Explainable k-Means and k-Medians Clustering

Sanjoy Dasgupta, Nave Frost, Michal Moshkovitz, Cyrus Rashtchian

Ilana Sivan and Agathe Benichou

Agenda

Background

- k-Means and k-Medians
- Explainability
- Explainable k-Means
- Motivation
- Challenges

IMM Algorithm

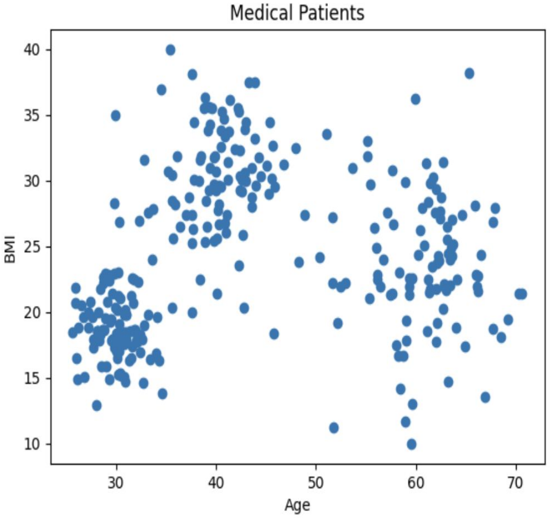
- Goal of the paper
- Procedure

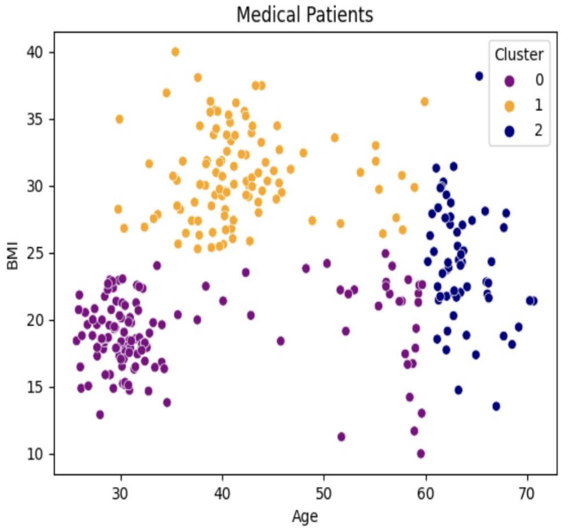
Implications

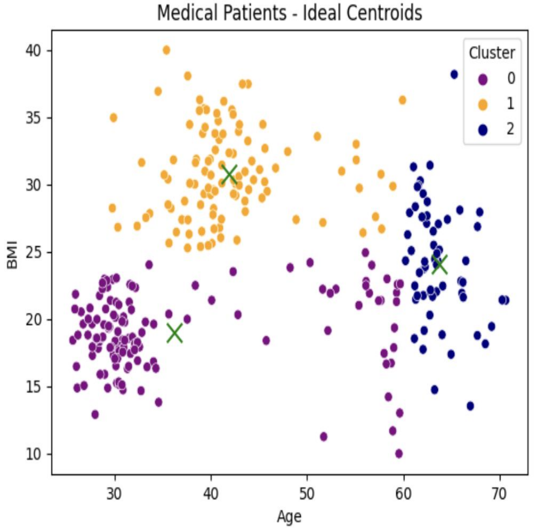
- Key Findings
- Comparison and Flaws
- Takeaways

Concept: Clustering

- Clustering algorithms group together similar data points
- Examples include density scanning, distance from the centroid, or hierarchical structure







Concept: k-Means and k-Medians

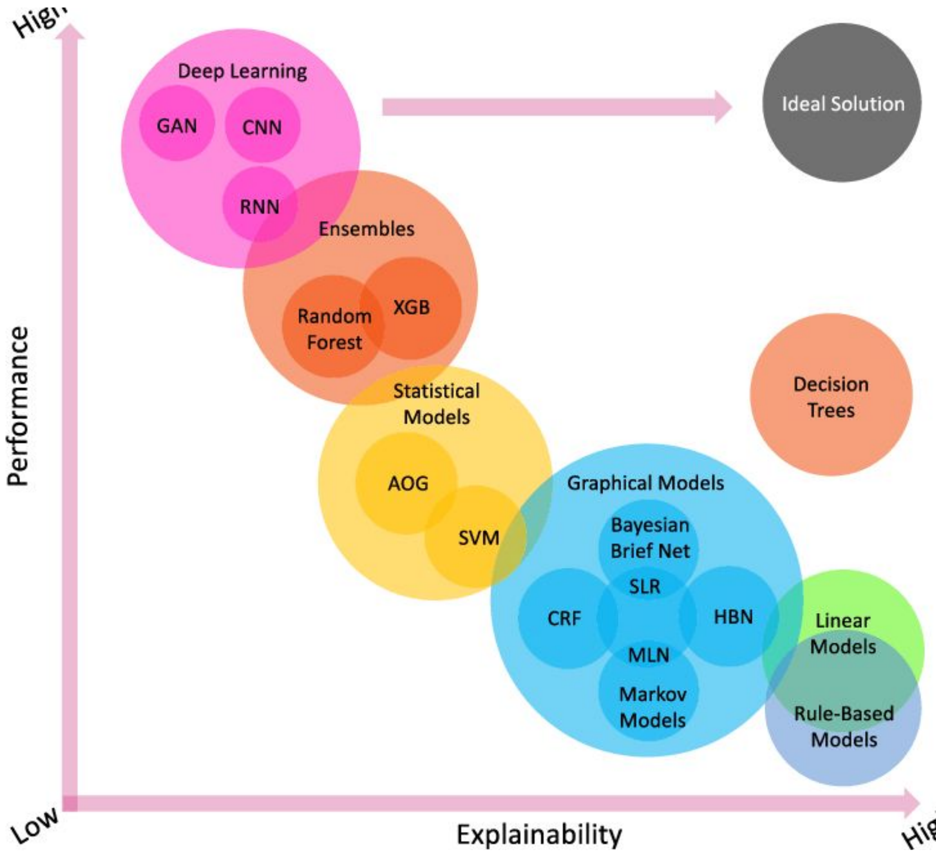
- Iterative algorithm that aims to divide N points into k distinct clusters
  - Goal: minimize of the sum of squared distances between points and their assigned centroids
  - Cost Function:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{m} a_{ij} ||x_i - \mu_j||_2^2$$

  - Procedure: Initialize, assign, update, repeat
- K-Medians is a variant that calculates median for each cluster to determine its centroid (median is more robust to outliers)

Introduction to Explainability

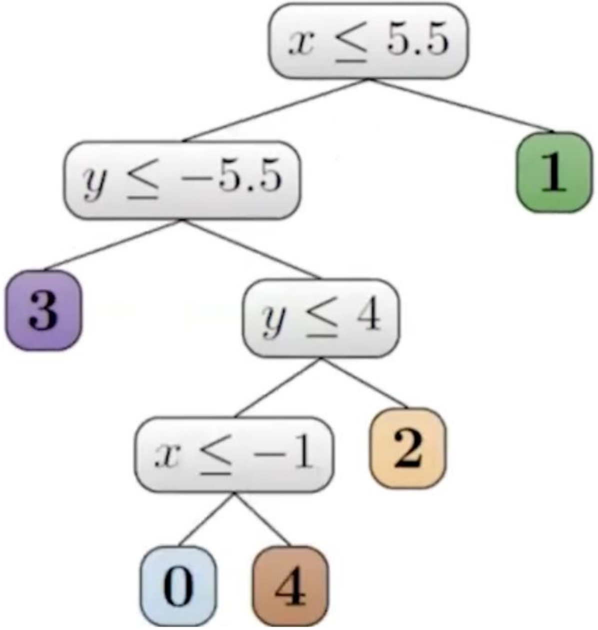
- As models become more complex, their decision become less transparent to human operators
- **Explainability**: the understandability of a models decision making process
- LIME is a method for explaining the predictions
  - Idea: approximate the decision boundary of a complex model locally with a simple model
  - Cons: doesn't provide direct insight into the dataset and the explanations depend on the model
- Goal: configure more principled approaches to interpretable methods



Yang, Guang & Ye, Qinghao & Xia, Jun. (2021). Unbox the black-box for the medical explainable AI via multi-modal and multi-centre data fusion: A mini-review, two showcases and beyond. Information Fusion. 77. 10.1016/j.inffus.2021.07.016.

Explainable k-Means

- Explainable k-Means is an approach to understand cluster groups
- How can we integrate explainability into the realm of traditional k-Means clustering?
  - Threshold trees are unsupervised variants of decision trees
- Fun fact: It is **NP-hard** to find the optimal k-means clustering (Aloise et al., 2009; Dasgupta, 2008) or even a very close approximation (Awasthi et al., 2015), so we focus on approximating to the best of our abilities.



Motivation: Explainable k-Means

- In high dimensional data, traditional clustering methods can lead to complex clusters
  - Harder to unweave the feature relations of the clustering
  - Impossible to represent this as a small decision tree
- Aim to provide simple explanations that represent the clusters
- Explainability matters in real world applications
  - When a doctor is told by an AI model that a patient needs surgery, they want to understand it for themselves
- How to build explainable clustering?
- Is this clustering as good as traditional methods?

Challenges: Explainable k-Means

- Complex feature relationship: may be a result of a combination of features
- Dimensionality reduction or feature selection does not improve interpretability
  - An unexplainable clustering algorithm is often invoked on the modified dataset
- **Tradeoff**: Achieving interpretability comes at the cost of increased computational complexity or lower clustering accuracy
  - Focusing on minimizing cost or improving clustering accuracy can lead to models that are mode difficult to interpret
- Balancing act between interpretability and cost in order to build effective and understandable models

IMM Algorithm

- Specifically designed for the k > 2 case
- A **mistake** occurs when a data point in one split is closer to a center in the other split, after the cut at that node
- Introduces an **approximation algorithm** that is independent of the number of dimensions and points
- Recurse over all cuts with **dynamic programming**

Algorithm 1 ITERATIVE MISTAKE MINIMIZATION

Input : $\mathbf{x}^1, \dots, \mathbf{x}^n$ – vectors in $\mathbb{R}^d$
k – number of clusters

Output : root of the threshold tree

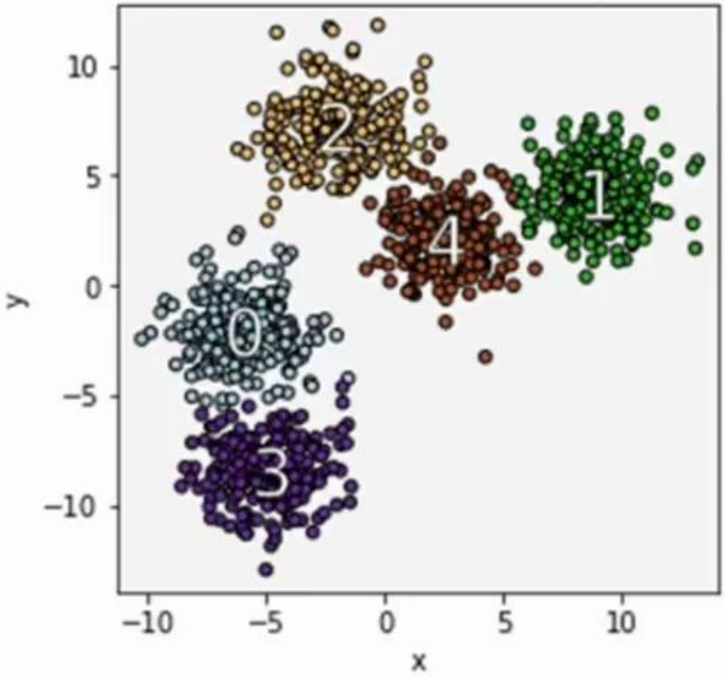
$\boldsymbol{\mu}^1, \dots \boldsymbol{\mu}^k \leftarrow$ `k-Means`$(\mathbf{x}^1, \dots, \mathbf{x}^n, k)$
foreach $j \in [1, \dots, n]$ **do**
  $y^j \leftarrow \arg\min_{1 \le \ell \le k} \|\mathbf{x}^j - \boldsymbol{\mu}^\ell\|$
end
return `build_tree`$(\{\mathbf{x}^j\}_{j=1}^n, \{y^j\}_{j=1}^n,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

`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
  end
  foreach $i \in [1, \dots, d]$ **do**
    $\ell_i \leftarrow \min_{1 \le j \le m} \mu_i^{y^j}$
    $r_i \leftarrow \max_{1 \le j \le m} \mu_i^{y^j}$
  end
  $i, \theta \leftarrow \arg\min_{i, \ell_i \le \theta < r_i} \sum_{j=1}^m$ `mistake`$(\mathbf{x}^j, \boldsymbol{\mu}^{y^j}, i, \theta)$
  $\mathsf{M} \leftarrow \{j \mid$ `mistake`$(\mathbf{x}^j, \boldsymbol{\mu}^{y^j}, i, \theta) = 1\}_{j=1}^m$
  $\mathsf{L} \leftarrow \{j \mid (x_i^j \le \theta) \wedge (j \notin \mathsf{M})\}_{j=1}^m$
  $\mathsf{R} \leftarrow \{j \mid (x_i^j > \theta) \wedge (j \notin \mathsf{M})\}_{j=1}^m$
  node.*condition* $\leftarrow$ "$x_i \le \theta$"
  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 $(x_i \le \theta) \ne (\mu_i \le \theta)$? 1 : 0

IMM Algorithm: Procedure

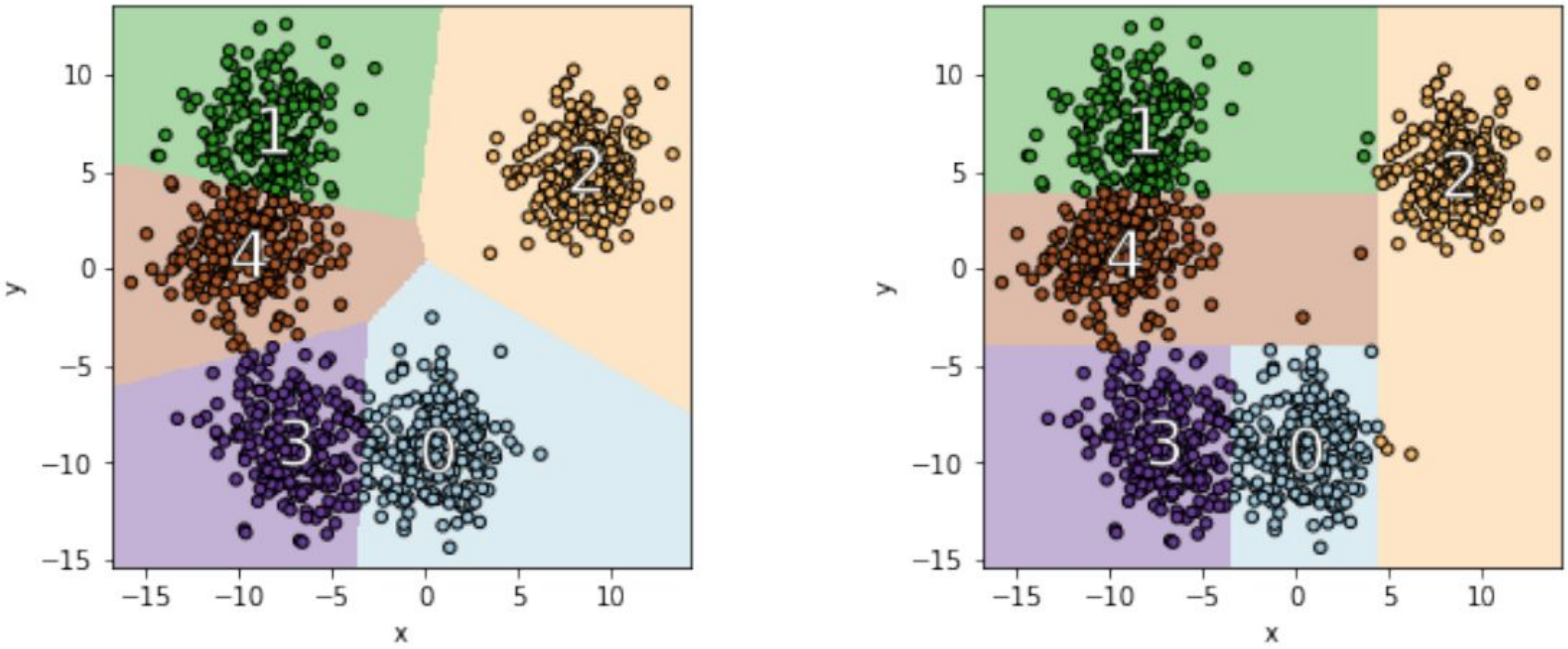
- Run a clustering algorithm of your choice
- Label each sample with its cluster
- Build a top down threshold tree from the root to the leaves
  - At each step, find the split with the minimal number of mistakes
  - Dynamic programming is used to find the optimal cuts at each level
- Result is k leaves and k cluster classes
  - Each internal node contains a single feature and a threshold
- Compare the cost of new model to the original model:
  - Run over all the clusters and compute the cost function



Mistakes

Current split: -

Total: -



(a) Optimal 5-means clusters

(b) Tree based 5-means clusters

IMM Algorithm: Key Findings

- Efficient run time for general k: $O(k * d * n * logn)$
  - k: number of clusters
  - d: dimensionality of the dataset,
  - n: total number of points
- Provable guarantees: it is an O(k^2) approximation
  - Doesn't depend on dimensionality or the number of points
  - Nearly optimal clustering
- Provides theoretical guarantees for k=2: there exists a threshold cut with low cost, compare to the optimal clustering, and shows it has locality (one feature, one threshold)
- K-means:
  - For k=2, the price of explainability is between 3 and 4
  - For k > 2, it is between log k and k^2
- K-medians:
  - For k=2, the price of explainability is exactly 2
  - For k > 2, it is between log k and k
- Holds for any dataset

	k-medians		k-means	
	$k = 2$	$k > 2$	$k = 2$	$k > 2$
Lower	$2 - \frac{1}{d}$	$\Omega(\log k)$	$3\left(1 - \frac{1}{d}\right)^2$	$\Omega(\log k)$
Upper	2	$O(k)$	4	$O(k^2)$

Comparisons

- IMM aims to make clusters explainable
  - Employs dynamic programming
  - Introduces concept of mistakes
  - Cost tradeoff for interpretability
- Performs really well compared to other techniques
  - ID3 is based on information gain
- IMM is comparable to k-means

Flaws

- Approximation bounds depend on the height of tree H
  - Higher depth may lead to a higher approximation cost, especially for k-means clustering, where cost can go up to O(Hk)
- Datasets with complex or overlapping distributions, mistakes can be very high
- Requires a predetermined k

Takeaways

- IMM displays the balance between providing an interpretable model and retaining a reasonable degree of clustering accuracy
  - Handles tradeoff between explainabilty and optimality in clustering costs
- Applied to both k-means and k-medians
- Uses a combination of dynamic programming and exhaustive search
- IMM is an exciting example of future research, with studies on using more features having been published recently

Thank you!

Questions?