# Explainable $k$-Means Clustering: Theory and Practice*

Sanjoy Dasgupta Nave Frost Michal Moshkovitz Cyrus Rashtchian

## Abstract

Despite the popularity of explainable AI, there is limited work on effective methods for unsupervised learning. We study algorithms for $k$-means clustering, where we use a small decision tree to partition a dataset into $k$ clusters. This enables us to explain each cluster assignment by a short sequence of single-feature thresholds. We present two explainable $k$-means clustering algorithms. First, the IMM algorithm produces a tree with $k$ leaves that induces an $O(k^2)$ approximation to the optimal $k$-means cost. Then, we develop a practical algorithm, ExKMC, that takes an additional parameter $k' \geq k$ and outputs a decision tree with $k'$ leaves. To improve the efficiency of ExKMC, we use a new surrogate cost to expand the tree and to label the leaves with one of $k$ clusters. We prove that as $k'$ increases, the surrogate cost is non-increasing, and hence, we trade explainability for accuracy. Empirically, we validate that both IMM and ExKMC produce low cost clusterings. We see that when ExKMC uses $4k$ leaves, it outperforms both standard decision tree methods and other algorithms for explainable clustering. Implementation of IMM and ExKMC available at https://github.com/navefr/ExKMC.

## 1. Introduction

Most research on explainable machine learning develops ways to interpret supervised methods, focusing on feature importance in black-box models (Arrieta et al., 2020; Deutch & Frost, 2019; Lipton, 2018; Lundberg & Lee, 2017; Molnar, 2019; Murdoch et al., 2019; Ribeiro et al., 2016; Rudin, 2019). To complement previous efforts, we study explainable algorithms for clustering, a canonical example of unsupervised learning. Clustering algorithms often operate iteratively, using global properties of the data to converge to a low-cost solution. For center-based clustering, the best explanation for a cluster assignment may simply be that an example is closer to some center than any others. While this type of explanation provides some insight, it obscures the impact of individual features, and the cluster assignments often depend on the data in a complicated way.

Recent work on explainable clustering goes one step further by enforcing that the clustering be derived from a binary threshold tree (Bertsimas et al., 2018; Chen et al., 2016; Fraiman et al., 2013; Ghattas et al., 2017; Liu et al., 2005). Each node is associated with a feature-threshold pair that recursively splits the dataset, and labels on the leaves correspond to clusters. Any cluster assignment can be explained by a small number of thresholds, each depending on a single feature. For large, high-dimensional datasets, this provides more information than typical clustering methods.

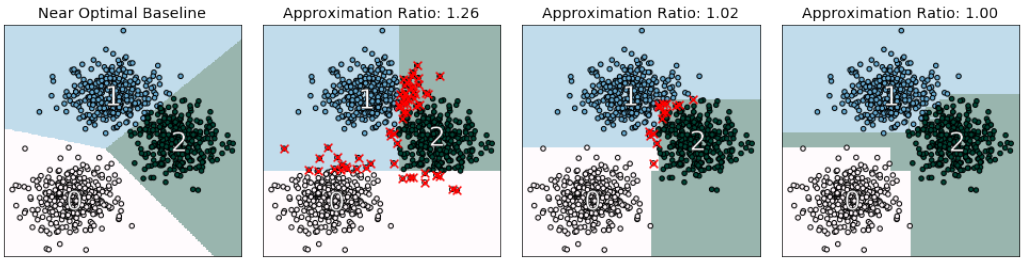
To make our study concrete, we focus on the $k$-means objective. The goal is to find $k$ centers that approximately minimize the sum of the squared distances between $n$ data points in $\mathbb{R}^d$ and their nearest center (Aggarwal et al., 2009; Aloise et al., 2009; Arthur & Vassilvitskii, 2007; Dasgupta, 2008; Kanungo et al., 2002; Ostrovsky et al., 2013).

We first present an explainable $k$-means algorithm, the Iterative Mistake Minimization (IMM) algorithm. It builds the smallest threshold tree (with $k$ leaves for $k$ clusters), and it achieves a worst-case $O(k^2)$ approximation to the optimal $k$-means cost. Prior to our work, no algorithms were known with approximation ratio independent of the dimension and dataset size. We also prove a lower bound showing that an $\Omega(\log k)$ approximation is necessary for exactly $k$ leaves.

Then, we propose an extension of IMM to expand the tree to use more leaves and achieve a better clustering. Our method, ExKMC, takes as input two parameters $k, k'$ and a set $\mathcal{X} \subseteq \mathbb{R}^d$ with $|\mathcal{X}| = n$. It first builds a threshold tree with $k$ leaves using the IMM algorithm. Then, given a budget of $k' > k$ leaves, it greedily expands the tree. At each step, the clusters form a refinement of the previous clustering. By adding more thresholds, we gain flexibility in the data partition, and we also allow multiple leaves to correspond to the same cluster (with $k$ clusters total).

To efficiently determine the assignment of leaves to clusters, we design and analyze a surrogate cost, which is non-increasing throughout the execution. The IMM algorithm first runs a standard $k$-means algorithm, producing a set of $k$ reference centers that are given as an additional input. As ExKMC expands the tree to $k' > k$ leaves, it minimizes the cost of the current clustering compared to the reference centers. By fixing the centers between steps, we can quickly compute the next feature-threshold pair to add. Finally, we label each leaf with the best reference center.

*Based on two papers (Dasgupta et al., 2020; Frost et al., 2020).




(a) $k$-means++ (reference) (b) $k$ leaves (IMM) (c) $2k$ leaves (ExKMC) (d) $n$ leaves

*Figure 1.* Tree size (explanation complexity) vs. $k$-means clustering quality.

Figure 1 depicts the improvement from using more leaves. The left picture shows a near-optimal 3-means clustering. Next, the IMM algorithm with $k = 3$ leaves leads to a large deviation from the reference clustering. Extending the tree to use $2k = 6$ leaves with ExKMC leads to a lower-cost result that better approximates the reference clustering. We form three clusters by subdividing the previous clusters and mapping multiple leaves to the same cluster. Finally, trees with enough leaves can perfectly fit the reference clustering.

**Related Work.** We address the challenge of obtaining a low cost $k$-means clustering using a small decision tree. It is NP-hard to find the optimal $k$-means clustering (Aloise et al., 2009; Dasgupta, 2008) or a close approximation (Awasthi et al., 2015). Our approach has roots in work on clustering with unsupervised decision trees (Basak & Krishnapuram, 2005; Chang & Jin, 2002; De Raedt & Blockeel, 1997; Yasami & Mozaffari, 2010) and in literature on extending decision trees for tasks beyond classification (Geurts & Louppe, 2011; Geurts et al., 2007; Jernite et al., 2017; Louppe et al., 2013; Pliakos et al., 2018). Prior explainable clustering algorithms optimize different objectives than $k$-means, such as the Silhouette metric (Bertsimas et al., 2018), density measures (Liu et al., 2005), or interpretability scores (Saisubramanian et al., 2020). A localized version of the 1-means cost has been used for greedily growing the tree (Fraiman et al., 2013; Ghattas et al., 2017).

Clustering via trees is explainable by design. We contrast this with the indirect approach of clustering with a neural network and then explaining the network (Kauffmann et al., 2019). A generalization of tree-based clustering has been studied using rectangle-based models (Chen et al., 2016; Chen, 2018; Pelleg & Moore, 2001). Their focus differs from ours as they consider including external information, via a graphical model and performing inference with variational methods. Clustering after feature selection (Boutsidis et al., 2009; Cohen et al., 2015) or feature extraction (Becchetti et al., 2019; Boutsidis et al., 2014; Makarychev et al., 2019) reduces the number of features, but it does not lead to an explainable solution since it runs a non-explainable $k$-means algorithm on the reduced space.

### 1.1. Our contributions

We present two new algorithms, IMM for building a tree with $k$ leaves that achieves an $O(k^2)$ approximation to the optimal $k$-means cost, and a practical extension, ExKMC that efficiently outputs an explainable $k$-means clustering with the following properties:

**Explainability-accuracy trade-off:** We provide a simple method that iteratively expands any threshold tree into a larger tree with a specified number of leaves. At each step, we aim to better approximate a given reference clustering (such as from a standard $k$-means implementation). The key idea is to minimize a surrogate cost that is based on the reference centers instead of using the $k$-means cost directly.

**Convergence:** We demonstrate empirically that ExKMC quickly converges to the reference clustering as the number of leaves increases. On many datasets, the cost ratio versus the reference clustering goes to 1.0 as the number of leaves goes from $k$ to $4k$, where $k$ is the number of labels for classification datasets. In theory, we prove that the surrogate cost is non-increasing throughout the execution of ExKMC, verifying that we trade explainability for clustering accuracy.

**Low cost:** Our most striking finding is that tree-based clusterings can match the cost of standard clusterings. This is possible with a small tree, even on large, high-dimensional datasets. ExKMC demonstrates that explanability can be obtained in conjunction with low cost on many datasets.

**Speed:** Using only standard optimizations, ExKMC can cluster fairly large datasets (e.g., CIFAR-10 or covtype) in under 15 minutes using a single processor, making it a suitable alternative to standard $k$-means in data science pipelines.

## 2. Preliminaries

We let $[n] = \{1, 2, \ldots, n\}$. For $k \geq 1$, a *$k$-clustering* refers to a partition of a dataset into $k$ clusters. Let $C^1, \ldots, C^k$ be a $k$-clustering of $\mathcal{X} \subseteq \mathbb{R}^d$ with $|\mathcal{X}| = n$ and $\boldsymbol{\mu}^j = \mathsf{mean}(C^j)$. The *$k$-means cost* is $\sum_{j=1}^{k} \sum_{\mathbf{x} \in C^j} \|\mathbf{x} - \boldsymbol{\mu}^j\|_2^2$.

**Explainable clustering.** Let $T$ be a binary threshold tree with $k' \geq k$ leaves, where each internal node contains a single feature $i \in [d]$ and threshold $\theta \in \mathbb{R}$. We also consider a labeling function $\ell : \text{leaves}(T) \to [k]$ that maps the leaves of $T$ to clusters. The pair $(T, \ell)$ induces a $k$-clustering of $\mathcal{X}$ as follows. First, $\mathcal{X}$ is partitioned via $T$ using the feature-threshold pairs on the root-to-leaf paths. Then, each point $\mathbf{x} \in \mathcal{X}$ is assigned to one of $k$ clusters according to how $\ell$ labels its leaf. Geometrically, the clusters reside in cells bounded by axis-aligned cuts; the number of cells equals the number of leaves. This results in a $k$-clustering $\widehat{C}^1, \ldots, \widehat{C}^k$ with means $\widehat{\boldsymbol{\mu}}^j = \text{mean}(\widehat{C}^j)$, where the $k$-means cost is

$$\text{cost}(T) = \sum_{j=1}^{k} \sum_{\mathbf{x} \in \widehat{C}^j} \|\mathbf{x} - \widehat{\boldsymbol{\mu}}^j\|_2^2.$$

**Problem Statement.** For a dataset $\mathcal{X} \subseteq \mathbb{R}^d$ and parameters $k, k'$ with $k' \geq k$, the goal is to efficiently construct a binary threshold tree $T$ with $k'$ leaves along with a labeling function $\ell : \text{leaves}(T) \to [k]$ such that $(T, \ell)$ induces a $k$-clustering of $\mathcal{X}$ with as small $k$-means cost as possible.

## 3. Our Algorithms

### 3.1. IMM

We first explain the IMM algorithm that produces a threshold tree with $k$ leaves. It first runs a standard $k$-means algorithm to find $k$ centers. Then, it iteratively finds the best feature-threshold pair to partition the data into two parts. At each step, the number of mistakes is minimized, where a mistake occurs if a data point is separated from its center. Each partition also enforces that at least one center ends up in both children, so that the tree terminates with exactly $k$ leaves. Each leaf contains one center at the end, and the clusters are assigned based this center.

Algorithm 1 takes as input a dataset $\mathcal{X} \subseteq \mathbb{R}^d$. The first step is to obtain a reference set of $k$ centers $\{\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^k\}$, for instance from a standard clustering algorithm. We assign each data point $\mathbf{x}^j$ the label $y^j$ of its closest center. We then call the `build_tree` procedure, which looks for a tree-induced clustering that fits these labels.

The tree is built top-down, using binary splits. Each node $u$ of the tree can be associated with the portion of the input space that passes through that node, a hyper-rectangular region $\text{cell}(u) \subseteq \mathbb{R}^d$. If this cell contains two or more of the centers $\boldsymbol{\mu}^j$, then it needs to be split. We do so by picking the feature $i \in [d]$ and threshold value $\theta \in \mathbb{R}$ such that the resulting split $x_i \leq \theta$ sends at least one center to each side and moreover produces the fewest *mistakes*: that is, separates the fewest points in $\mathcal{X} \cap \text{cell}(u)$ from their corresponding centers in $\{\boldsymbol{\mu}^j : 1 \leq j \leq k\} \cap \text{cell}(u)$. We do not count points whose centers lie outside $\text{cell}(u)$, since they are associated with mistakes in earlier splits.

**Algorithm 1** ITERATIVE MISTAKE MINIMIZATION

**Input** : $\mathbf{x}^1, \ldots, \mathbf{x}^n$ – vectors in $\mathbb{R}^d$
$k$ – number of clusters
**Output** : root of the threshold tree

$\boldsymbol{\mu}^1, \ldots \boldsymbol{\mu}^k \leftarrow$ `k-Means`$(\mathbf{x}^1, \ldots, \mathbf{x}^n, k)$
**foreach** $j \in [1, \ldots, n]$ **do**
  $y^j \leftarrow \arg\min_{1 \leq \ell \leq k} \|\mathbf{x}^j - \boldsymbol{\mu}^\ell\|$
**return** `build_tree`$(\{\mathbf{x}^j\}_{j=1}^n, \{y^j\}_{j=1}^n, \{\boldsymbol{\mu}^j\}_{j=1}^k)$

**build_tree**$(\{\mathbf{x}^j\}_{j=1}^m, \{y^j\}_{j=1}^m, \{\boldsymbol{\mu}^j\}_{j=1}^k)$**:**
  **if** $\{y^j\}_{j=1}^m$ *is homogeneous* **then**
    leaf.$cluster \leftarrow y^1$
    **return** leaf
  **foreach** $i \in [1, \ldots, d]$ **do**
    $\ell_i \leftarrow \min_{1 \leq j \leq m} \mu_i^{y^j}$
    $r_i \leftarrow \max_{1 \leq j \leq m} \mu_i^{y^j}$
  $i, \theta \leftarrow \arg\min_{i, \ell_i \leq \theta < r_i} \sum_{j=1}^m$ `mistake`$(\mathbf{x}^j, \boldsymbol{\mu}^{y^j}, i, \theta)$
  $\mathsf{M} \leftarrow \{j \mid$ `mistake`$(\mathbf{x}^j, \boldsymbol{\mu}^{y^j}, i, \theta) = 1\}_{j=1}^m$
  $\mathsf{L} \leftarrow \{j \mid (x_i^j \leq \theta) \wedge (j \notin \mathsf{M})\}_{j=1}^m$
  $\mathsf{R} \leftarrow \{j \mid (x_i^j > \theta) \wedge (j \notin \mathsf{M})\}_{j=1}^m$
  node.$condition \leftarrow$ "$x_i \leq \theta$"
  node.$lt \leftarrow$ `build_tree`$(\{\mathbf{x}^j\}_{j \in \mathsf{L}}, \{y^j\}_{j \in \mathsf{L}}, \{\boldsymbol{\mu}^j\}_{j=1}^k)$
  node.$rt \leftarrow$ `build_tree`$(\{\mathbf{x}^j\}_{j \in \mathsf{R}}, \{y^j\}_{j \in \mathsf{R}}, \{\boldsymbol{\mu}^j\}_{j=1}^k)$
  **return** node

**mistake**$(\mathbf{x}, \boldsymbol{\mu}, i, \theta)$**:**
  **return** $(x_i \leq \theta) \neq (\mu_i \leq \theta)$ ? 1 : 0

We find the optimal split $(i, \theta)$ by searching over all pairs efficiently using dynamic programming. We then add this node to the tree, and discard the mistakes (the points that got split from their centers) before recursing on the left and right children. We terminate at a leaf node whenever all points have the same label (i.e., the subset of the data is *homogeneous*). Because there were $k$ different labels to begin with, the resulting tree has exactly $k$ leaves.

**Guarantees for IMM.** The IMM algorithm provides an $O(k^2)$ approximation to the optimal $k$-means cost, assuming that a constant-factor approximation algorithm generates the initial $k$ centers. We refer the reader to the full paper for proofs and more details (Dasgupta et al., 2020).

**Theorem 1.** *Suppose that IMM takes centers $\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^k$ and returns a tree $T$ of depth $H$. The $k$-means cost satisfies*

$$\text{cost}(T) \leq (8Hk + 2) \cdot \text{cost}(\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^k)$$

*In particular, IMM achieves worst case approximation factors of $O(k^2)$ using any $O(1)$ approximation to $k$-means.*

We state the theorem in terms of the depth of the tree to highlight that the approximation guarantee may depend on the structure of the input data. We prove the approximation bound by characterizing the excess clustering cost induced by the tree. Any point $\mathbf{x}$ that ends up in a different leaf from its correct center $\boldsymbol{\mu}^j$ incurs some extra cost. To bound this, we consider the internal node $u$ at which $\mathbf{x}$ is separated from $\boldsymbol{\mu}^j$. Node $u$ also contains the center $\boldsymbol{\mu}^i$ that ultimately

ends up in the same leaf as $\mathbf{x}$. The excess cost for $\mathbf{x}$ can then be bounded by $\|\boldsymbol{\mu}^i - \boldsymbol{\mu}^j\|_2^2$ and this is at most the $k$ times the diameter of the cell's bounding box. These terms can be bounded in terms of the cost of the reference clustering.

**Lower Bound.** We also prove that a threshold tree with $k$ leaves cannot, in general, yield better than an $\Omega(\log k)$ approximation to the optimal $k$-means clustering.

**Theorem 2.** *For any $k \geq 2$, there exists a dataset with $k$ clusters such that any threshold tree $T$ with $k$ leaves must have $k$-means cost at least $\text{cost}(T) \geq \Omega(\log k) \cdot \text{cost}(opt)$, where opt is the optimal $k$-medians or means clustering.*

### 3.2. ExkMC

We describe our explainable clustering algorithm, `ExKMC`, that efficiently finds a tree-based $k$-clustering of a dataset. Starting with a base tree (either empty or from an existing algorithm like `IMM`), `ExKMC` expands the tree by replacing a leaf node with two new children. It refines the clustering, while allowing the new children to be mapped to different clusters. A key optimization is to use a new surrogate cost to determine both the best threshold cut and the labeling of the leaves. At the beginning, we run a standard $k$-means algorithm and generate $k$ reference centers. Then, the surrogate cost is the $k$-means cost if the centers were the reference centers. By fixing the centers, instead of changing them at every step, we determine the cluster label for each leaf independently (via the best reference center). For a parameter $k'$, our algorithm terminates when the tree has $k'$ leaves.

**Surrogate cost.** We start with a set of $k$ reference centers $\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^k$, obtained from a standard $k$-means algorithm. This induces a clustering with low $k$-means cost. While it is possible to calculate the actual $k$-means cost as we expand the tree, it is time-consuming to recalculate the distances to a dynamic set of centers. Instead, we fix the reference centers and define the surrogate cost as the sum of squared distances between points and their closest reference center:

**Definition 1** (Surrogate cost)**.** *Given centers $\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^k$ and a threshold tree $T$ that defines the clustering $(\widehat{C}^1, \ldots, \widehat{C}^{k'})$, the* surrogate cost *is defined as*

$$\widetilde{\text{cost}}^{\boldsymbol{\mu}^1,\ldots,\boldsymbol{\mu}^k}(T) = \sum_{j=1}^{k'} \min_{i \in [k]} \sum_{\mathbf{x} \in \widehat{C}^j} \left\|\mathbf{x} - \boldsymbol{\mu}^i\right\|_2^2 .$$

The difference between the new surrogate cost and the $k$-means cost is that the centers are *fixed*.

Algorithm 2 describes the `ExKMC` algorithm, which uses subroutines in Algorithm 3. It takes as input a value $k$, a dataset $\mathcal{X}$, and a number of leaves $k' \geq k$. The first step is to generate $k$ reference centers $\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^k$ from a standard $k$-means implementation and to build a threshold tree $T$ with $k$ leaves (for evaluation, $T$ is the output of the `IMM` algorithm). For simplicity, we refer to these as inputs. `ExKMC` outputs a tree $T'$ and labeling $\ell : \text{leaves}(T') \to [k]$ that assigns leaves to clusters. Notably, the clustering induced by $(T', \ell)$ always refines the one from $T$. At a high level, we compute the best feature-threshold pair to expand the tree one node at a time. For efficiency, we use dynamic programming to scan all thresholds in each coordinate. We expand the tree by splitting the node with the largest improvement to the surrogate cost. Throughout, we store the improvement for each potential split and only update the ones that change. In the end, we create a tree $T'$ with $k'$ labeled leaves, where the labeling $\ell$ maps a leaf to the lowest cost reference center.

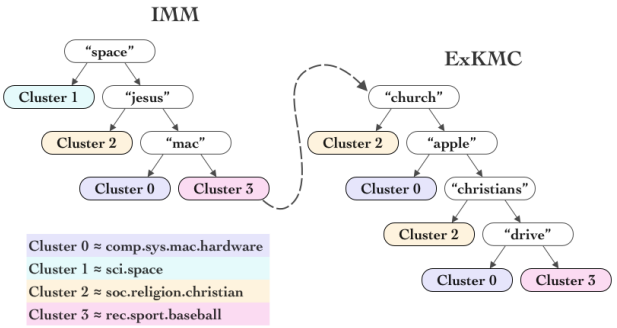



*Figure 2.* **Explainability-accuracy trade-off:** We showcase the effect of adding leaves to the tree while clustering a subset of 20newsgroups. As `ExKMC` expands the `IMM` tree, it refines the clusters and reduces the 4-means cost by using a larger tree.

**Guarantees of `ExKMC`.** We provide some guarantees on the performance of `ExKMC`. We first prove the surrogate cost is non-increasing and it is easy to show that this implies `ExKMC` eventually converges to the reference clustering. See the full paper for details and proofs (Frost et al., 2020).

**Theorem 3.** *The surrogate cost, $\widetilde{\text{cost}}$, is non-increasing throughout the execution of* `ExKMC`.

We also verify that `ExKMC` has a worst-case approximation ratio of $O(k^2)$ compared to the optimal $k$-means cost when using `IMM` to build the base tree. Finally, we provide a separation between `IMM` and `ExKMC` for the lower bound dataset from Theorem 2, where we prove that `ExKMC` leads to an optimal tree-based clustering with $O(k \log k)$ leaves.

**Qualitative analysis.** Figure 2 depicts two example trees with four and eight leaves, respectively. on a subset of four clusters from the 20newsgroups dataset. The `IMM` base tree uses three features (words) to define four clusters. Then, `ExKMC` expands one of the leaves into a larger subtree, using seven total words to construct more nuanced clusters that better correlate with the newsgroup topics.

**Algorithm 2** ExKMC: Expanding Explainable $k$-Means Clustering

**Input** : $\mathcal{X}$ – Set of vectors in $\mathbb{R}^d$
$\mathcal{M}$ – Set of $k$ reference centers
$T$ – Base tree
$k'$ – Number of leaves
**Output** : Labeled tree with $k'$ leaves

splits $\leftarrow$ `dict()`
gains $\leftarrow$ `dict()`
**foreach** *leaf* $\in T.leaves$ **do**
  `add_gain`(leaf, $\mathcal{X}$, $\mathcal{M}$, splits, gains)
**while** $|T.leaves| < k'$ **do**
  leaf $\leftarrow \arg\max_{\text{leaf}}$ gains[leaf]
  $i, \theta \leftarrow$ splits[leaf]
  $\boldsymbol{\mu}^L, \boldsymbol{\mu}^R \leftarrow$ `find_labels`$(\mathcal{X}, \mathcal{M}, i, \theta)$
  leaf.$condition \leftarrow$ "$x_i \leq \theta$"
  leaf.$l \leftarrow$ `new Leaf`$(label = \boldsymbol{\mu}^L)$
  leaf.$r \leftarrow$ `new Leaf`$(label = \boldsymbol{\mu}^R)$
  `add_gain`(leaf.$l$, $\mathcal{X}$, $\mathcal{M}$, splits, gains)
  `add_gain`(leaf.$r$, $\mathcal{X}$, $\mathcal{M}$, splits, gains)
  `delete`(splits[leaf], gains[leaf])
**return** $T$

**Algorithm 3** Subroutines

**`add_gain`**(leaf, $\mathcal{X}$, $\mathcal{M}$, splits, gains):
  $\mathcal{X}_l \leftarrow \{\mathbf{x} \in \mathcal{X} \mid \mathbf{x}$ path ends in leaf$\}$
  $i, \theta \leftarrow \arg\min_{i,\theta}$ `split_cost`$(\mathcal{X}_l, \mathcal{M}, i, \theta)$
  best_cost $\leftarrow$ `split_cost`$(\mathcal{X}_l, \mathcal{M}, i, \theta)$
  splits[leaf] $\leftarrow (i, \theta)$
  gains[leaf] $\leftarrow \widetilde{\text{cost}}_{\mathcal{M}}(\mathcal{X}_l) -$ best_cost

**`split_cost`**$(\mathcal{X}, \mathcal{M}, i, \theta)$:
  $\mathcal{X}_L \leftarrow \{\mathbf{x} \in \mathcal{X} \mid x_i \leq \theta\}$
  $\mathcal{X}_R \leftarrow \{\mathbf{x} \in \mathcal{X} \mid x_i > \theta\}$
  **return** $\widetilde{\text{cost}}_{\mathcal{M}}(\mathcal{X}_L) + \widetilde{\text{cost}}_{\mathcal{M}}(\mathcal{X}_R)$

**`find_labels`**$(\mathcal{X}, \mathcal{M}, i, \theta)$:
  $\boldsymbol{\mu}^L \leftarrow \arg\min_{\boldsymbol{\mu} \in \mathcal{M}} \sum_{\mathbf{x} \in \mathcal{X}: x_i \leq \theta} \|\mathbf{x} - \boldsymbol{\mu}\|_2^2$
  $\boldsymbol{\mu}^R \leftarrow \arg\min_{\boldsymbol{\mu} \in \mathcal{M}} \sum_{\mathbf{x} \in \mathcal{X}: x_i > \theta} \|\mathbf{x} - \boldsymbol{\mu}\|_2^2$
  **return** $\boldsymbol{\mu}^L, \boldsymbol{\mu}^R$

# 4. Empirical Evaluation

**Algorithms.** We compare the following clustering methods:

- **Reference Clustering.** We use `sklearn KMeans`, 10 random initializations, 300 iterations.
- **CART.** Standard decision tree from `sklearn` that minimizes `gini` impurity. Points in the dataset are assigned labels using the reference clustering.
- **KDTree.** Split highest variance feature at median. Size determined by `leaf_size` parameter. Labels minimize $\widetilde{\text{cost}}$ w.r.t. centers of the reference clustering.
- **CLTree.** Explainable clustering method. Public implementation (Christodoulou; Liu et al., 2005).
- **CUBT.** Explainable clustering method. Public implementation (Ghattas; Fraiman et al., 2013).
- **ExKMC.** Algorithm 2 with empty base tree; then, minimizes $\widetilde{\text{cost}}$ at each split w.r.t. reference centers.
- **ExKMC (base: IMM).** Algorithm 2 with `IMM` base tree starting with $k$ leaves; then, minimizes $\widetilde{\text{cost}}$ at each split w.r.t. reference centers.

**Set-up.** We use 10 standard real datasets. The number of clusters $k$ and the number of leaves $k'$ are inputs. We start with $k$ equal to number of labels for classification datasets. We plot the cost ratio compared to the reference clustering (best $= 1.0$). For the baselines, we do hyperparameter tuning and choose the lowest cost clustering at each $k'$. `CUBT` and `CLTree` could only be feasibly executed on six small real datasets (we cap the running time at one hour).

## 4.1. Experimental Results and Discussion

On all of the datasets, our method `ExKMC` performs well. It often has the lowest cost throughout, except for CIFAR-10, where it requires around $3.5 \cdot k$ leaves to be competitive. On the Avila dataset, `ExKMC` significantly outperforms CART.

When the number of leaves is exactly $k$, we obtain the performance of the `IMM` algorithm, where we see that its cost is quite low (much better than the theoretical analysis).

`CLTree` performs the worst in most cases, and the cost is often much larger than the other methods. Turning to `CUBT`, we see that on most datasets it is competitive (but often not the best). However, on Digits and Mice Protein, `CUBT` fails to converge to a good clustering. We see that `CART` performs well on many of the datasets, as expected. On Avila and 20newsgroups, `CART` has a fairly high cost. For the small datasets, KDTree performs competitively, but on large datasets, the cost remains high. We observe that the `CLTree` cost varies as a function of the number of leaves. We separately perform a hyperparameter search for each instance, and surprisingly, the cost can sometimes increase.

**Trade-off.** The main objective of our new algorithm is to provide a flexible trade-off between explainability and accuracy. Compared to the `IMM` algorithm, we see that using `ExKMC` to expand the threshold tree consistently leads to a lower cost clustering. We also see that our surrogate cost improves the running time without sacrificing effectiveness.

**Low cost.** `ExKMC` often achieves a lower $k$-means cost for a given number of leaves compared to all four baselines `CUBT` (Fraiman et al., 2013), `CLTree` (Liu et al., 2005), KDTree (Bentley, 1975), and `CART` (Loh, 2011).

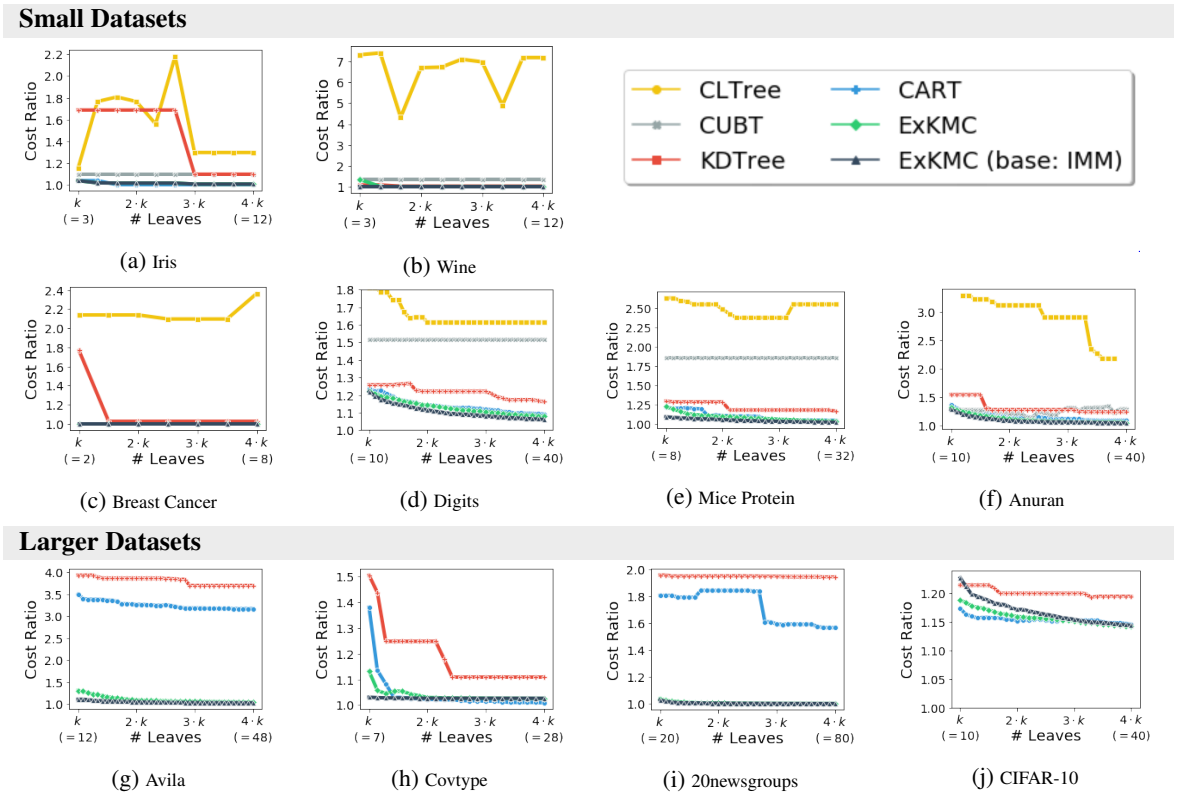



*Figure 3.* Ratio of the tree-based clustering cost to the near-optimal $k$-means clustering ($y$-axis) varying the number of leaves ($x$-axis). **Lower is better, best** $= 1.0$**.** Our algorithm (black line) consistently performs well.

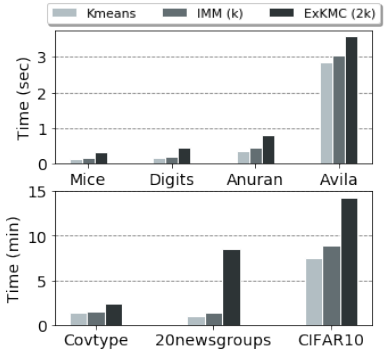



*Figure 4.* Runtime.

**Convergence.** `IMM` produces a fairly low $k$-means cost with $k$ leaves. Expanding the tree with `ExKMC` to $4k$ leaves often results in nearly the same cost as the reference clustering. CIFAR-10 is an outlier, where none of the methods converge when using pixels as features. In practice, a tree with $4k$ leaves only slightly increases the explanation complexity compared to a tree with $k$ leaves (and $k$ leaves are necessary). `ExKMC` successfully achieves a good tree-based clustering with better interpretability than standard methods.

**Running time.** Figure 4 shows the runtime (single process, i7 CPU @ 2.80GHz, 16GB RAM). Both `IMM` and `ExKMC` first run `KMeans` (from `sklearn`, 10 initializations, 300 iterations), and we report cumulative times. The explainable algorithms construct trees in under 15 minutes. On six datasets, they incur $0.25\times$ to $1.5\times$ overhead compared to standard `KMeans`. The 20newsgroups dataset has the largest overhead because `sklearn` optimizes for sparse vectors while `IMM` and `ExKMC` currently do not.

## 5. Conclusion

We present two new algorithms, `IMM` and `ExKMC`, for explainable $k$-means clustering. Theoretically, `IMM` is the first such algorithm with a provable guarantee that only depends on the number of clusters. To overcome lower bounds on trees with $k$ leaves, we develop `ExKMC`, which generates a threshold tree with a specified number of leaves. Empirically, we find that the `IMM` worst-case analysis is pessimistic because even with $k$ leaves it achieves clustering cost within 5–30% versus standard $k$-means algorithms. Moreover, `ExKMC` often has lower $k$-means cost than four baselines. We find that threshold trees with $4k$ leaves suffice to get within 1–2% of the cost of a typical $k$-means algorithm. Overall, we verify that it is possible to find an explainable clustering with high accuracy, while using only $O(k)$ leaves for $k$-means clustering. `ExKMC` could replace standard $k$-means implementations in data science pipelines.

## References


Aggarwal, A., Deshpande, A., and Kannan, R. Adaptive sampling for k-means clustering. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 15–28. Springer, 2009.

Aloise, D., Deshpande, A., Hansen, P., and Popat, P. NP-hardness of Euclidean sum-of-squares clustering. *Machine learning*, 75 (2):245–248, 2009.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

Arthur, D. and Vassilvitskii, S. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete algorithms*, pp. 1027–1035. Society for Industrial and Applied Mathematics, 2007.

Awasthi, P., Charikar, M., Krishnaswamy, R., and Sinop, A. K. The hardness of approximation of Euclidean k-means. In *31st International Symposium on Computational Geometry, SoCG 2015*, pp. 754–767. Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, 2015.

Basak, J. and Krishnapuram, R. Interpretable hierarchical clustering by constructing an unsupervised decision tree. *IEEE transactions on knowledge and data engineering*, 17(1):121–132, 2005.

Becchetti, L., Bury, M., Cohen-Addad, V., Grandoni, F., and Schwiegelshohn, C. Oblivious dimension reduction for k-means: beyond subspaces and the Johnson-Lindenstrauss lemma. In *STOC*, 2019.

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

Bertsimas, D., Orfanoudaki, A., and Wiberg, H. Interpretable clustering via optimal trees. *arXiv preprint arXiv:1812.00539*, 2018.

Boutsidis, C., Drineas, P., and Mahoney, M. W. Unsupervised feature selection for the k-means clustering problem. In *NIPS*, pp. 153–161, 2009.

Boutsidis, C., Zouzias, A., Mahoney, M. W., and Drineas, P. Randomized dimensionality reduction for $k$-means clustering. *IEEE Transactions on Information Theory*, 61(2):1045–1062, 2014.

Chang, J.-W. and Jin, D.-S. A new cell-based clustering method for large, high-dimensional data in data mining applications. In *Proceedings of the 2002 ACM symposium on Applied computing*, pp. 503–507, 2002.

Chen, J. *Interpretable Clustering Methods*. PhD thesis, Northeastern University, 2018.

Chen, J., Chang, Y., Hobbs, B., Castaldi, P., Cho, M., Silverman, E., and Dy, J. Interpretable clustering via discriminative rectangle mixture model. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 823–828. IEEE, 2016.

Christodoulou, D. Python-package for clustering via decision tree construction. https://github.com/dimitrs/CLTree.

Cohen, M. B., Elder, S., Musco, C., Musco, C., and Persu, M. Dimensionality reduction for k-means clustering and low rank approximation. In *STOC*, 2015.

Dasgupta, S. The hardness of k-means clustering. In *Technical Report*. University of California, San Diego (Technical Report), 2008.

Dasgupta, S., Frost, N., Moshkovitz, M., and Rashtchian, C. Explainable $k$-means and $k$-medians clustering. *(To appear) International Conference on Machine Learning*, 2020.

De Raedt, L. and Blockeel, H. Using logical decision trees for clustering. In *International Conference on Inductive Logic Programming*, pp. 133–140. Springer, 1997.

Deutch, D. and Frost, N. Constraints-based explanations of classifications. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 530–541. IEEE, 2019.

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

Frost, N., Moshkovitz, M., and Rashtchian, C. Exkmc: Expanding explainable $k$-means clustering. *arXiv preprint arXiv:2006.02399*, 2020.

Geurts, P. and Louppe, G. Learning to rank with extremely randomized trees. In *JMLR: workshop and conference proceedings*, volume 14, pp. 49–61, 2011.

Geurts, P., Touleimat, N., Dutreix, M., and d'Alché Buc, F. Inferring biological networks with output kernel trees. *BMC Bioinformatics*, 8(2):S4, 2007.

Ghattas, B. R-package for interpretable clustering using unsupervised binary trees. http://www.i2m.univ-amu.fr/perso/badih.ghattas/CUBT.html.

Ghattas, B., Michel, P., and Boyer, L. Clustering nominal data using unsupervised binary decision trees: Comparisons with the state of the art methods. *Pattern Recognition*, 67:177–185, 2017.

Jernite, Y., Choromanska, A., and Sontag, D. Simultaneous learning of trees and representations for extreme classification and density estimation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1665–1674. JMLR. org, 2017.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. A local search approximation algorithm for k-means clustering. In *Proceedings of the Eighteenth Annual Symposium on Computational Geometry*, pp. 10–18, 2002.

Kauffmann, J., Esders, M., Montavon, G., Samek, W., and Müller, K.-R. From clustering to cluster explanations via neural networks. *arXiv preprint arXiv:1906.07633*, 2019.

Lipton, Z. C. The mythos of model interpretability. *Queue*, 16(3): 31–57, 2018.

Liu, B., Xia, Y., and Yu, P. S. Clustering via decision tree construction. In *Foundations and Advances in Data Mining*, pp. 97–124. Springer, 2005.

Loh, W.-Y. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1): 14–23, 2011.

Louppe, G., Wehenkel, L., Sutera, A., and Geurts, P. Understanding variable importances in forests of randomized trees. In *Advances in neural information processing systems*, pp. 431–439, 2013.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pp. 4765–4774, 2017.

Makarychev, K., Makarychev, Y., and Razenshteyn, I. Performance of Johnson-Lindenstrauss transform for k-means and k-medians clustering. In *STOC*, 2019.

Molnar, C. *Interpretable Machine Learning*. Lulu. com, 2019. https://christophm.github.io/interpretable-ml-book/.

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019.

Ostrovsky, R., Rabani, Y., Schulman, L. J., and Swamy, C. The effectiveness of Lloyd-type methods for the k-means problem. *Journal of the ACM*, 59(6):1–22, 2013.

Pelleg, D. and Moore, A. W. Mixtures of rectangles: Interpretable soft clustering. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pp. 401–408, 2001.

Pliakos, K., Geurts, P., and Vens, C. Global multi-output decision trees for interaction prediction. *Machine Learning*, 107(8-10): 1257–1281, 2018.

Ribeiro, M. T., Singh, S., and Guestrin, C. 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*, pp. 1135–1144. ACM, 2016.

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

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

Yasami, Y. and Mozaffari, S. P. A novel unsupervised classification approach for network anomaly detection by k-means clustering and id3 decision tree learning methods. *The Journal of Supercomputing*, 53(1):231–245, 2010.