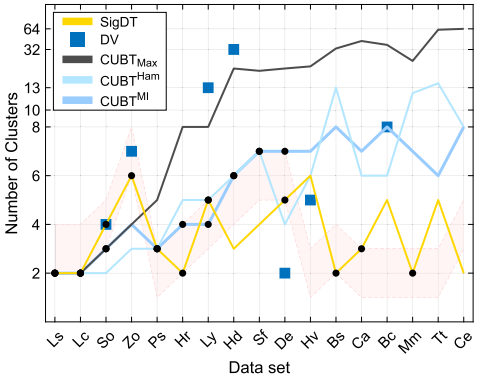

Fig. 7. The number of clusters predicted by SigDT, DV and CUBT. Suppose the ground-truth cluster number is K, the region between $K-1$ and $K+1$ for each data set is colored in pink. If the number of clusters predicted by one algorithm falls into this region, then the corresponding cluster number is marked with a black dot.

over CDE and CDCDR on 4 data sets (Lc, So, Hr, Mm). Conversely, both CDE and CDCDR are superior to SigDT on only 2 data sets (Zo, Bc). Moreover, SigDT runs faster than nearly all these algorithms, with the exception of CDCDR. However, CDCDR does not significantly outpace SigDT in speed, as confirmed by the one-sided Wilcoxon signed-rank test at the 95% confidence interval.

(5) **Comparison with interpretable clustering algorithms**: SigDT is competitive with these algorithms in terms of both clustering quality and running efficiency. Particularly, it significantly outperforms RDM in clustering quality among the ten compared algorithms, as illustrated in Fig. 6(a), and also runs significantly faster than RDM and SHA. For specific data sets, SigDT achieves superior performance over IMM, RDM, and SHA on 6, 8, and 4 data sets, respectively. However, only on the Zo data set do any of these algorithms surpass SigDT.

4.4. Performance comparison on explainability

Fig. 8 assesses the explainability of SigDT alongside other interpretable clustering algorithms, including the tree growth algorithm CUBT_{max} and others listed in Fig. 6(b), by comparing metrics such as 'maxDepth' and 'avgDepth'. The comparison on the 'nLeaf' metric, which represents the number of clusters produced by each algorithm, is included in Fig. 7, where the 'nLeaf' values for IMM, RDM, and SHA are fixed to be the ground-truth cluster number K.

In the comparison of 'nLeaf', since the number of clusters is not predetermined for SigDT and CUBT, these algorithms can produce any number of leaf nodes. As shown in Fig. 7, SigDT generally produces leaf nodes not exceeding K. In contrast, CUBT, even with post-hoc processing procedures to trim excess leaf nodes through two different measures, still tends to generate more than K leaf nodes, especially in data sets with a larger number of objects. Specifically, SigDT produces fewer leaf nodes than K on 9 data sets: Ls, Lc, Zo, Hr, Hd, Sf, De, Bs, and Ce. Conversely, CUBT_{max}, which uses a tree growth method with trivial stopping conditions, only manages this goal on 4 data sets with the fewest objects. Moreover, CUBT_{max} produces significantly more leaf nodes than all other interpretable algorithms. This is confirmed by the one-sided Wilcoxon signed-rank test at the 95% confidence interval.

In terms of both maxDepth and avgDepth, SigDT significantly achieves a more streamlined tree structure compared to CUBT_{max} and its two algorithmic variants. Notably, SHA achieves the second-best conciseness, surpassed only by our SigDT method, which aims to form shallow trees by incorporating measures like Weighted Average Depth into its objective function. However, constrained by the ground-truth number of clusters and external clustering algorithms, SHA fails to achieve a more streamlined tree structure. Even excluding Pt, the data set with the largest number of clusters ($K = 21$), the average maxDepth produced by SigDT is still smaller than that of SHA (2.06 vs. 2.11).

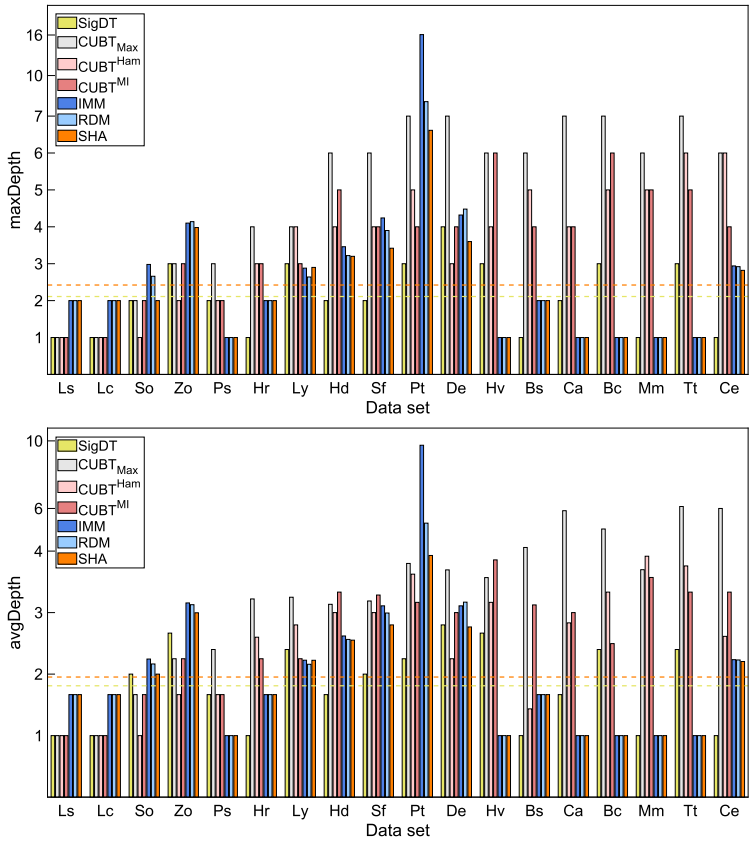

Fig. 8. The explainability comparison in terms of maxDepth and avgDepth among all interpretable clustering algorithms. The dash lines indicate the means of maxDepth and avgDepth across all data sets produced by SigDT and SHA. It is confirmed by the one-sided Wilcoxon signed-rank test at the 95% confidence interval that CUBT_{max} and its two algorithmic variants, CUBT^{Ham} and CUBT^{MI}, generally exhibit significantly higher maxDepth and avgDepth compared to SigDT.

Table 5
Summary of interpretable clustering methods used in the experiments, categorized by various criteria and intersecting with SigDT.

Interpretable model	Optimization approach	Stage	
		Pre-modeling	Post-modeling
Tree-based	Greedy search	**SigDT**; CUBT	IMM; RDM; SHA
Non-tree-based	PDM	**DReaM** [22]	
	MIO	**MPC-1** [24]	

4.5. *Extended performance comparisons*

In the above experiments, we compared the performance of SigDT in terms of clustering quality with both classic and advanced non-interpretable clustering methods for categorical data. We also assessed clustering explainability against recently proposed interpretable clustering methods that, like ours, use tree-based models. Selecting tree-based interpretable methods allows for a direct comparison of explainability by measuring the structural parameters of the clustering trees. However, many advanced interpretable clustering methods remain, most of which are designed for numerical data. Due to space limitations, we could not include all of them. To broaden the spectrum of advanced interpretable clustering methods for comparison, we selected two representative methods that differ from the tree-based approaches used in previous experiments. The differences between the newly added methods and those previously compared are illustrated in Table 5, as described in [46], summarizing various criteria. The two selected contrasting advanced interpretable algorithms are both pre-modeling approaches, ensuring strong counterparts for comparison with SigDT. Detailed descriptions of the experiments are provided in Section 4.5.1.

Another extended experiment focuses on testing several interpretable clustering methods alongside ours on more complex and challenging data sets, specifically high-dimensional and sparse data, to further evaluate applicability in real-world contexts. We will present experimental results only with IMM, RDM, and SHA for several reasons: (1) They are tree-based interpretable clustering methods, allowing for a direct comparison of explainability, which better highlights the strengths and weaknesses of SigDT. (2) They are time-efficient and can complete experiments on this type of data within a relatively acceptable total runtime, which is crucial in

Table 6
Performance comparison of non-tree-based interpretable clustering methods in terms of clustering quality and time efficiency. The compared methods are designed for numerical data as input, where categorical data is transformed using one-hot encoding. Note that " / " indicates that results on some data sets could not be generated by MPC-1 due to Multiple Integer Programming model optimizer limitations in its source code, imposed by the IBM® Decision Optimization CPLEX® PyPI version (https://pypi.org/project/docplex), exceeding the problem size limits (1000 variables, 1000 constraints).

	Purity			F-score			Running Times		
	SigDT	DReaM	MPC-1	SigDT	DReaM	MPC-1	SigDT	DReaM	MPC-1
Ls	0.625	0.646	**0.917**	**0.498**	0.404	0.276	**0.001**	1.636	18.864
Lc	**0.563**	0.506	0.438	**0.513**	0.399	0.481	**0.017**	9.229	315.463
So	**1**	**1**	**1**	**1**	**1**	**1**	**0.006**	5.288	71.182
Zo	0.802	0.897	**0.921**	0.685	0.776	**0.902**	**0.007**	8.756	51.124
Ps	0.802	0.642	**0.840**	**0.584**	0.579	0.560	**0.066**	26.292	1025.487
Hr	0.485	**0.512**	0.470	**0.436**	0.402	0.149	**0.001**	2.160	28.331
Ly	0.723	**0.734**	0.568	0.475	0.577	**0.654**	**0.012**	11.274	101.896
Hd	0.551	**0.573**	0.545	0.472	0.441	**0.523**	**0.008**	27.219	33.869
Sf	0.495	0.527	**0.567**	**0.418**	0.403	0.344	**0.004**	10.420	21.208
Pt	0.310	**0.433**	0.248	**0.228**	0.179	0.193	**0.010**	76.041	26.778
De	0.833	**0.880**	/	**0.857**	0.832	/	**0.046**	102.582	/
Hv	**0.956**	0.864	0.848	0.569	**0.771**	0.748	**0.016**	12.104	33.185
Bs	**0.590**	0.546	/	**0.557**	0.445	/	**0.002**	7.841	/
Ca	**0.855**	0.555	/	**0.635**	0.612	/	**0.007**	19.520	/
Bc	0.911	**0.970**	/	0.719	**0.946**	/	**0.021**	47.448	/
Mm	**0.824**	0.625	/	**0.717**	0.618	/	**0.002**	6.076	/
Tt	**0.721**	0.653	/	0.419	**0.529**	/	**0.009**	11.653	/
Ce	**0.700**	**0.700**	/	**0.552**	0.358	/	**0.003**	26.430	/
Avg	**0.708**	0.681	0.669	**0.574**	0.571	0.530	**0.013**	22.887	157.035

practical scenarios, unlike CUBT, DReaM, and MPC-1, which are more time-consuming. The specific experimental details are provided in Section 4.5.2.

4.5.1. Comparison with non-tree-based interpretable clustering methods

The two selected advanced interpretable clustering methods utilize non-tree-based interpretable models that form geometric boundaries to enclose each cluster. This enables the clustering results to be interpretable through understandable closed regions. Specifically, DReaM [22] identifies clusters using hyper-rectangles, while the MPC methods [24] use polytopes. The latter includes two variants, MPC-1 and MPC-2 (details and parameter settings are provided in the original paper). We adopted approaches that ensure the data is partitioned along feature axes. In other words, the boundaries of DReaM and the chosen MPC-1 are axis-parallel, which is commonly considered to improve interpretability compared to non-axis-parallel methods, such as those based on prototypes described in [46]. Given that both methods are relatively time-consuming, we ran each independently 10 times and reported the average results across 18 UCI data sets, as shown in Table 6.

Based on both prior analyses and the experimental results presented here, we conclude that our method, SigDT, demonstrates considerable strength in scenarios where the number of clusters is not provided as input. SigDT excels in finding high-quality clustering structures without requiring a predefined K. In the absence of ground-truth K, identifying meaningful clusters becomes much more challenging, as evidenced by the CUBT methods, which exhibit the most inferior clustering accuracy among the comparisons. While DReaM and MPC-1 achieve satisfactory clustering quality, they depend on the provision of K as input. Specifically, DReaM directly specifies the number of clusters, while MPC-1 [9] initializes with a range of K values. The clustering performance of MPC-1 declines compared to DReaM when K is not fixed as strong supervisory information. Furthermore, both methods introduce computational overhead due to the complexity of the optimization problems they solve, even with the use of commercial solvers.

4.5.2. Comparison on high-dimensional and sparse categorical data

We selected five benchmark DNA barcoding data sets sourced from [47] for this examination. Each data set consists of biological sequence samples aligned to the same length, allowing them to be treated as categorical data sets. The characteristics of these data sets are listed in Table 7, and they are typically high-dimensional and sparse, with large M and small $\frac{|\mathbf{Q}|}{NM}$. Given the substantially higher computational costs these data sets impose on various algorithms, we ran each comparison method independently 5 times on each data set and reported the average results, as shown in Table 8.

Based on the comprehensive comparison conducted on these data sets, we conclude that SigDT exhibits limitations and weaknesses in addressing the challenges posed by the data characteristics. One issue lies in the sharp increase in the number of hypothesis tests ($|\mathbf{Q}|$), where alternative hypotheses in Equation (9) become more susceptible to noise, resulting in more false positives, i.e., extremely small p-values calculated in Equation (12). This complicates the distinction between the two groups formed during the selection of

[9] Following the original configuration of `FCPS Experiment.ipynb` from https://github.com/conlaw/PolytopeClustering, we implemented their code to output the final result with the optimal silhouette score for each data set.

Table 7
The characteristics of five benchmark DNA barcoding data sets. For each categorical data set, N represents the number of barcodes in the data set, K represents the number of species, and $\frac{|\mathbf{Q}|}{NM}$ measures the sparsity of the categorical data by counting the number of unique categories distributed across the $N \times M$ data matrix.

Data set	*Abbr.*	N	M	$\|\mathbf{Q}\|$	K	$\frac{\|\mathbf{Q}\|}{NM}$
Área de Conservación Guanacaste	ACG	4267	663	2716	573	0.10%
Bats of Guyana	Bats	840	659	1504	96	0.27%
Birds of North America	Birds	2589	990	3321	656	0.13%
Fish of Australia	Fish	754	901	2115	211	0.31%
Hesperiidae	/	2185	664	1995	364	0.14%

Table 8
Performance comparison of tree-based interpretable clustering methods on benchmark DNA barcoding data sets in terms of clustering quality, time efficiency and explainability. Method with its performance measure significantly inferior to that of SigDT is highlighted in red, as confirmed by pairwise comparisons using the one-sided Wilcoxon signed-rank test at the 95% confidence interval.

Measure	Method	ACG	Bats	Birds	Fish	Hesperiidae	*Avg*
Purity	SigDT	0.262	0.604	0.134	0.267	0.337	0.321
	IMM	**0.813**	**0.964**	0.790	**0.918**	**0.825**	**0.862**
	RDM	0.699	0.900	0.641	0.869	0.719	0.766
	SHA	0.811	0.959	**0.794**	0.880	0.821	0.853
F-score	SigDT	0.232	0.634	0.110	0.223	0.325	0.305
	IMM	0.717	**0.871**	0.676	**0.803**	**0.720**	**0.757**
	RDM	0.631	0.774	0.534	0.719	0.647	0.661
	SHA	**0.724**	0.864	**0.688**	0.768	0.718	0.752
Running Times	SigDT	474.114	20.671	257.711	34.751	103.141	178.078
	IMM	**50.127**	**1.154**	**54.532**	**3.025**	**18.088**	**25.385**
	RDM	1654.183	33.292	1895.702	88.257	380.541	810.395
	SHA	537.347	8.146	639.287	21.268	103.654	261.940
nLeaf	SigDT	**106**	**37**	**52**	**41**	**64**	**60**
	IMM	574	96	657	211	366	381
	RDM	573	96	656	211	364	380
	SHA	571	91	624	184	361	366
maxDepth	SigDT	46	**8**	**12**	**10**	19	19
	IMM	173	32	180	37	159	116
	RDM	46	17	48	20	46	35
	SHA	**14**	9	13	**10**	**13**	**12**
avgDepth	SigDT	17	**6**	**7**	**7**	**9**	**9**
	IMM	84	17	91	19	87	60
	RDM	26	10	28	12	24	20
	SHA	**10**	7	10	8	**9**	**9**

the optimal branch node, and as tree depth increases, leads to numerous erroneous splits, ultimately resulting in inaccurate clustering outcomes. Additionally, controlling the significance threshold becomes more challenging. Without stringent control, as outlined in Equation (13), the method can generate a large number of leaf nodes, contradicting our goal of forming clusters with a concise tree structure. Although the clustering quality may be predictably suboptimal, as shown in Table 8, despite some inherent issues with the use of p-values [48], our designed dynamic significance thresholds still produce relatively shallow trees for these data sets, comparable to SHA. Specifically, our method significantly outperforms 3, 1, and 2 of the comparison algorithms in terms of nLeaf, maxDepth, and avgDepth, respectively. Moreover, the runtime of our method is the second fastest, following IMM, which leverages CPython (a faster language than the Matlab used in our implementation).

Furthermore, SHA performs well across nearly all measures. This can be partly attributed to its incorporation of explainability (tree structure parameters) into the optimization objective, resulting in a more balanced tree. Although SHA produces more leaves than SigDT, it achieves a smaller maxDepth and a nearly identical avgDepth. This suggests that, despite our goal of choosing significance-based split, our method may have overlooked some important branch nodes, leading to an imbalanced tree.

4.6. Summary

To consolidate the experimental results, we summarize the pros and cons of SigDT based on its performance across various data sets and in comparison with other methods:

- Strengths: (1) It offers interpretability in the clustering process, making the clustering algorithm more transparent. (2) It has the ability to predict clusterability, refusing to conduct cluster analysis on data sets without inherent clustering structure. (3) The

clustering result can be represented using a concise decision tree structure, enhancing human understanding of the clustering outcomes.

- Weaknesses: (1) It cannot perfectly predict the number of clusters, which may pose issues in practical applications when it is critical to know the cluster number exactly. (2) Its runtime is still time-consuming on large data sets, which could hinder its deployment in practice. (3) It may be less effective for high-dimensional or sparse categorical data, as it may omit key splits needed to detect clustering structures at finer granularity, leading to inaccurate results.

5. Conclusions

In this paper, we introduced SigDT, a novel decision tree-based method designed to enhance interpretability in clustering categorical data. SigDT offers several features that facilitate human understanding of the clustering process: (1) It operates without the need for hard-to-specify parameters or reliance on external algorithms, enabling automatic and controllable decisions. This approach removes external interference, allowing users to focus entirely on the clustering process. (2) It produces a concise clustering decision tree with fewer rules. (3) It assesses whether a data set inherently possesses a clustering structure, helping to prevent the generation of meaningless and perplexing clustering results. Experiments on real data sets empirically demonstrate that our method is competitive with advanced categorical data clustering algorithms in terms of clustering quality and running efficiency. Crucially, SigDT can achieve a more streamlined tree structure compared to existing optimization-based interpretable clustering algorithms when applied to categorical data.

Our approach faces challenges in enhancing its flexibility: (1) When users wish to specify a custom number of clusters instead of relying on the automatically determined cluster number, it necessitates recalibrating significance thresholds to control splits. (2) Although the smallest p-value typically determines the optimal split, considering other splits with small p-values might also be effective and could influence the global tree structure. We will explore alternative splits that are also statistically significant to check if better clustering results may be obtained.

Finally, while SigDT currently relies on p-values to assess the statistical significance of splits within our decision tree clustering process, we acknowledge the limitations associated with their use. These limitations, including sensitivity to sample size and potential misinterpretation, have been well-documented in the statistical literature [48]. Despite their computational efficiency and applicability in handling complex data sets in real-world contexts, p-values have generated ongoing debates regarding their robustness. In future work, we plan to explore alternative statistical measures, such as effect sizes, Bayesian approaches like Bayes factors [49], or information criteria like AIC and BIC. These alternatives are expected to improve the robustness and reliability of our method by addressing the inherent challenges associated with using p-values. By integrating these methods, we aim not only to mitigate the limitations of p-values but also to improve the overall effectiveness of our significance-based framework for interpretable clustering problems.

CRediT authorship contribution statement

Lianyu Hu: Writing – original draft, Visualization, Software, Methodology. **Mudi Jiang:** Validation, Investigation. **Xinying Liu:** Formal analysis, Data curation. **Zengyou He:** Writing – review & editing, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been supported by the Natural Science Foundation of China under Grant No. 62472064.

Data availability

Data will be made available on request.

References

[1] A.M. Ikotun, A.E. Ezugwu, L. Abualigah, B. Abuhaija, J. Heming, K-means clustering algorithms: a comprehensive review, variants analysis, and advances in the era of big data, Inf. Sci. 622 (2023) 178–210.
[2] P. Bhattacharjee, P. Mitra, A survey of density based clustering algorithms, Front. Comput. Sci. 15 (2021) 1–27.
[3] L. Yang, W. Fan, N. Bouguila, Clustering analysis via deep generative models with mixture models, IEEE Trans. Neural Netw. Learn. Syst. 33 (1) (2020) 340–350.
[4] I. Chami, A. Gu, V. Chatziafratis, C. Ré, From trees to continuous embeddings and back: hyperbolic hierarchical clustering, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, vol. 33, 2020, pp. 15065–15076.
[5] K.R. Varshney, H. Alemzadeh, On the safety of machine learning: cyber-physical systems, decision sciences, and data products, Big Data 5 (3) (2017) 246–255.
[6] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. 1 (5) (2019) 206–215.
[7] S. Naouali, S. Ben Salem, Z. Chtourou, Clustering categorical data: a survey, Int. J. Inf. Technol. Decis. Mak. 19 (01) (2020) 49–96.

[8] B. Ghattas, P. Michel, L. Boyer, Clustering nominal data using unsupervised binary decision trees: comparisons with the state of the art methods, Pattern Recognit. 67 (2017) 177–185.
[9] S. Bandyapadhyay, F.V. Fomin, P.A. Golovach, W. Lochet, N. Purohit, K. Simonov, How to find a good explanation for clustering?, Artif. Intell. 322 (2023) 103948.
[10] H. Hwang, S.E. Whang, Xclusters: explainability-first clustering, in: Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence, vol. 37, 2023, pp. 7962–7970.
[11] K. Makarychev, L. Shan, Random cuts are optimal for explainable k-medians, in: Proceedings of the 37th International Conference on Neural Information Processing Systems, vol. 36, 2023, pp. 66890–66901.
[12] E. Laber, L. Murtinho, F. Oliveira, Shallow decision trees for explainable k-means clustering, Pattern Recognit. 137 (2023) 109239.
[13] M. Fleissner, L.C. Vankadara, D. Ghoshdastidar, Explaining kernel clustering via decision trees, in: The Twelfth International Conference on Learning Representations, 2024.
[14] D. Bertsimas, A. Orfanoudaki, H. Wiberg, Interpretable clustering: an optimization approach, Mach. Learn. 110 (1) (2021) 89–138.
[15] M. Gabidolla, M.Á. Carreira-Perpiñán, Optimal interpretable clustering using oblique decision trees, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 400–410.
[16] H. Blockeel, L.D. Raedt, J. Ramon, Top-down induction of clustering trees, in: Proceedings of the 15th International Conference on Machine Learning, 1998, pp. 55–63.
[17] R. Fraiman, B. Ghattas, M. Svarc, Interpretable clustering using unsupervised binary trees, Adv. Data Anal. Classif. 7 (2013) 125–145.
[18] B. Kim, J.A. Shah, F. Doshi-Velez, Mind the gap: a generative approach to interpretable feature selection and extraction, in: Proceedings of the 28th International Conference on Neural Information Processing Systems, vol. 28, 2015, pp. 2260–2268.
[19] E. Carrizosa, K. Kurishchenko, A. Marín, D. Romero Morales, On clustering and interpreting with rules by means of mathematical optimization, Comput. Oper. Res. 154 (2023) 106180.
[20] B. Liu, Y. Xia, P.S. Yu, Clustering through decision tree construction, in: Proceedings of the Ninth International Conference on Information and Knowledge Management, 2000, pp. 20–29.
[21] E. Carrizosa, K. Kurishchenko, A. Marín, D.R. Morales, Interpreting clusters via prototype optimization, Omega 107 (2022) 102543.
[22] J. Chen, Y. Chang, B. Hobbs, P. Castaldi, M. Cho, E. Silverman, J. Dy, Interpretable clustering via discriminative rectangle mixture model, in: IEEE 16th International Conference on Data Mining, 2016, pp. 823–828.
[23] L. Chen, C. Zhong, Z. Zhang, Explanation of clustering result based on multi-objective optimization, PLoS ONE 18 (10) (2023) 1–30.
[24] C. Lawless, J. Kalagnanam, L.M. Nguyen, D. Phan, C. Reddy, Interpretable clustering via multi-polytope machines, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, 2022, pp. 7309–7316.
[25] C. Lawless, O. Gunluk, Cluster explanation via polyhedral descriptions, in: Proceedings of the 40th International Conference on Machine Learning, vol. 202, 2023, pp. 18652–18666.
[26] M. Moshkovitz, S. Dasgupta, C. Rashtchian, N. Frost, Explainable k-means and k-medians clustering, in: Proceedings of the 37th International Conference on Machine Learning, vol. 119, 2020, pp. 7055–7065.
[27] K. Makarychev, L. Shan, Explainable k-means: don't be greedy, plant bigger trees!, in: Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing, 2022, pp. 1629–1642.
[28] L. Jiao, H. Yang, Z.-g. Liu, Q. Pan, Interpretable fuzzy clustering using unsupervised fuzzy decision trees, Inf. Sci. 611 (2022) 540–563.
[29] G.V. Kass, An exploratory technique for investigating large quantities of categorical data, J. R. Stat. Soc., Ser. C, Appl. Stat. 29 (2) (1980) 119–127.
[30] Z. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, Data Min. Knowl. Discov. 2 (3) (1998) 283–304.
[31] D. Gibson, J. Kleinberg, P. Raghavan, Clustering categorical data: an approach based on dynamical systems, VLDB J. 8 (2000) 222–236.
[32] P. Cheeseman, J. Stutz, Bayesian classification (autoclass): theory and results, Advances in Knowledge Discovery and Data Mining (1996) 153–180.
[33] S. Guha, R. Rastogi, K. Shim, Rock: a robust clustering algorithm for categorical attributes, Inf. Syst. 25 (5) (2000) 345–366.
[34] T. Li, S. Ma, M. Ogihara, Entropy-based criterion in categorical clustering, in: Proceedings of the 21st International Conference on Machine Learning, 2004, p. 68.
[35] S. Jian, G. Pang, L. Cao, K. Lu, H. Gao, Cure: flexible categorical data representation by hierarchical coupling learning, IEEE Trans. Knowl. Data Eng. 31 (5) (2019) 853–866.
[36] L. Bai, J. Liang, A categorical data clustering framework on graph representation, Pattern Recognit. 128 (2022) 108694.
[37] P. Zhang, X. Wang, P.X.-K. Song, Clustering categorical data based on distance vectors, J. Am. Stat. Assoc. 101 (473) (2006) 355–367.
[38] A. Adolfsson, M. Ackerman, N.C. Brownstein, To cluster, or not to cluster: an analysis of clusterability methods, Pattern Recognit. 88 (2019) 13–26.
[39] J. Laborde, P.A. Stewart, Z. Chen, Y.A. Chen, N.C. Brownstein, Sparse clusterability: testing for cluster structure in high dimensions, BMC Bioinform. 24 (1) (2023) 1–27.
[40] A.F. Diallo, P. Patras, Deciphering clusters with a deterministic measure of clustering tendency, IEEE Trans. Knowl. Data Eng. 36 (4) (2024) 1489–1501.
[41] H. Chen, X. Chen, Y. Su, A weighted edge-count two-sample test for multivariate and object data, J. Am. Stat. Assoc. 113 (523) (2018) 1146–1155.
[42] M. Hollander, D.A. Wolfe, E. Chicken, Nonparametric Statistical Methods, John Wiley & Sons, 2013.
[43] O. Cinar, W. Viechtbauer, The poolr package for combining independent and dependent p values, J. Stat. Softw. 101 (2022) 1–42.
[44] X. Cui, T. Dickhaus, Y. Ding, J.C. Hsu, Handbook of Multiple Comparisons, CRC Press, 2021.
[45] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[46] L. Hu, M. Jiang, J. Dong, X. Liu, Z. He, Interpretable clustering: a survey, preprint, arXiv:2409.00743, 2024.
[47] P. Kuksa, V. Pavlovic, Efficient alignment-free dna barcode analytics, BMC Bioinform. 10 (2009) 1–18.
[48] R.L. Wasserstein, N.A. Lazar, The asa statement on p-values: context, process, and purpose, Am. Stat. 70 (2) (2016) 129–133.
[49] L. Held, M. Ott, On p-values and Bayes factors, Annu. Rev. Stat. Appl. 5 (1) (2018) 393–419.