Shallow decision trees for explainable k-means clustering

Eduardo Laber
Department of Computer Science
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro, RJ - Brazil
eduardo.laber@puc-rio.br

Lucas Murtinho
Department of Computer Science
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro, RJ - Brazil
lmurtinho@aluno.puc-rio.br

Felipe Oliveira
Department of Mathematics
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro, RJ - Brazil
felipedeoliveira1407@gmail.com

Abstract

A number of recent works have employed decision trees for the construction of explainable partitions that aim to minimize the k-means cost function. These works, however, largely ignore metrics related to the depths of the leaves in the resulting tree, which is perhaps surprising considering how the explainability of a decision tree depends on these depths. To fill this gap in the literature, we propose an efficient algorithm that takes into account these metrics. In experiments on 16 datasets, our algorithm yields better results than decision-tree clustering algorithms such as the ones presented in Dasgupta et al. [2020], Frost et al. [2020], Laber and Murtinho [2021] and Makarychev and Shan [2021a], typically achieving lower or equivalent costs with considerably shallower trees. We also show, through a simple adaptation of existing techniques, that the problem of building explainable partitions induced by binary trees for the k-means cost function does not admit an $(1+\epsilon)$-approximation in polynomial time unless $P = NP$, which justifies the quest for approximation algorithms and/or heuristics.

1 Introduction

As machine learning models have become used in a wide range of fields, the topic of *explainability* has grown in importance. Understanding the reasoning behind a model's decision may be crucial to increase user confidence; to satisfy legal requirements; to conform to moral and ethical expectations; and to verify the model's work. Since more complex models tend to be harder to interpret but are also more capable of returning good results, there is a trade-off between model performance and explainability. The challenge of navigating this trade-off is increasingly being explored in the machine learning literature.

Although initial efforts towards explainability focused on supervised learning models, a number of studies on explainable unsupervised models, and clustering models in particular, have appeared more recently. One idea that has earned some attention in the literature is to partition the data based on axis-aligned cuts, which can be induced by binary decision trees: at each node u of the tree, a value v and a dimension i are selected, so that all data points that have reached u go to one of its two children according to whether their values for dimension i are smaller than v or not. In this kind of approach, usually, each cluster is associated with a leaf.

Decision trees are widely considered to be explainable models by machine learning standards. However, the explainability of a cluster induced by a decision tree greatly depends on the depth of its associated leaf – the explanations for leaves that are far from the root involve many tests, which makes it harder to grasp the model's logic.

There are many possible metrics that can be associated with the depths of the leaves, as the maximum depth and the average depth. Here, we focus on metrics that consider as equally important the explanation of each data point. More specifically, we consider the *Weighted Average Depth* (`WAD`) and the *Weighted Average Explanation Size* (`WAES`). The former weighs the depth of each leaf by the number of points of its associated cluster; to minimize it, large clusters shall be associated with shallower leaves (shorter explanations). The latter is a variation of `WAD` that replaces the depth of a leaf by the number of non-redundant tests in the path from the root to the leaf. As an example, to explain the cluster associated with the leftmost leaf in Figure 1b we do not need the condition $D13 \leq 68$ because the condition $D13 \leq 17$ makes it redundant, so this cluster can be explained with 3 conditions: $D13 \leq 17$ AND $D14 \leq 56$ AND $D0 \leq 66$. These measures are formalized and discussed more thoroughly in Section 1.1.

Figure 1 shows two decision trees that partition the `Pendigits` dataset [Dua and Graff, 2017] into 10 clusters. The tree at the top, built by the IMM algorithm from Dasgupta et al. [2020], has `WAD` $\approx$ `5.5`, while the one in the bottom has `WAD` $\approx$ `3.6` — meaning that it provides a "gain" of almost 2 conditions on average. In terms of `WAES`, the gain is of approximately 1 test on average.

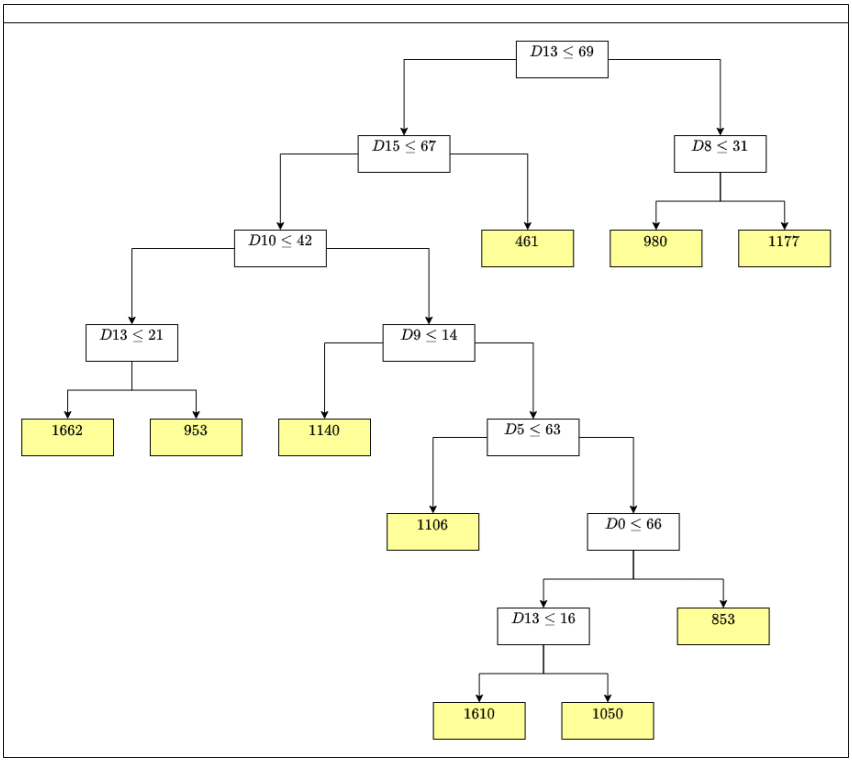
Moreover, the cost (with respect to the k-means goal) of the partition induced by the tree in Figure 1b is approximately 11% smaller than that induced by the tree in Figure 1a. In other words, the shallower tree induces a partition that is both less costly and more explainable. This example suggests that there is significant room to improve the explainability of the partitions provided by algorithms available in the literature.

Our contributions. As in Frost et al. [2020], Dasgupta et al. [2020], Laber and Murtinho [2021], we investigate the problem of building explainable clustering via decision trees. The main difference of our work with respect to the previous ones is our focus on building decision trees that simultaneously yield short explanations and induce partitions of good quality in terms of the k-means cost function. We understand that one relevant contribution of our paper is the observation that previous approaches overlook the important aspect of minimizing measures related to the tree's depth.

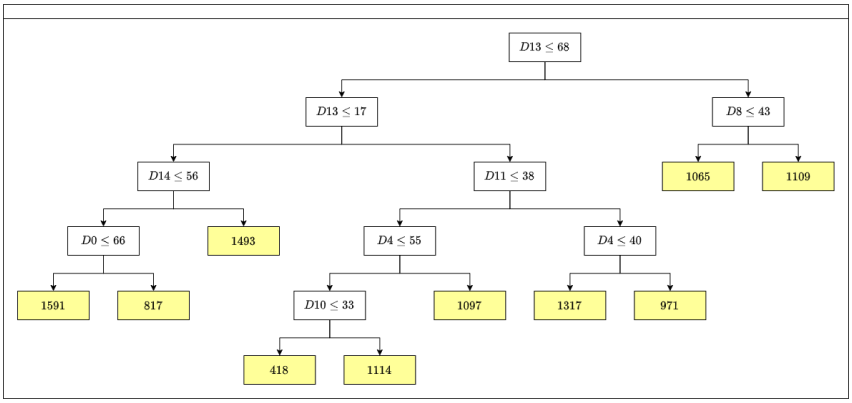
The other contributions are as follows. In Section 3, we show that the problem of building a partition via decision trees does not admit an $(1+\epsilon)$-approximation in polynomial time unless $P = NP$. Our proof is a simple adaptation of that employed by Awasthi et al. [2015] to obtain an equivalent result for the standard k-means clustering problem, which we document here simply to provide additional ground to the topic and to formally justify the quest for approximation algorithms and/or heuristics.

In Section 4 we present a strategy that builds decision trees that induce partitions of low k-means cost and have low values for `WAES` and `WAD`. As other proposals in the literature, we start from the partition provided by some algorithm for the (non-explainable) k-means clustering problem and build the tree in a top-down fashion, by selecting at each node a cut that is "good" in terms of minimizing our metrics. The key novelties we present here are an effective and efficient way to evaluate the potential of a cut in terms of `WAD`/`WAES` and how to efficiently trade-off the (potentially conflicting) goals of minimizing both these metrics and the cost of the induced partition.

To evaluate our strategy, in Section 5, we compare its performance against recently proposed algorithms over 16 datasets. Our strategy generated partitions as good as the best of its competitors in terms of the k-means cost, while being significantly better in terms of the aforementioned explainability measures. It also compares to the best of these competitors in terms of explainability, while inducing much better partitions than this competitor in terms of the k-means cost. Moreover, these gains were obtained without compromising computational efficiency. Thus, we believe that our



(a) Tree from IMM algorithm.



(b) An alternative tree

Figure 1: Two trees for partitioning the `Pendigits` dataset into 10 clusters.

strategy is a valuable tool for those that are interested in explainable partitions that optimize the quite popular k-means cost.

1.1 Preliminaries and Problem Definition

Let $\mathcal{X}$ be a collection of n data points in $\mathbb{R}^d$ and $k \geq 2$ be an integer. In (cost-oriented) hard clustering problems, we want to find a partition of $\mathcal{X}$ that minimizes a given cost function. In the widely studied k-means clustering problem, the cost of a partition $\mathcal{P} = \{C_1, \ldots, C_k\}$ is the sum of the squared Euclidean distances between all points in $\mathcal{X}$ and the representatives of the clusters to which they belong:

$$\text{cost}(\mathcal{P}) = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} ||\mathbf{x} - \mathbf{c}_i||_2^2. \tag{1}$$

In this case, the representative $\mathbf{c}_i$ of cluster C_i is given by the mean of its points, $\mathbf{c}_i = \frac{\sum_{\mathbf{x} \in C_i} \mathbf{x}}{|C_i|}$.

In our study we are interested in partitions induced by axis-aligned binary decision trees. A decision tree is axis-aligned if each internal node v is associated with a test (cut), specified by a coordinate $i_v \in [d]$ and a real value θ_v, that partitions the points in $\mathcal{X}$ that reach v into two sets: those having the coordinate i_v smaller than or equal to θ_v and those having it larger than θ_v. The leaves induce a partition of $\mathbb{R}^d$ into axis-aligned regions and, naturally, a partition of $\mathcal{X}$ into clusters.

For our purposes, it will be convenient to associate a condition to each edge of the tree: the left edge leaving a node v is associated with the condition $x_{i_v} < \theta_v$ and the right one with the condition $x_{i_v} > \theta_v$. The explanation of a cluster C in a decision tree $\mathcal{D}$ is given by the logical AND of the conditions associated with the edges in the path from the root of $\mathcal{D}$ to the leaf associated with C. We say that a condition is *redundant* with respect to cluster C if its removal does not change the explanation for C. As an example, if the explanation of cluster C is $x_1 > 30$ AND $x_2 < 20$ AND $x_1 > 70$, then the condition $x_1 > 30$ is redundant.

We consider two explainability measures for our study, namely the Weighted Average Explaination Size (WAES) and the Weighted Average Depth (WAD). For a partition $\mathcal{P} = (C_1, \ldots, C_k)$ induced by a binary decision tree $\mathcal{D}$ with k leaves, where the cluster C_i is associated with the leaf i, we have

$$\texttt{WAD}(\mathcal{D}) = \frac{\sum_{i=1}^{k} |C_i| \ell_i}{n}. \tag{2}$$

and

$$\texttt{WAES}(\mathcal{D}) = \frac{\sum_{\texttt{i=1}}^{\texttt{k}} |\texttt{C}_\texttt{i}| \ell_\texttt{i}^{\texttt{nr}}}{\texttt{n}}, \tag{3}$$

where l_i and ℓ_i^{nr} are, respectively, the number of conditions and non-redundant conditions (w.r.t. C_i) in the path from the root to leaf i.

In terms of explainability, a decision tree is a single structure that allows us to visualize explanations for all clusters (some of them potentially having redundant conditions), and WAD gives the average length (weighted by the cluster's sizes) of these explanations. For each specific cluster, however, we may derive more compact explanations by removing redundant conditions, and WAES measures the average size of these explanations, again weighted by the cluster's sizes.

The problem proposed in Dasgupta et al. [2020] is that of finding the partition that minimizes (1), among those that can be induced by a decision tree of k leaves. In addition to minimize (1), we also focus on building trees with low values for WAD (2) and WAES (3).

To accomplish our goal, we note that it is important to take into account both WAD and WAES during the decision tree construction, since the optimization of one metric does not imply on the optimization of the other. Indeed, in Appendix A, we show an example where the same partition can be induced by different decision trees that vary a lot with respect to our metrics.

We conclude this section by introducing terminologies and notations that will be useful throughout this paper. We use the term *explainable clustering* to refer to a clustering that it is induced by some axis-aligned decision tree. By an i-cut we mean a cut of the form (i, θ), that is, a cut for which component i is fixed. If a node in a decision tree is associated with an i-cut we say that it is an i-node.

2 Related work

Dasgupta et al. [2020] presents a poly-time algorithm, IMM, that receives a (non-explainable) partition $\mathcal{P}_u$ to the k-means clustering problem and builds a decision tree, in top-down fashion, by selecting at each node the cut that, among those that separate at least two representatives in $\mathcal{P}_u$, minimizes the number of data points separated from their representatives in $\mathcal{P}_u$. In addition, they prove that the cost of the resulting partition is $O(k^2)cost(\mathcal{P}_u)$. A consequence of this result is that the *price of explainability*, measured by the ratio between the cost of an optimal explainable partition and that of an optimal (non-explainable) one, is $O(k^2)$.

After Dasgupta et al. [2020], new algorithms, yielding to improved bounds on the price of explainability, were proposed [Laber and Murtinho, 2021, Makarychev and Shan, 2021a, Charikar and Hu, 2021, Esfandiari et al., 2021, Gamlath et al., 2021]. The best known upper bound, among those that only depend on k, is $O(k \log k)$ from Esfandiari et al. [2021]. We note that this bound is nearly tight since the same paper also provides an $\Omega(k)$ lower bound.

Empirical studies with algorithms for building explainable partitions can be found in Frost et al. [2020], Laber and Murtinho [2021]. The former proposes the ExKMC algorithm and compares it with IMM, CART [Breiman et al., 1984], KDTree [Bentley, 1975], CUBT [Fraiman et al., 2013], and CLTree [Liu et al., 2005]. One conclusion that can be drawn from this study is that IMM outperforms the other competitors when the objective is building trees with exactly k leaves. ExKMC, though being "defeated" by `IMM`, is not limited to building trees with k leaves, allowing partitions where the same cluster is associated with more than one leaf. This flexibility allows partitions with lower costs (though less explainable). An algorithm with provable guarantees for this scenario was recently obtained in Makarychev and Shan [2021b].

Laber and Murtinho [2021] introduce a simple greedy algorithm, ExGreedy, and show that it produces partitions with lower costs than those produced by IMM. We note that neither Frost et al. [2020] nor Laber and Murtinho [2021] analyze the produced trees in terms of their explainability. In our experiments we compare IMM, ExGreedy, ExKMC and the algorithm from Makarychev and Shan [2021a] against our method using different measures of explainability.

The aforementioned papers focus on the k-means clustering problem. However, a number of papers [Fraiman et al., 2013, Bertsimas et al., 2018, Saisubramanian et al., 2020] propose decision-tree algorithms to build partitions that optimize other measures.

Explainability and interpretability are topics of growing interest in the machine learning community [Ribeiro et al., 2016, Lundberg and Lee, 2017, Adadi and Berrada, 2018, Rudin, 2019, Murdoch et al., 2019, Molnar, 2020]. While there has been some focus on what Dasgupta et al. [2020] calls *post-modeling explainability*, or the ability to explain the output of a black-box model [Ribeiro et al., 2016, Lundberg and Lee, 2017, Kauffmann et al., 2019], the practice has also been criticized in contrast with *pre-modelling explainability*, or the use of interpretable models to begin with [Rudin, 2019]. The present work as the ones that follow Dasgupta et al. [2020] may be considered a middle-of-the-road approach, as the end result is a fully interpretable model (instead of, for instance, a model for locally interpreting the original model, or for explaining individual predictions) based on the output from a potentially black-box model (the unrestricted partition the decision tree aims to approximate).

3 On the complexity of building explainable k-means clustering

We prove that the problem of finding an explainable clustering with minimum cost is hard to approximate. The reduction employed here is the one used by Awasthi et al. [2015] to show that it is hard to find an $(1+\epsilon)$-approximation for the k-means clustering problem. The only difference in the proof is that we need to argue that the k-means clustering instance employed in the reduction admits solutions of low cost that can be induced by decision trees.

Theorem 1. *The problem of building an explainable clustering, via decision trees, that minimizes the $k-$means cost function does not admit an $(1+\epsilon)$-approximation in polynomial time unless $P = NP$.*

Proof. We use a result from Awasthi et al. [2015], which presents a polynomial-time reduction from the vertex-cover problem on triangle-free graphs to the k-means problem. In this reduction, given a graph $G = (V, E)$, where $V = \{1, \ldots, n\}$, every edge e in E is mapped onto a vector

$\mathbf{v}^e = (v_1^e, \ldots, v_n^e)$ in $\{0,1\}^n$ where $v_i^e = 1$ if vertex i is incident on e and $v_i^e = 0$ otherwise. It is proved that if the minimum vertex cover of G has size k, then the minimum cost of the corresponding k-means problem is at most $|E| - k$, and if the minimum vertex cover has size at least $(1+\epsilon)k$ then the minimum cost is at least $|E| - (1 - \Omega(\epsilon))k$. We show that the same reduction works when we are considering explainable clustering.

First, we show that, if the minimum vertex cover of G has size k, then there is an explainable clustering of cost at most $|E| - k$. Let $C = \{i_1, i_2, \ldots, i_k\}$ be a cover of size k for G, where each i_j is an integer in $[n]$ and $i_j < i_{j+1}$. We build a k-clustering $(E_1, \ldots, E_k)$ for the vectors $\{\mathbf{v}^e | e \in E\}$ as follows: the group E_j includes all vectors $\mathbf{v}$ that simultaneously satisfy: its component i_j is 1 and its component $i_{j'}$, for $j' < j$, is 0. This clustering can be obtained by a decision with $k-1$ levels, with one node per level, where the single node of level j is associated with cut $(i_j, 1/2)$.

Now we show that the cost of this clustering is at most $|E| - k$. Let $\ell_j = |E_j|$. The centroid of E_j has 1 at coordinate i_j and $1/\ell_j$ in the remaining ℓ_j coordinates with non-zero values. Thus, E_j contributes to the total cost with $\ell_j(1 - 1/\ell_j)^2$. The cost of the clustering $(E_1, \ldots, E_k)$ is, then, given by

$$\sum_{j=1}^{k} \ell_j \frac{(\ell_j - 1)^2}{\ell_j^2} = |E| - 2k + \sum_{j=1}^{k} \frac{1}{\ell_j} \leq |E| - k$$

Now, it remains to argue that if the minimum vertex cover has size at least $(1+\epsilon)k$ then every explainable clustering has cost at least $|E| - (1 - \Omega(\epsilon))k$. This follows from Awasthi et al. [2015], as in this case every clustering (and, in particular, every explainable one) has cost at least $|E| - (1 - \Omega(\epsilon))k$. □

The previous result motivates the quest for approximation algorithm as well as for heuristics, as the one we present in the next section.

4 A strategy for building shallow trees with low cost

Our strategy, denoted by $ExShallow$, builds a decision tree in a top-down fashion as shown in Algorithm 1. As an input the strategy receives a set of points $\mathcal{X}'$ and also a set $\mathcal{S}'$ of k representatives (denoted here by reference centers). We say that two cuts are equivalent with respect to set $\mathcal{X}' \cup \mathcal{S}'$ if they are associated with the same component (both are i-cuts fro some i) and if they induce the same binary partition on $\mathcal{X}' \cup \mathcal{S}'$. Note that there are at most $|\mathcal{X}' \cup \mathcal{S}'|d$ pairwise non-equivalent cuts. At each node the strategy selects the cut γ for which

$$\texttt{Price}(\gamma, \mathcal{X}', \mathcal{S}') + \lambda \cdot \texttt{DExp}(\gamma, \mathcal{X}', \mathcal{S}') \tag{4}$$

is minimum, among the non-equivalent cuts that separate at least two reference centers from $\mathcal{S}'$.

In Equation (4), $\texttt{Price}(\gamma,\mathcal{X}',\mathcal{S}')$ and $\texttt{DExp}(\gamma,\mathcal{X}',\mathcal{S}')$ (both detailed further below) estimate how good γ is for the goal of building a partition with low cost and with low values for $\texttt{WAES}/\texttt{WAD}$, respectively. We note the DExp stands for *Depth Explainability*. To simplify the notation, whenever the context is clear, we drop $\mathcal{X}'$ and $\mathcal{S}'$ from $\texttt{Price}()$ and $\texttt{DExp}()$. This is also done for other metrics introduced further.

After selecting γ, the strategy is recursively performed for each of the groups of the binary partition induced by γ. The recursion stops when $\mathcal{S}'$ contains only one reference center. The initial set of reference centers can be built by any algorithm for the (non-explainable) k-means clustering problem, such as Lloyd's algorithm [Lloyd, 1982].

Algorithm 1 EXSHALLOW ($\mathcal{X}'$: set of points; $\mathcal{S}'$: set of reference centers)

if $|S'| = 1$ **then**
  Return $\mathcal{X}'$ and the single reference center in S'
else
  $\mathcal{C} \leftarrow$ set of non-equivalent cuts w.r.t. $\mathcal{X}' \cup \mathcal{S}'$ that separate at least two centers in $\mathcal{S}'$
  $\gamma \leftarrow$ cut in $\mathcal{C}$ for which $\texttt{Price}(\gamma) + \lambda \cdot \texttt{DExp}(\gamma)$ is minimum
  $(\mathcal{X}'_L, \mathcal{X}'_R) \leftarrow$ partition of $\mathcal{X}'$ induced by γ
  $(\mathcal{S}'_L, \mathcal{S}'_R) \leftarrow$ partition of $\mathcal{S}$ induced by γ
  Create a node u
  $u.\texttt{LeftChild} \leftarrow \text{EXSHALLOW}(\mathcal{X}'_L, \mathcal{S}'_L)$
  $u.\texttt{RightChild} \leftarrow \text{EXSHALLOW}(\mathcal{X}'_R, \mathcal{S}'_R)$
  Return the tree rooted at u
end if

Let $\mathcal{X}'$ and $\mathcal{S}'$ be, respectively, the sets of points and centers that reach some given node in the decision tree. In addition, let γ be a cut that splits $\mathcal{X}'$ into groups ${\mathcal{X}'}_L$ and ${\mathcal{X}'}_R$ and splits $\mathcal{S}'$ into groups ${\mathcal{S}'}_L$ and ${\mathcal{S}'}_R$, each of them containing at least one reference center. $\texttt{Price}(\gamma)$ is given by $\texttt{InducedCost}(\gamma, \mathcal{X}', \mathcal{S}')/\texttt{CurrentCost}(\mathcal{X}', \mathcal{S}')$, where

$$\texttt{CurrentCost}(\mathcal{X}', \mathcal{S}') = \sum_{\mathbf{x} \in \mathcal{X}'} \min_{\mathbf{c} \in \mathcal{S}'} ||\mathbf{x} - \mathbf{c}||_2^2 \tag{5}$$

and

$$\texttt{Inducedcost}(\gamma, \mathcal{X}', \mathcal{S}') = \left(\sum_{\mathbf{x} \in \mathcal{X}'_{\mathcal{L}}} \min_{\mathbf{c} \in \mathcal{S}'_{\mathcal{L}}} ||\mathbf{x} - \mathbf{c}||_2^2 + \sum_{\mathbf{x} \in \mathcal{X}'_{\mathcal{R}}} \min_{\mathbf{c} \in \mathcal{S}'_{\mathcal{R}}} ||\mathbf{x} - \mathbf{c}||_2^2 \right); \tag{6}$$

that is, `CurrentCost` and `InducedCost` give, respectively, the cost of the partition before and after applying cut γ. In both cases, each point is associated with the closest valid reference center. We note that `InducedCost`() is the cost function used by the EXGREEDY algorithm proposed in Laber and Murtinho [2021] to select a cut at each node.

To obtain $\texttt{DExp}(\gamma, \mathcal{X}', \mathcal{S}')$, we first calculate $\texttt{WAD}(\gamma, \mathcal{X}', \mathcal{S}')$, an estimation of the quality of γ in terms of the weighted average depth, and then we adjust $\texttt{WAD}(\gamma)$ to take into account the metric `WAES`.

$\texttt{WAD}(\gamma)$ is given by the return of procedure $\texttt{EvalWAD}(|\mathcal{X}'|, |\mathcal{S}'|)$ presented in Algorithm 2. $\texttt{EvalWAD}(N, K)$ returns the weighted average depth of a tree with K leaves (corresponding to centers) for a set of N points, where each node in the tree splits the points and the centers in the same proportion as γ does, that is, proportionally to $r_{points} = |{\mathcal{X}'}_L|/|\mathcal{X}'|$ and $r_{center} = |{\mathcal{S}'}_L|/|\mathcal{S}'|$, respectively. We note that these ratios do not change along the algorithm execution and that the resulting decision tree is just a theoretical tree (which may not even be feasible for the instance under consideration), built to estimate how good the cut γ is for the goal of minimizing the `WAD`.

The value of $\texttt{DExp}(\gamma)$ is given by the return of procedure $\texttt{EvalDExp}(\gamma, \mathcal{X}', \mathcal{S}')$ presented in Algorithm 3. To explain the procedure, let v be the current node of the decision tree under construction. Recall that if a cut $\gamma = (i, \theta)$ is applied on v then it induces two edges leaving v, one associated with condition $x_i < \theta$ and the other with condition $x_i > \theta$. We say that an edge leaving v is *killer* if its associated condition turns some non-redundant condition in the path, from the root to v, into a redundant one. The procedure first determines which edges leaving v are killer and, based on that, it adjusts the value of $\texttt{WAD}(\gamma)$ to take into account the metric `WAES`. As an example, if only the left edge leaving v is killer then we discount $|\mathcal{X}'_L|/|\mathcal{X}'|$ from $\texttt{WAD}(\gamma)$ because one condition in the path from the root to v becomes redundant to explain the clusters of the left subtree of v.

By design, `DExp` prioritizes the choice of cuts at node v that are associated with coordinates that have already been used by some cut in the path from the root to v. This way the strategy tends to produce redundant conditions and, therefore, to minimize the `WAES`.

Algorithm 2 `EvalWAD`(N: Current number of points; K: Current number of reference centers)

if $K = 1$ **then**
  Return 0
else
  $K_L \leftarrow K \cdot r_{center}$
  $K_r \leftarrow K - K_L$
  $N_L \leftarrow N \cdot r_{points}$
  $N_R \leftarrow N - N_L$
  Return $1 + (N_L \cdot \texttt{EvalWAD}(N_L, K_L) + N_R \cdot \texttt{EvalWAD}(N_R, K_R))/N$
end if

Algorithm 3 `DExp`(γ: cut; $\mathcal{X}'$: set of points; $\mathcal{S}'$: set of centers)

if no edge induced by γ on v is killer **then**
  Return `EvalWAD`$(|\mathcal{X}'|, |\mathcal{S}'|)$
else if only the left edge induced by γ on v is killer **then**
  Return `EvalWAD`$(|\mathcal{X}'|, |\mathcal{S}'|)$-$|\mathcal{X}'_L|/|\mathcal{X}'|$
else if only the right edge induced by γ on v is killer **then**
  Return `EvalWAD`$(|\mathcal{X}'|, |\mathcal{S}'|)$-$|\mathcal{X}'_R|/|\mathcal{X}'|$
else
  Return `EvalWAD`$(|\mathcal{X}'|, |\mathcal{S}'|)$-1
end if

4.1 Setting the trade-off parameter

In a typical case, an user is interested in obtaining an explainable clustering with low cost. To achieve this goal she/he has to properly set the value of λ. One possibility is performing a brute-force search over some set of values to find the one that yields the most suitable tree. However, this could be computationally expensive and also non-practical from the user perspective, as she/he would have to analyze many trees. Fortunately, as we explain, we can avoid that.

First we note that a reasonable interpretation for λ is how much we are willing to (locally) give up of cost, in percentage, to reduce by one unit the average size of the explanations. As an example, setting $\lambda = 0.1$ means that we accept an additive loss of up to 10% in terms of the partition cost to have explanations one unit shorter on average.

Under this perspective, we shall avoid large values for λ, since partitions with high costs are not likely to produce coherent clusters, and making incoherent explainable clusters would be useless. In fact, as we show in our experiments, by setting λ close to 0.03 we obtain significant improvements over the existing methods.

We note that it is possible to successfully set λ to a constant value (e.g. 0.03) because we work with `Price`() in our cost function. For the sake of contrasting, if we worked with `InducedCost`(), which is arguably a more natural measure than `Price`(), we would have difficulties in establishing a trade-off with `DExp`(). For example, is an increment of 5 units in `InducedCost`() a small increment, or a large one? Is it worth allowing this increment to reduce `DExp`() by one unit? It is not clear how we would answer these questions.

Another advantage of `Price`() is that its value for cuts of low `InducedCost`() (the most relevant ones) lies in the interval $[1, 8k + 2]$, the same one in which both `WAES` and `WAD` lie, except for a constant factor. As a consequence, we are trading off quantities with similar magnitudes, which is beneficial. We finish this section by formalizing the observation above about the range of `Price`().

Lemma 1. *Let $\mathcal{X}'$ and $\mathcal{S}'$ be the set of data points and reference centers that reach a given node v. Then, there is a cut γ' that satisfies $1 \leq \texttt{Price}(\gamma', \mathcal{X}', \mathcal{S}') \leq 8k + 2$*

Proof. The lefthand side follows because any assignment between points and reference centers that is valid after applying a cut is also valid before the cut, so that $\texttt{CurrentCost}(\mathcal{X}', \mathcal{S}') \leq \texttt{InducedCost}(\gamma, \mathcal{X}', \mathcal{S}')$, for every cut γ.

Regarding the righthand side, Theorem 5.1 of Dasgupta et al. [2020] shows that if `IMM` (the algorithm proposed in the paper) builds a tree with height H, then the cost of the partition induced by this tree

is at most $8Hk + 2$ times larger than that of the initial partition (the one from which the decision tree is built upon). More specifically, the proof shows that the cost of the partition associated with each level of the tree is at most $8k$ times larger than that of the initial partition.

In terms of our scenario, it suffices to consider the initial partition provided by `IMM` as the one before applying the cut at node v and the partition induced by level 1 of the `IMM`'s tree as the partition obtained due to the application of a cut at v. Thus, we conclude that `InducedCost`(γ')`/CurrentCost` is at most $8k + 2$, where γ' is the first cut employed by `IMM`.

□

4.2 Implementation details and time-complexity analysis

Given the set of points $\mathcal{X}$ and the reference centers $\mathcal{S}$, the algorithm first obtains d sorted lists, where the i-th list corresponds to the set of points in $\mathcal{X} \cup \mathcal{S}$ sorted by component i. This initial sorting step takes $O(d(n + k) \log(n + k))$ time and it is only performed in the root of the tree.

Having the d sorted lists at node v, it is shown in Laber and Murtinho [2021] that (6) can be computed for all valid cuts in $\mathcal{O}(dn_v k_v)$ time, where n_v and k_v are, respectively, the number of points and centers that reach v. In addition, the computation of `WAD`(), via Algorithm 2, takes $O(k_v)$ time per cut and, then, $O(dn_v k_v)$ time for all cuts.

To find out which of the edges are killer in Algorithm 3, we maintain a data structure, namely `A`, with $2d$ entries. For each $i \in [d]$, `A`[i].left (resp. `A`[i].right) stores the number of left (resp. right) edges that leave i-nodes that lie in the path from the root to the current node. To determine if a left (resp. right) edge leaving an i-node is killer we test whether `A`[i].left>0 (resp. `A`[i].right>0) or not. In the positive case the edge is killer, otherwise it is not.

The data structure `A` can be updated in $O(1)$ time: if the chosen cut at node v is an i-cut, then right before the recursive call at line 9 (resp. line 10) of ExShallow we increment by one unit `A`[i].left (resp. `A`[i].right), and when we return from the recursion we decrease the respective counter by 1.

After selecting the cut at node v, the d sorted lists for the children of v are obtained in $O(n_v d)$ time from the sorted lists for v.

Thus, the total cost of the algorithm to build a tree $\mathcal{D}$ is proportional to

$$\sum_{v \in \mathcal{D}} n_v \cdot d \cdot k_v \leq \sum_{i=1}^{n} \ell_i \cdot d \cdot k,$$

where ℓ_i is the depth of data point i at $\mathcal{D}$. The rightmost term, however, is equal to $\texttt{WAD}(\mathcal{D}) \cdot \texttt{n} \cdot \texttt{d} \cdot \texttt{k}$.

The $\mathcal{O}(\texttt{WAD}(\mathcal{D}) \cdot \texttt{n} \cdot \texttt{d} \cdot \texttt{k})$ time complexity suggests that trees with low `WAD` are faster to build – which is good for our purposes, since by design our algorithm tries to build trees with this property.

5 Experiments

In this section we report our experimental study. We have two goals: understanding the impact of λ and comparing our strategy with other available proposals for building explainable clustering, more specifically, IMM, ExKMC, and ExGreedy algorithms from Dasgupta et al. [2020], Frost et al. [2020], Laber and Murtinho [2021], respectively. These methods start with the reference centers of a partition for the unrestricted k-means clustering problem and, then, build a tree in a top-down fashion by selecting at each node a cut that separates at least two reference centers. What distinguishes them is the strategy employed to choose the cut:

- IMM selects the cut that minimizes the number of data points separated from their representatives;
- ExKMC selects the cut that minimizes the overall k-means cost of the split when a single center (chosen from the original centers of the unrestricted solution) is assigned to all points in each side of the cut;
- ExGreedy, as already mentioned, selects the cut that minimizes the `InducedCost` given by Equation (6);

Table 1: Dataset summary: n is the number of data points, d is the dimension, and k is the number of desired clusters.

Dataset	n	d	k	**Source**
20Newsgroups	18,846	1,069	20	`http://qwone.com/~jason/20Newsgroups/`
Anuran	7,195	22	10	UCI
Avila	20,867	10	12	UCI [De Stefano et al., 2018]
Beer	1,514,999	5	104	OpenML
BNG (audiology)	1,000,000	85	24	OpenML
Cifar10	60,000	3,072	10	Krizhevsky et al. [2009]
Collins	1,000	19	30	OpenML
Covtype	581,012	54	7	OpenML [Collobert et al., 2002]
Digits	1797	64	10	UCI [Alpaydin and Kaynak, 1998]
Iris	150	4	3	UCI [Fisher, 1936]
Letter	20,000	16	26	Hsu and Lin [2002]
Mice	552	77	8	OpenML [Higuera et al., 2015]
Pendigits	10,992	16	10	UCI
Poker	1,025,010	10	10	UCI
Sensorless	58,509	48	11	UCI
Vowel	990	10	11	UCI

- The algorithm from Makarychev and Shan [2021a] selects at each node a random cut from the bounding box induced by the set of k reference centers. To avoid dense tables, the comparison with this algorithm was moved to the appendix.

In our evaluation, we considered 16 datasets of different sizes and characteristics, performing 10 or 30 seeded iterations in each of them, depending on the experiment. For each iteration, we find an unrestricted partition of the data by running Lloyd's algorithm [Lloyd, 1982] with the ++ initialization [Arthur and Vassilvitskii, 2007], as implemented in Python's `scikit-learn` package (Pedregosa et al. [2011], released under the BSD 3-Clause License). This unrestricted partition is provided to IMM and to EXKMC, as implemented in the `ExKMC` package (Frost et al. [2020], released under the MIT License), and to EX-GREEDY, implemented as an extension of the `ExKMC` package and available in `https://github.com/lmurtinho/ExKMC` (Laber and Murtinho [2021]). Then we provide the same unrestricted partition to EXSHALLOW.

Table 1 presents the size, dimension, and number of classes (which we use as the number of clusters) of the datasets in which we perform the experiments. All datasets are available online, and our code includes a script for retrieving and running tests on them. The number of instances, dimensions, and features is that of the final dataset used in our experiments (after removal of missing values and one-hot encoding of categorical variables, for instance). Most datasets are retrieved from OpenML [Vanschoren et al., 2013] or UCI [Dua and Graff, 2017]. All datasets are anonimized and present no offensive content.

5.1 Results

5.1.1 Comparison of EXSHALLOW with other explainable clustering algorithms

Tables 2 and 3 show the average weighted depths, explanation sizes, and partition costs for the 16 datasets and for 4 different explainable clustering algorithms: EXSHALLOW with $\lambda = 0.03$ (EXSHALLOW), EX-GREEDY [Laber and Murtinho, 2021], IMM [Dasgupta et al., 2020], and EXKMC [Frost et al., 2020]. For each dataset, we ran 30 seeded iterations of Lloyd's algorithm, and used the resulting unexplainable partition as a starting point for each explainable clustering algorithm analyzed here.

We also performed statistical tests (one-sided t-tests, assuming the same variance for both distributions, and with a confidence level of 95%) to check the statistical significance of the difference between results from EXSHALLOW and each of the other algorithms. Values in red (resp. blue) in Tables 2 and 3 indicate that results for the algorithm in question are worse (resp. better) on average than those of EXSHALLOW with a confidence level of 95%. For instance, although KMC's average `WAD` for

	WAD				normalized partition cost			
dataset	ExShallow	Ex-Greedy	IMM	KMC	ExShallow	Ex-Greedy	IMM	KMC
anuran	3.78	4.46	5.75	**3.44**	1.16	**1.15**	1.28	1.32
avila	4.44	6.53	6.51	**4.43**	1.05	**1.05**	1.06	1.17
beer	10.42	15.11	54.97	**7.39**	**1.17**	1.18	1.86	1.27
bng	**3.46**	5.40	13.79	4.61	1.05	**1.02**	1.04	1.03
cifar10	**3.37**	3.60	5.71	3.63	**1.16**	1.17	1.22	1.19
collins	5.80	15.17	16.67	**5.76**	1.18	**1.17**	1.22	1.22
covtype	3.15	3.56	3.55	**2.82**	1.03	1.03	**1.03**	1.13
digits	3.96	5.65	5.60	**3.80**	**1.19**	1.21	1.24	1.22
iris	**1.67**	**1.67**	**1.67**	**1.67**	**1.04**	**1.04**	**1.04**	**1.04**
letter	**5.49**	12.53	14.97	5.50	**1.19**	1.23	1.30	1.38
mice	3.29	3.54	3.70	**3.14**	**1.07**	1.08	1.12	1.16
newsgroups	**1.37**	15.70	15.73	13.74	1.05	**1.01**	1.01	1.01
pendigits	3.76	4.46	4.44	**3.53**	**1.14**	1.14	1.25	1.32
poker	3.35	3.36	3.36	**3.23**	1.10	1.10	**1.10**	1.12
sensorless	**3.86**	4.50	4.40	4.06	1.02	**1.02**	1.03	1.07
vowel	3.92	5.66	6.21	**3.61**	**1.20**	1.25	1.34	1.28
Medians	3.77	4.95	5.73	3.72	1.12	1.12	1.17	1.18

Table 2: Average weighted depth and normalized partition cost for all datasets and algorithms. ExShallow is run with $\lambda = 0.03$. Best results for each dataset are in bold. Values in red (blue) are statistcally larger (smaller) than those of ExShallow with a confidence level of 95% (see text for details on the statistical tests performed).

`Avila` is smaller than ExShallow 's, the difference is not statistically significant considering the 30 iterations performed.

The partition costs are normalized by the cost of the unrestricted partition used as a starting point for the explainable clustering algorithms. For ease of comparison, the normalized partition cost is presented in both tables.

In terms of average cost, ExShallow beats (with 95% confidence) Ex-Greedy in 9 datasets, IMM in 11 and KMC in 13. It is beaten by at least one algorithm on 4 datasets, in two of them by less than 1%. Only for `bng` and `newsgroups` the partitions generated by ExShallow are clearly worse (by at most 4%), and for both datasets ExShallow returns partitions that are much more explainable (in terms of `WAD` and `WAES`) than those of the other algorithms.

In terms of `WAD` and `WAES`, Algorithm 1 outperforms Ex-Greedy and IMM on 15 and 14 datasets, respectively. For many datasets it is beaten by KMC by a small margin and, when this happens, Algorithm 1 almost always beats KMC in terms of partition cost, frequently by large margins. Observe the median of `WAD` and `WAES` in the last lines of Tables 2 and 3

In summary, our experiments suggest that

- IMM and KMC are the worst performers in terms of partition cost. IMM is also the worst in terms of explainability;
- KMC tends to sacrifice partition quality in exchange for more explainable clusters (which could also be done with ExShallow by selecting a larger value for λ);
- Ex-Greedy typically achieves partitions with similar costs as ExShallow , but with less explainable clusters.
- ExShallow is almost always at least close to the best result in terms of both partition cost and explainability, and frequently has a significant advantage in at least one of these dimensions when compared to the other 3 algorithms (as can be seen in the results for `Avila`, `Collins`, `Letter`, and `Pendigits`, for instance).

To illustrate the last point made above, we present in Figure 2 two trees generated by ExShallow and Ex-Greedy for the Avila dataset, starting from the same unexplainable partition. Both have

	WAES				normalized partition cost			
dataset	ExShallow	Ex-Greedy	IMM	KMC	ExShallow	Ex-Greedy	IMM	KMC
anuran	3.75	4.34	5.45	**3.44**	1.16	**1.15**	1.28	1.32
avila	3.74	5.46	5.07	**3.25**	1.05	**1.05**	1.06	1.17
beer	7.31	8.04	7.71	**6.29**	**1.17**	1.18	1.86	1.27
bng	**3.46**	5.40	9.61	4.61	1.05	**1.02**	1.04	1.03
cifar10	**3.37**	3.60	5.71	3.63	**1.16**	1.17	1.22	1.19
collins	**5.42**	12.85	12.69	5.53	1.18	**1.17**	1.22	1.22
covtype	2.61	2.62	2.61	**2.45**	1.03	1.03	**1.03**	1.13
digits	3.96	5.65	5.60	**3.80**	**1.19**	1.21	1.24	1.22
iris	1.67	1.67	**1.44**	**1.44**	**1.04**	**1.04**	**1.04**	**1.04**
letter	**5.29**	11.37	12.72	5.38	**1.19**	1.23	1.30	1.38
mice	3.17	3.39	3.54	**3.11**	**1.07**	1.08	1.12	1.16
newsgroups	**1.28**	15.70	15.73	13.74	1.05	**1.01**	1.01	1.01
pendigits	3.67	4.42	4.29	**3.49**	**1.14**	1.14	1.25	1.32
poker	3.35	3.36	3.36	**3.23**	1.10	1.10	**1.10**	1.12
sensorless	**3.08**	4.28	4.15	3.98	1.02	**1.02**	1.03	1.07
vowel	3.84	5.09	5.65	**3.61**	**1.20**	1.25	1.34	1.28
Medians	3.57	4.76	5.53	3.62	1.12	1.12	1.17	1.18

Table 3: Average weighted depth and normalized partition cost for all datasets and algorithms. ExShallow is run with $\lambda = 0.03$. Best results for each dataset are in bold. Values in red (blue) are statistcally larger (smaller) than those of ExShallow with a confidence level of 95% (see text for details on the statistical tests performed).

essentially the same partition cost, but ExShallow generates a more balanced tree with a smaller weighted depth.

The maximum depth of the Ex-Greedy tree is 7, against 5 for the ExShallow tree. Furthermore, the largest cluster generated by ExShallow , with 6,861 elements, has depth 4 and is explained by 4 cuts; the largest cluster from the Ex-Greedy tree has depth 7 and is explained by 6 cuts (cut $E > 0.414$ at depth 3 is made redundant by cut $E > 0.555$ at depth 4). The second largest cluster from ExShallow , with 3,521 elements, is defined by 2 cuts ($RN > 0.716$ and $E > 0.662$, with cut $E > 0.547$ made redundant by the latter); the second largest cluster from Ex-Greedy, with 3,696 elements, is defined by 6 cuts.

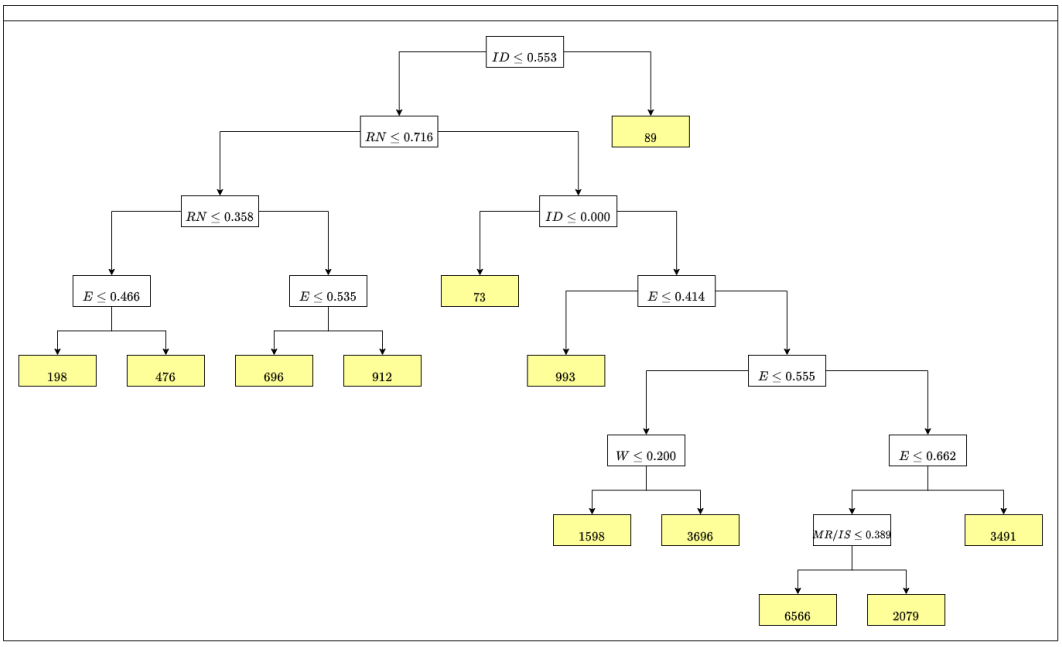
In Appendix B we present some additional information. We compare the algorithms with respect to the maximum and (non-weighted) average depth and we also show that ExShallow outperforms the algorithm from Makarychev and Shan [2021b] on our datasets.

5.2 Sensitivity of cost and weighted depth to variations in λ

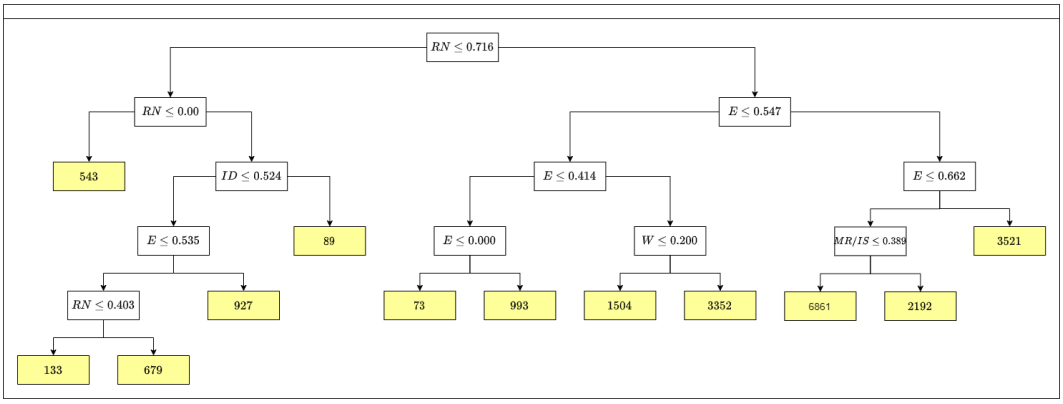
Figures 3a and 3b show, respectively, how the average weighted depth and the explanation size of the partitions produced by ExShallow changes as λ increases. To allow for a comparison between datasets, the values are normalized by those of the tree when $\lambda = 0$ (i.e., when depth is not taken into account by our cost function). For each dataset, we ran 10 seeded iterations of Lloyd's algorithm and used the resulting unexplainable partitions as a starting point for each instance of ExShallow with different values of λ.

ExShallow behaves as expected, with larger values of λ associated with shallower trees, and simpler explanations, on average. We observe a sharp drop for small values of λ. The value in red is 0.03, the one employed in the previous experiments.

Figure 3c shows how the mean cost of the partitions produced by our algorithm changes as λ increases. Again, to allow for a comparison between datasets, the costs are normalized, this time by the cost of the unrestricted partition generated by Lloyd's algorithm. And again, results are generally as expected, with larger values of λ (which impose more of a restriction in terms of the depth of the trees) associated with higher costs. An important finding and, perharps surprising, is that the partition cost for small values of λ is as good as that for $\lambda = 0$.

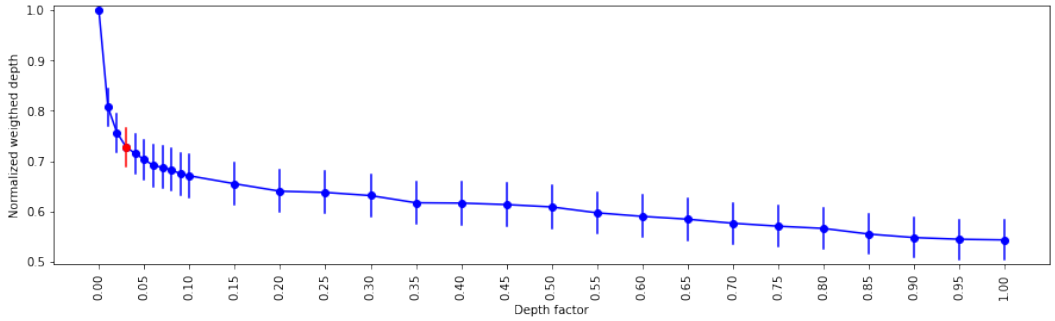


(a) Tree from the EXGREEDY algorithm

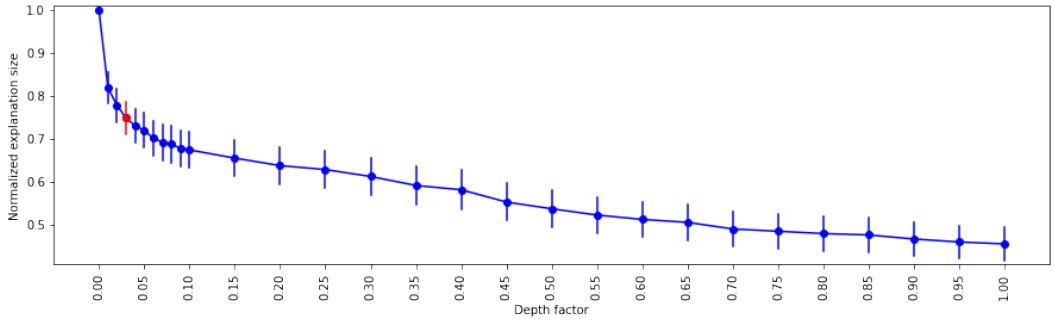


(b) Tree from EXSHALLOW

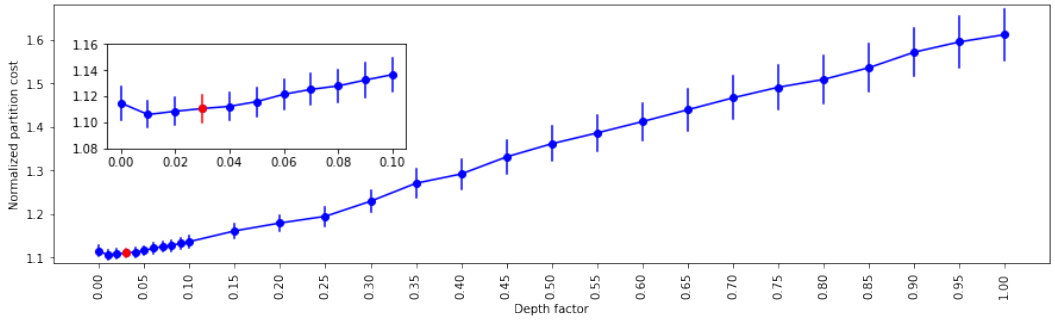
Figure 2: Two trees for partitioning the `Avila` dataset into 12 clusters. The letters inside the nodes correspond to the initials of the attributes' names: E=Exploitation; W=Weight; ID=Intercolumnar Distance; RN=Row Number; MR/IS=Modular Ratio/Interlinear Spacing.



(a) Mean normalized `WAD` per depth factor (for all datasets)



(b) Mean normalized `WAES` per depth factor (for all datasets)



(c) Mean normalized partition cost per depth factor (for all datasets)

Figure 3: Mean normalized depth, explanation size, and partition cost per depth factor λ, for all datasets. Depths and explanation sizes are normalized by results for $\lambda = 0$; partition costs are normalized by the cost of the unrestricted partition used as basis for finding the explainable partition. Error bars (with a confidence interval of 95%) are calculated using Python's `scipy` package [Virtanen et al., 2020].

dataset	ExShallow	Ex-Greedy	IMM	KMC	kmeans
anuran	0.27	0.27	0.09	0.16	0.96
avila	0.34	0.43	0.15	0.32	1.35
beer	6.07	6.77	29.28	34.32	307.28
bng	132.48	168.20	37.39	52.01	242.73
cifar10	186.85	199.17	67.64	83.81	145.31
collins	0.09	0.16	0.02	0.07	0.81
covtype	21.50	23.80	5.52	10.59	15.95
digits	0.17	0.21	0.05	0.09	0.92
iris	0.00	0.00	0.00	0.00	0.02
letter	0.64	1.38	0.25	0.39	7.85
mice	0.07	0.10	0.01	0.03	0.38
newsgroups	10.00	58.60	6.94	12.73	24.67
pendigits	0.22	0.27	0.09	0.17	1.04
poker	13.24	13.34	3.41	10.05	28.61
sensorless	3.95	4.45	0.93	1.81	2.17
vowel	0.03	0.03	0.01	0.01	0.66

Table 4: Average running times (in seconds) for each algorithm and dataset, including LLoyd's algorithm (KMEANS), which is used to find the unexplainable partition used as a starting point for the explainable clustering algorithms.

Combining these figures leads to the empirical conclusion that working with a small λ is very beneficial, as it significantly reduces the average weighted depth and explanation size without increasing the average cost of the partition.

5.3 Running times

Table 4 presents the average running times, over 30 seeded iterations, for each dataset and algorithm – including Lloyd's algorithm (KMEANS), which finds the unexplainable partition used as a starting point for all four explainable clustering algorithms analyzed here. For most datasets, KMEANS takes the largest time to return a partition among all algorithms analzyed; the exceptions are `cifar10`, `covtype`, `sensorless`, and `20newsgroups` – EX-GREEDY takes longer than KMEANS in all 4, and EXSHALLOW takes longer than KMEANS in the first 3.

Regarding only the explainable clustering algorithms, EX-GREEDY is the slowest one for all datasets except `beer`, for which both IMM and KMC are slower. EXSHALLOW 's running times are typically closer to those of EX-GREEDY than those of IMM and KMC, which tend to be faster. Overall, we do not perceive running time to be a significant hindrance in choosing EXSHALLOW over the other explainable clustering algorithms analyzed here, particularly due to the overhead imposed by having to run Lloyd's algorithm before any of them.

6 Conclusions

We discussed how explainable an "explainable partition" actually is, by analyzing the average depth of its underlying decision tree and the average number of rules needed to explain each cluster in the partition (both metrics being weighted by the number of points assigned to each leaf/cluster). In previous works on explainable clustering via decision trees, this aspect of the problem was largely ignored, except insofar as the trees may have no more than k leaves.

Introducing this preoccupation leads to a quantification of the explainability goal: we may desire, for instance, the best possible partition so that no more than h rules are used, on average, to define the cluster of a data point; or, conversely, we may wish to find the smallest number of rules needed (on average) to explain a partition, given that the cost of this partition does not exceed some predefined threshold.

We present an algorithm that works this trade-off by seeking to minimize both the cost of the resulting explainable partition and metrics related to how explainable this partition is – namely, the weighted depth of the leaves in the tree (which gives us an idea of the overall explainability of the partition)

and the weighted number of rules needed to explain each cluster in the partition (which gives us an idea of how explainable each individual cluster is). Our experiments on 16 datasets of different sizes and numbers of classes show that our algorithm typically produces partitions that are at least as good (many times better) then those obtained by recently published works on the topic.

We conclude the paper mentioning that we believe that the techniques developed here can be naturally adapted to build explainable partitions that aim to minimize other metrics as the relevant k-median cost function.

References

A. Adadi and M. Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018. doi: 10.1109/ACCESS.2018.2870052.

Ethem Alpaydin and Cenk Kaynak. Cascading classifiers. *Kybernetika*, 34(4):369–374, 1998.

David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245.

Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The hardness of approximation of euclidean k-means. In Lars Arge and János Pach, editors, *31st International Symposium on Computational Geometry, SoCG 2015, June 22-25, 2015, Eindhoven, The Netherlands*, volume 34 of *LIPIcs*, pages 754–767. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi: 10.4230/LIPIcs.SOCG.2015.754. URL `https://doi.org/10.4230/LIPIcs.SOCG.2015.754`.

Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

Dimitris Bertsimas, Agni Orfanoudaki, and Holly Wiberg. Interpretable clustering via optimal trees, 2018.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

Moses Charikar and Lunjia Hu. Near-optimal explainable k-means for all dimensions. *arXiv preprint arXiv:2106.15566*, 2021.

Ronan Collobert, Samy Bengio, and Yoshua Bengio. A parallel mixture of svms for very large scale problems. *Neural computation*, 14(5):1105–1114, 2002.

Sanjoy Dasgupta, Nave Frost, Michal Moshkovitz, and Cyrus Rashtchian. Explainable k-means and k-medians clustering. *Proceedings of the International Conference on Machine Learning*, 2020.

Claudio De Stefano, Marilena Maniaci, Francesco Fontanella, and A Scotto di Freca. Reliable writer identification in medieval manuscripts through page layout features: The "avila" bible case. *Engineering Applications of Artificial Intelligence*, 72:99–110, 2018.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. Almost tight approximation algorithms for explainable clustering. 2021.

Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7 (2):179–188, 1936.

Ricardo Fraiman, Badih Ghattas, and Marcela Svarc. Interpretable clustering using unsupervised binary trees. *Advances in Data Analysis and Classification*, 7(2):125–145, 2013.

Nave Frost, Michal Moshkovitz, and Cyrus Rashtchian. Exkmc: Expanding explainable k-means clustering. *arXiv preprint arXiv:2006.02399*, 2020.

Buddhima Gamlath, Xinrui Jia, Adam Polak, and Ola Svensson. Nearly-tight and oblivious algorithms for explainable clustering. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Clara Higuera, Katheleen J Gardiner, and Krzysztof J Cios. Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PloS one*, 10(6):e0129126, 2015.

Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

Jacob Kauffmann, Malte Esders, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. From clustering to cluster explanations via neural networks. *arXiv preprint arXiv:1906.07633*, 2019.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Eduardo Laber and Lucas Murtinho. On the price of explainability for some clustering problems. *arXiv preprint arXiv:2101.01576*, 2021.

Bing Liu, Yiyuan Xia, and Philip S Yu. Clustering via decision tree construction. In *Foundations and advances in data mining*, pages 97–124. Springer, 2005.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

Konstantin Makarychev and Liren Shan. Near-optimal algorithms for explainable k-medians and k-means. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 7358–7367. PMLR, 2021a. URL `http://proceedings.mlr.press/v139/makarychev21a.html`.

Konstantin Makarychev and Liren Shan. Explainable k-means. don't be greedy, plant bigger trees! *arXiv preprint arXiv:2111.03193*, 2021b.

Christoph Molnar. *Interpretable Machine Learning*. Lulu.com, 2020.

W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Sandhya Saisubramanian, Sainyam Galhotra, and Shlomo Zilberstein. Balancing the tradeoff between clustering value and interpretability. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 351–357, 2020.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198. URL `http://doi.acm.org/10.1145/2641190.2641198`.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

A Same Partition with Different Decision Trees

We show that the same partition can be induced by distinct decision trees that differ a lot in terms of explainability (according to our metrics). Let X_i be the set of $2^{2i} - 1$ points in R^k, where the jth point has all its k components equal to $2^{2i} + j$. Let $\mathcal{X} = X_1 \cup \ldots \cup X_k$. Clearly the optimal unrestricted k-partition for $\mathcal{X}$ is $(X_1, \ldots, X_k)$. This partition can be induced by many decision trees as the following ones that have only one node per level:

- $\mathcal{D}_1$: the cut at level i is $(i, 2^{2i+1})$ so that both $\texttt{WAD}(\mathcal{D}_1)$ and $\texttt{WAES}(\mathcal{D}_1)$ are $\approx k$;
- $\mathcal{D}_2$: the cut at level i is $(1, 2^{2(k-i)+1})$ so that both $\texttt{WAD}(\mathcal{D}_2)$ and $\texttt{WAES}(\mathcal{D}_2)$ are $O(1)$;
- $\mathcal{D}_3$: the cut at level i is $(i, 2^{2(k-i)+1})$ so that $\texttt{WAD}(\mathcal{D}_3)$ is $O(1)$ and $\texttt{WAES}(\mathcal{D}_3)$ is $\approx k$
- $\mathcal{D}_4$: the cut at level i is $(1, 2^{2i+1})$ so that $\texttt{WAD}(\mathcal{D}_4)$ is $\approx k$ and $\texttt{WAES}(\mathcal{D}_4)$ is $O(1)$

The message here is that we shall consider both WAES and WAD while building the tree, otherwise we can end up with a tree that performs poorly with respect to one of the metrics.

B Experiments - Additional Information

B.1 Additional metrics: maximum and average depths

Table 5 shows the average maximum depth, across 30 seeded iterations, for all datasets and explainable clustering algorithms analyzed in this paper. The average tree produced by ExShallow is usually shallower, or at least as shallow, as those produced by both Ex-Greedy and IMM. KMC, on the other hand, frequently produces trees that are shallower on average than those of ExShallow, but, as already discussed, the resulting partitions also tend to have a higher average cost.

Table 6 shows the average depths of the trees for each dataset and algorithm, without weighting the depth of each leaf by the number of elements in the corresponding cluster. As for the maximum depth, ExShallow on average produces trees whose leaves tend to be shallower than those of Ex-Greedy and IMM, and somewhat deeper than those of KMC.

B.2 Additional algorithm

Table 7 compares the results (in terms of WAD, WAES, and the normalized cost partition) of ExShallow (with $\lambda = 0.03$) and the algorithm from Makarychev and Shan [2021b]. We ran 10 seeded iterations of each algorithm for each dataset. ExShallow presents better average WAD in 13 of the 16 datasets analyzed, and better WAES in 10 of them; and, when beaten in explainability metrics, it is generally by small margins. In terms of cost, ExShallow is better in 14 of the 16 datasets, frequently by very large margins. The median results, shown at the bottom of Table 7 confirms that ExShallow tends to perform better than the algorithm from Makarychev and Shan [2021b] across all 3 metrics.

	maximum depth				normalized partition cost			
dataset	ExShallow	Ex-Greedy	IMM	KMC	ExShallow	Ex-Greedy	IMM	KMC
anuran	5.20	5.63	7.47	4.70	1.16	1.15	1.28	1.32
avila	5.43	7.83	7.77	5.00	1.05	1.05	1.06	1.17
beer	15.00	19.77	66.63	10.43	1.17	1.18	1.86	1.27
bng	14.23	10.57	20.10	7.33	1.05	1.02	1.04	1.03
cifar10	4.00	4.00	9.00	5.00	1.16	1.17	1.22	1.19
collins	8.70	22.60	26.00	8.50	1.18	1.17	1.22	1.22
covtype	4.00	4.00	4.00	4.00	1.03	1.03	1.03	1.13
digits	5.97	7.97	8.33	5.00	1.19	1.21	1.24	1.22
iris	2.00	2.00	2.00	2.00	1.04	1.04	1.04	1.04
letter	8.00	16.67	21.70	8.90	1.19	1.23	1.30	1.38
mice	4.53	4.90	5.17	4.03	1.07	1.08	1.12	1.16
newsgroups	10.10	18.90	19.00	18.77	1.05	1.01	1.01	1.01
pendigits	5.00	6.27	7.00	4.23	1.14	1.14	1.25	1.32
poker	4.00	4.00	4.00	4.00	1.10	1.10	1.10	1.12
sensorless	4.53	5.77	5.50	4.97	1.02	1.02	1.03	1.07
vowel	5.20	7.17	8.67	4.47	1.20	1.25	1.34	1.28

Table 5: Average maximum depth and normalized partition cost per dataset and algorithm. EXSHALLOW is run with $\lambda = 0.03$.

	average depth				normalized partition cost			
dataset	ExShallow	Ex-Greedy	IMM	KMC	ExShallow	Ex-Greedy	IMM	KMC
anuran	3.74	4.03	4.93	3.53	1.16	1.15	1.28	1.32
avila	3.92	4.84	4.93	3.92	1.05	1.05	1.06	1.17
beer	9.58	13.30	43.52	7.18	1.17	1.18	1.86	1.27
bng	7.87	6.22	11.64	5.11	1.05	1.02	1.04	1.03
cifar10	3.50	3.60	5.40	3.60	1.16	1.17	1.22	1.19
collins	5.62	12.55	13.98	5.57	1.18	1.17	1.22	1.22
covtype	3.00	3.14	3.14	3.00	1.03	1.03	1.03	1.13
digits	3.99	5.19	5.20	3.79	1.19	1.21	1.24	1.22
iris	1.67	1.67	1.67	1.67	1.04	1.04	1.04	1.04
letter	5.39	10.46	12.16	5.48	1.19	1.23	1.30	1.38
mice	3.27	3.45	3.52	3.17	1.07	1.08	1.12	1.16
newsgroups	6.73	10.40	10.45	10.30	1.05	1.01	1.01	1.01
pendigits	3.62	4.00	4.22	3.45	1.14	1.14	1.25	1.32
poker	3.40	3.40	3.40	3.40	1.10	1.10	1.10	1.12
sensorless	3.69	4.04	3.99	3.78	1.02	1.02	1.03	1.07
vowel	3.87	5.06	5.50	3.65	1.20	1.25	1.34	1.28

Table 6: Average depths and normalized partition costs for all datasets and algorithms. EXSHALLOW is run with $\lambda = 0.03$.

	WAD		WAES		normalized partition cost	
	ExShallow	Makarychev	ExShallow	Makarychev	ExShallow	Makarychev
anuran	**3.78**	4.26	**3.76**	4.14	**1.16**	1.67
avila	**4.45**	6.24	**3.72**	4.67	**1.05**	1.34
beer	**10.48**	11.56	**7.25**	7.45	**1.17**	1.54
bng	**3.43**	4.57	**3.43**	4.45	1.05	**1.04**
cifar10	**3.37**	3.50	**3.37**	3.50	**1.16**	1.25
collins	**5.87**	8.65	**5.56**	7.94	**1.17**	1.41
covtype	**3.15**	3.39	2.61	**2.60**	**1.03**	1.30
digits	3.96	**3.56**	3.96	**3.44**	**1.19**	1.43
iris	**1.67**	1.68	1.67	**1.49**	**1.04**	1.31
letter	**5.49**	7.88	**5.29**	7.13	**1.20**	1.53
mice	**3.27**	3.38	**3.11**	3.16	**1.07**	1.39
newsgroups	**1.44**	15.25	**1.35**	15.19	1.05	**1.01**
pendigits	3.78	**3.73**	3.71	**3.59**	**1.13**	1.74
poker	3.35	**3.26**	3.35	**3.18**	**1.10**	1.17
sensorless	**3.87**	4.54	**3.13**	4.32	**1.02**	1.26
vowel	**3.88**	3.94	3.80	**3.71**	**1.20**	1.53
median	**3.78**	4.10	**3.57**	3.93	**1.12**	1.37

Table 7: Comparison between results from EXSHALLOW (with $\lambda = 0.03$) and the algorithm from Makarychev and Shan [2021b] across 16 datasets (10 seeded iterations for each dataset).