The story of explainable clustering

Adam Polak

ML in PL, Warsaw, October 28th, 2023

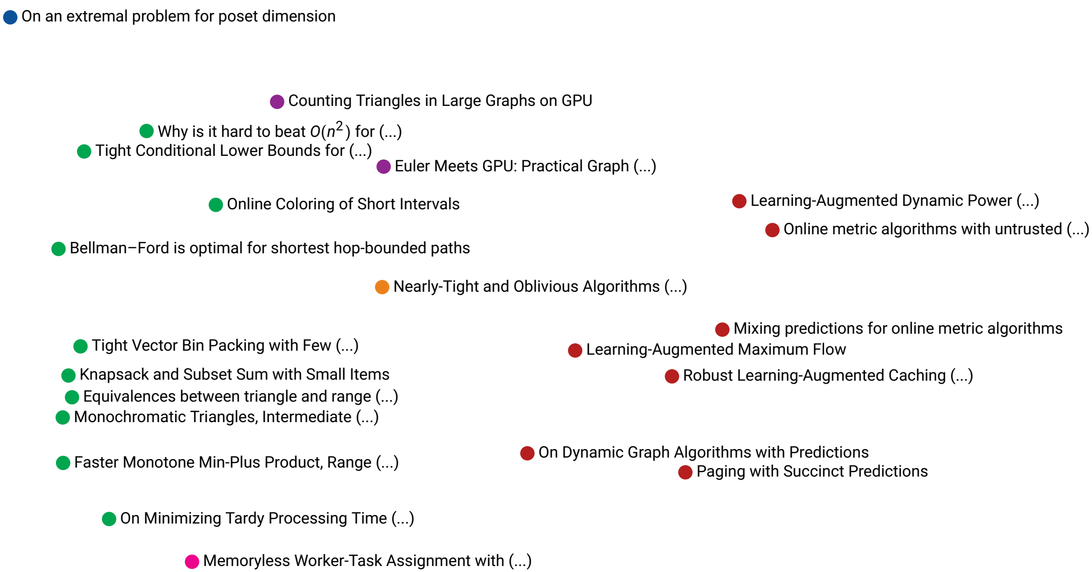
```python
url = ("https://export.arxiv.org/api/query" +
  "?search_query=au:%22Adam%20Polak%22&max_results=50")
feed = feedparser.parse(urllib.request.urlopen(url).read())
data = [(entry.title, entry.summary) for entry in feed.entries]
```

```python
url = ("https://export.arxiv.org/api/query" +
  "?search_query=au:%22Adam%20Polak%22&max_results=50")
feed = feedparser.parse(urllib.request.urlopen(url).read())
data = [(entry.title, entry.summary) for entry in feed.entries]
embed = hub.load(
  "https://tfhub.dev/google/universal-sentence-encoder-large/5")
embeddings = embed([abstract for _, abstract in data]).numpy()
```

```python
url = ("https://export.arxiv.org/api/query" +
  "?search_query=au:%22Adam%20Polak%22&max_results=50")
feed = feedparser.parse(urllib.request.urlopen(url).read())
data = [(entry.title, entry.summary) for entry in feed.entries]
embed = hub.load(
  "https://tfhub.dev/google/universal-sentence-encoder-large/5")
embeddings = embed([abstract for _, abstract in data]).numpy()
clusters = sklearn.cluster.KMeans(n_clusters=6).fit_predict(embeddings)
```

```python
url = ("https://export.arxiv.org/api/query" +
  "?search_query=au:%22Adam%20Polak%22&max_results=50")
feed = feedparser.parse(urllib.request.urlopen(url).read())
data = [(entry.title, entry.summary) for entry in feed.entries]
embed = hub.load(
  "https://tfhub.dev/google/universal-sentence-encoder-large/5")
embeddings = embed([abstract for _, abstract in data]).numpy()
clusters = sklearn.cluster.KMeans(n_clusters=6).fit_predict(embeddings)
pca = sklearn.decomposition.PCA(n_components=2)
coordinates = pca.fit_transform(embeddings)
```

```python
url = ("https://export.arxiv.org/api/query" +
  "?search_query=au:%22Adam%20Polak%22&max_results=50")
feed = feedparser.parse(urllib.request.urlopen(url).read())
data = [(entry.title, entry.summary) for entry in feed.entries]
embed = hub.load(
  "https://tfhub.dev/google/universal-sentence-encoder-large/5")
embeddings = embed([abstract for _, abstract in data]).numpy()
clusters = sklearn.cluster.KMeans(n_clusters=6).fit_predict(embeddings)
pca = sklearn.decomposition.PCA(n_components=2)
coordinates = pca.fit_transform(embeddings)
for (x, y), (title, _) in zip(coordinates, data):
  plt.text(x, y, title)
for i in range(max(clusters) + 1):
  plt.scatter(coordinates[clusters==i, 0], coordinates[clusters==i, 1])
```


On an extremal problem for poset dimension
Counting Triangles in Large Graphs on GPU
Why is it hard to beat $O(n^2)$ for (...)
Tight Conditional Lower Bounds for (...)
Euler Meets GPU: Practical Graph (...)
Online Coloring of Short Intervals
Learning-Augmented Dynamic Power (...)
Online metric algorithms with untrusted (...)
Bellman–Ford is optimal for shortest hop-bounded paths
Nearly-Tight and Oblivious Algorithms (...)
Mixing predictions for online metric algorithms
Tight Vector Bin Packing with Few (...)
Learning-Augmented Maximum Flow
Knapsack and Subset Sum with Small Items
Robust Learning-Augmented Caching (...)
Equivalences between triangle and range (...)
Monochromatic Triangles, Intermediate (...)
On Dynamic Graph Algorithms with Predictions
Faster Monotone Min-Plus Product, Range (...)
Paging with Succinct Predictions
On Minimizing Tardy Processing Time (...)
Memoryless Worker-Task Assignment with (...)

On an extremal problem for poset dimension

Fine-grained complexity

Counting Triangles in Large Graphs on GPU

Why is it hard to beat $O(n^2)$ for (...)

Tight Conditional Lower Bounds for (...)

Euler Meets GPU: Practical Graph (...)

Online Coloring of Short Intervals

Learning-Augmented Dynamic Power (...)

Online metric algorithms with untrusted (...)

Bellman–Ford is optimal for shortest hop-bounded paths

Nearly-Tight and Oblivious Algorithms (...)

Mixing predictions for online metric algorithms

Tight Vector Bin Packing with Few (...)

Learning-Augmented Maximum Flow

Knapsack and Subset Sum with Small Items

Robust Learning-Augmented Caching (...)

Equivalences between triangle and range (...)

Monochromatic Triangles, Intermediate (...)

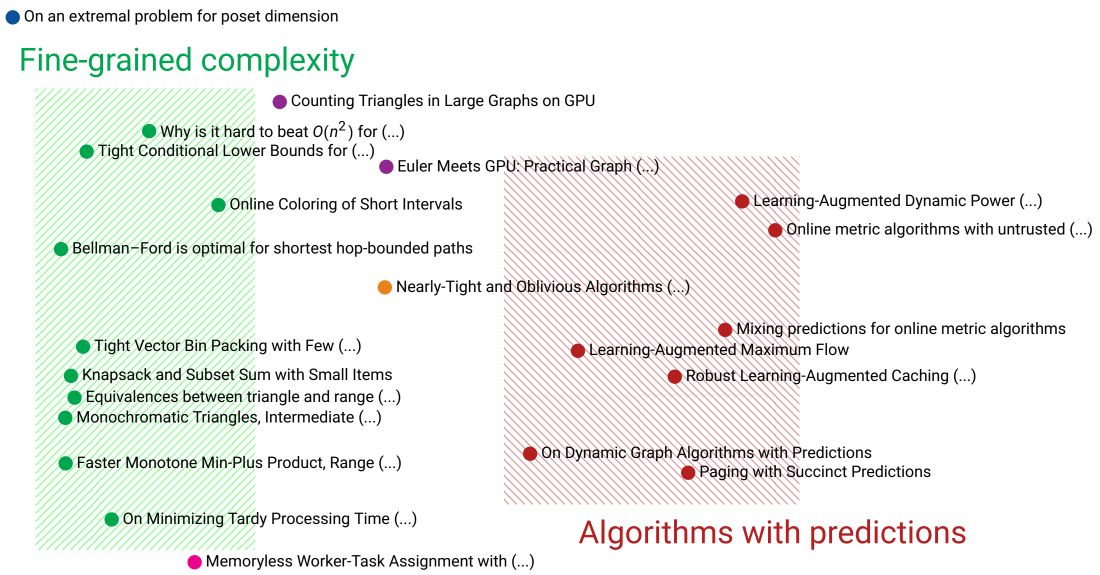
On Dynamic Graph Algorithms with Predictions

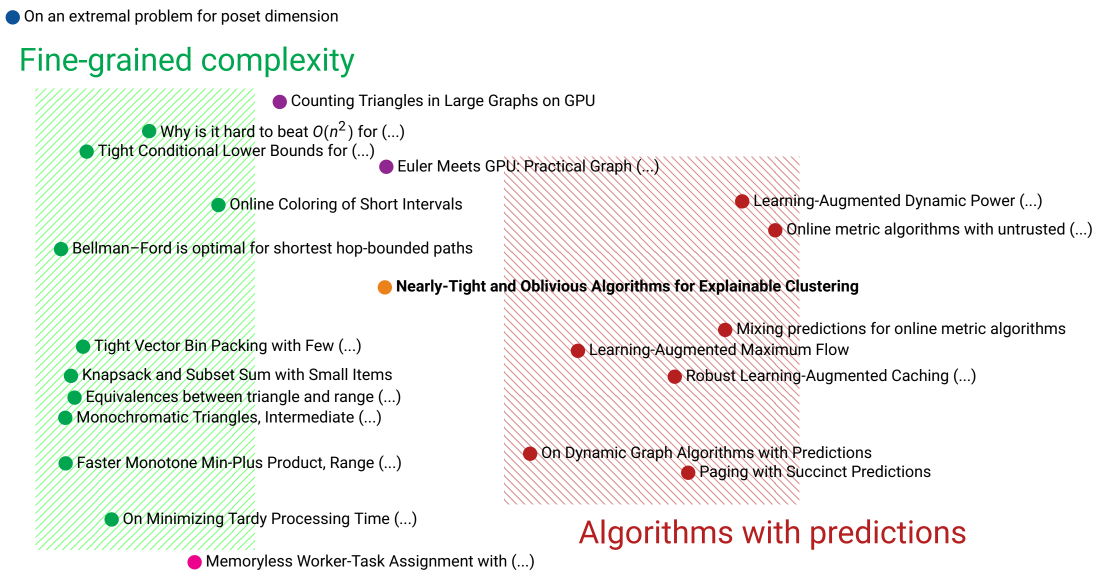
Faster Monotone Min-Plus Product, Range (...)

Paging with Succinct Predictions

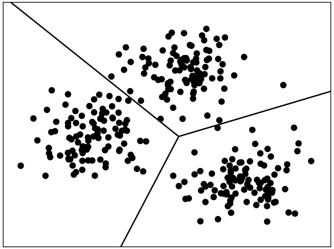
On Minimizing Tardy Processing Time (...)

Memoryless Worker-Task Assignment with (...)

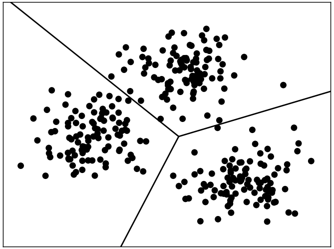

On an extremal problem for poset dimension
Fine-grained complexity
Counting Triangles in Large Graphs on GPU
Why is it hard to beat $O(n^2)$ for (...)
Tight Conditional Lower Bounds for (...)
Euler Meets GPU: Practical Graph (...)
Learning-Augmented Dynamic Power (...)
Online Coloring of Short Intervals
Online metric algorithms with untrusted (...)
Bellman–Ford is optimal for shortest hop-bounded paths
Nearly-Tight and Oblivious Algorithms (...)
Mixing predictions for online metric algorithms
Tight Vector Bin Packing with Few (...)
Learning-Augmented Maximum Flow
Knapsack and Subset Sum with Small Items
Robust Learning-Augmented Caching (...)
Equivalences between triangle and range (...)
Monochromatic Triangles, Intermediate (...)
On Dynamic Graph Algorithms with Predictions
Faster Monotone Min-Plus Product, Range (...)
Paging with Succinct Predictions
On Minimizing Tardy Processing Time (...)
Algorithms with predictions
Memoryless Worker-Task Assignment with (...)


On an extremal problem for poset dimension
Fine-grained complexity
Counting Triangles in Large Graphs on GPU
Why is it hard to beat $O(n^2)$ for (...)
Tight Conditional Lower Bounds for (...)
Euler Meets GPU: Practical Graph (...)
Online Coloring of Short Intervals
Learning-Augmented Dynamic Power (...)
Online metric algorithms with untrusted (...)
Bellman–Ford is optimal for shortest hop-bounded paths
Nearly-Tight and Oblivious Algorithms for Explainable Clustering
Mixing predictions for online metric algorithms
Tight Vector Bin Packing with Few (...)
Learning-Augmented Maximum Flow
Knapsack and Subset Sum with Small Items
Robust Learning-Augmented Caching (...)
Equivalences between triangle and range (...)
Monochromatic Triangles, Intermediate (...)
On Dynamic Graph Algorithms with Predictions
Faster Monotone Min-Plus Product, Range (...)
Paging with Succinct Predictions
On Minimizing Tardy Processing Time (...)
Algorithms with predictions
Memoryless Worker-Task Assignment with (...)

Clustering can be hard to explain



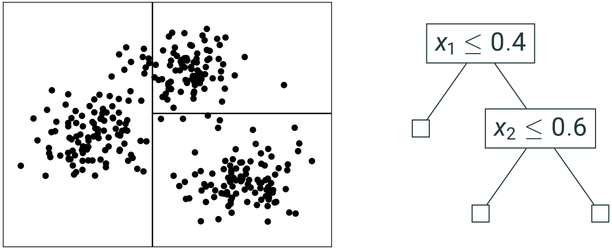
Clustering can be hard to explain



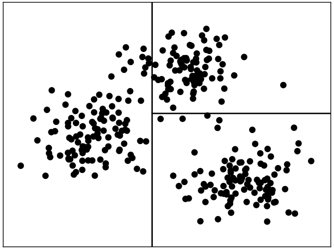
$0.6 \cdot \textit{weight} + 0.7 \cdot \textit{age} + 2 \cdot \textit{vaccinated} \leqslant 1.5$ AND

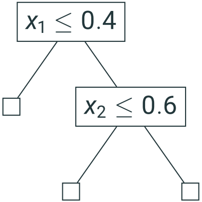
$0.9 \cdot \textit{location} + 1.4 \cdot \textit{weight} + 0.7 \cdot \textit{age} \geqslant 2.5$

Decision tree is easier to understand



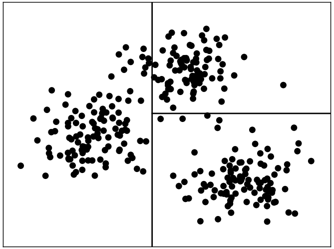
Decision tree is easier to understand

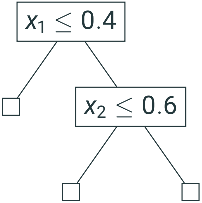




weight $\geqslant$ 100

Decision tree is easier to understand

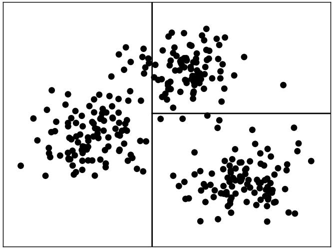


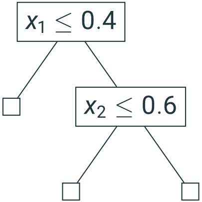


$weight \geqslant 100$ AND

$age \geqslant 90$

Decision tree is easier to understand



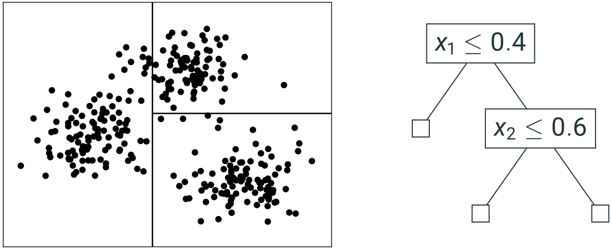


weight $\geqslant$ 100 AND

age $\geqslant$ 90 AND

unvaccinated

Explainable clustering



A *threshold tree* is a binary tree-

where each non-leaf node is an axis-aligned threshold cut.

An explainable k-clustering is one formed by a threshold tree with k leaves.

Price of explainability

How much more expensive is an optimal explainable clustering?

Price of explainability

How much more expensive is an optimal explainable clustering?

Can we find a good explainable clustering efficiently?

Price of explainability

How much more expensive is an optimal explainable clustering?

Can we find a good explainable clustering efficiently?

First introduced and studied by Moshkovitz, Dasgupta, Rashtchian, Frost (ICML 2020)

Let's focus on k-median

Input: points X in $\mathbb{R}^d$

Distance: L1-norm

i.e. $\text{dist}(x, y) = \sum_{i=1}^{d} |x_i - y_i|$

Goal: find k centers C minimizing

$\sum_{x \in X} \min_{c \in C} \text{dist}(x, y)$

Let's focus on k-median

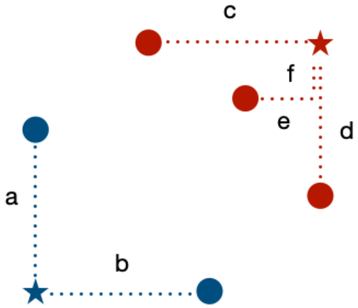
Input: points X in $\mathbb{R}^d$

Distance: L1-norm

i.e. $\text{dist}(x, y) = \sum_{i=1}^{d} |x_i - y_i|$

Goal: find k centers C minimizing

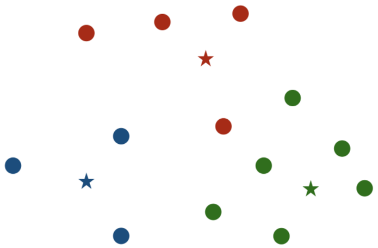
$\sum_{x \in X} \min_{c \in C} \text{dist}(x, y)$



$OPT = a + b + c + d + e + f$

General approach

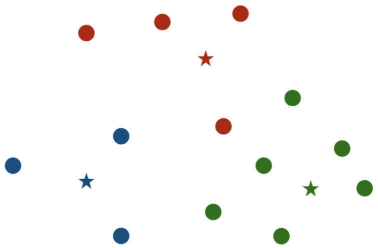
Transform given reference clustering to an explainable clustering



General approach

Transform given reference clustering to an explainable clustering

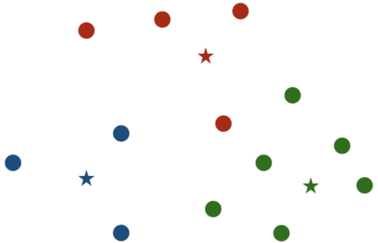
Keep splitting until one leaf per center



General approach

Transform given reference clustering to an explainable clustering

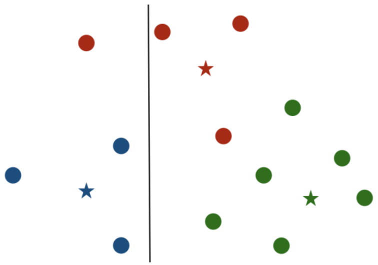
Keep splitting until one leaf per center

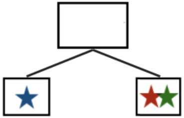


General approach

Transform given reference clustering to an explainable clustering

Keep splitting until one leaf per center

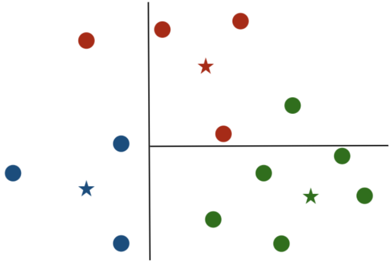


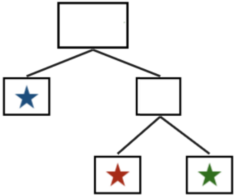


General approach

Transform given reference clustering to an explainable clustering

Keep splitting until one leaf per center





Moshkovitz–Dasgupta–Rashtchian–Frost algorithm

While there is a leaf with more than one center, **select a min-cut**

Moshkovitz–Dasgupta–Rashtchian–Frost algorithm

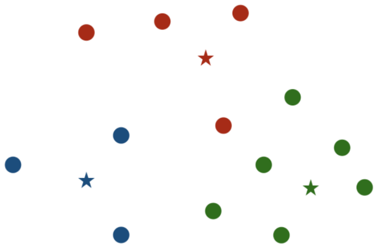
While there is a leaf with more than one center, **select a min-cut**

= a cut that separates the fewest number of points from their closest centers

Moshkovitz–Dasgupta–Rashtchian–Frost algorithm

While there is a leaf with more than one center, **select a min-cut**

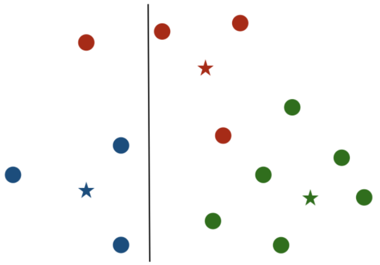
= a cut that separates the fewest number of points from their closest centers

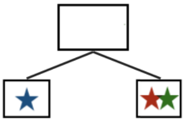


Moshkovitz–Dasgupta–Rashtchian–Frost algorithm

While there is a leaf with more than one center, **select a min-cut**

= a cut that separates the fewest number of points from their closest centers

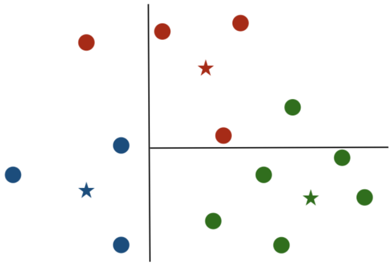


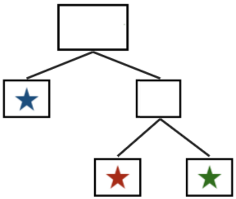


Moshkovitz–Dasgupta–Rashtchian–Frost algorithm

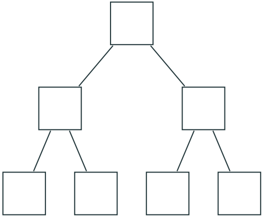
While there is a leaf with more than one center, **select a min-cut**

= a cut that separates the fewest number of points from their closest centers



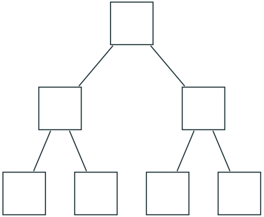


Moshkovitz–Dasgupta–Rashtchian–Frost analysis



#points separated by min-cut · distance to furthest center $\leqslant OPT$

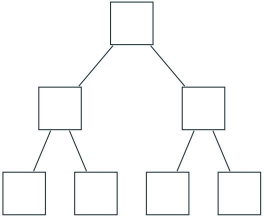
Moshkovitz–Dasgupta–Rashtchian–Frost analysis



#points separated by min-cut $\cdot$ distance to furthest center $\leqslant OPT$

$$OPT(left) + OPT(right) \leqslant OPT$$

Moshkovitz–Dasgupta–Rashtchian–Frost analysis

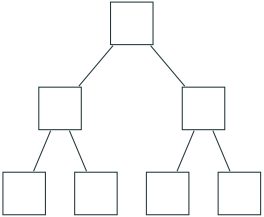


#points separated by min-cut · distance to furthest center $\leqslant OPT$

$OPT(left) + OPT(right) \leqslant OPT$

cost increase at each level $\leqslant OPT$

Moshkovitz–Dasgupta–Rashtchian–Frost analysis



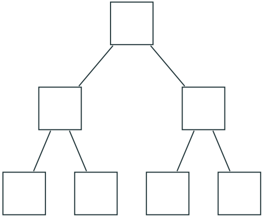
#points separated by min-cut · distance to furthest center $\leqslant OPT$

$OPT(left) + OPT(right) \leqslant OPT$

cost increase at each level $\leqslant OPT$

Price of explainability is at most **height of the tree**, and hence **at most** ***k***.

Moshkovitz–Dasgupta–Rashtchian–Frost analysis



#points separated by min-cut · distance to furthest center $\leqslant OPT$

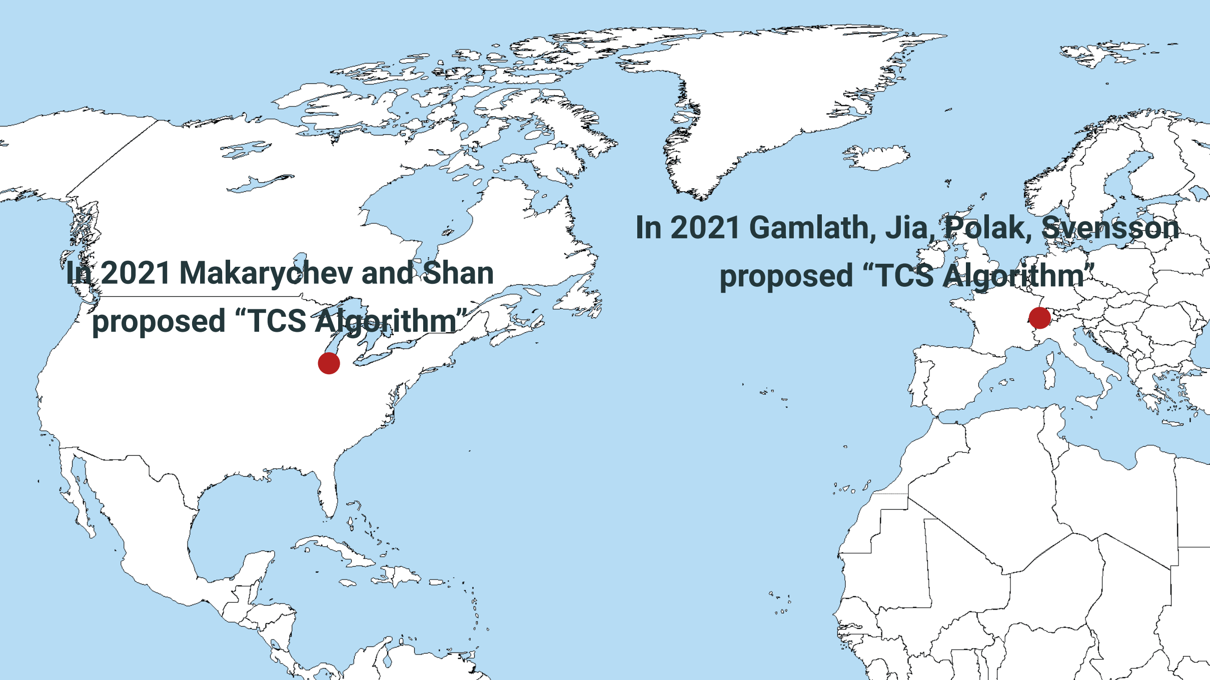
$$OPT(left) + OPT(right) \leqslant OPT$$

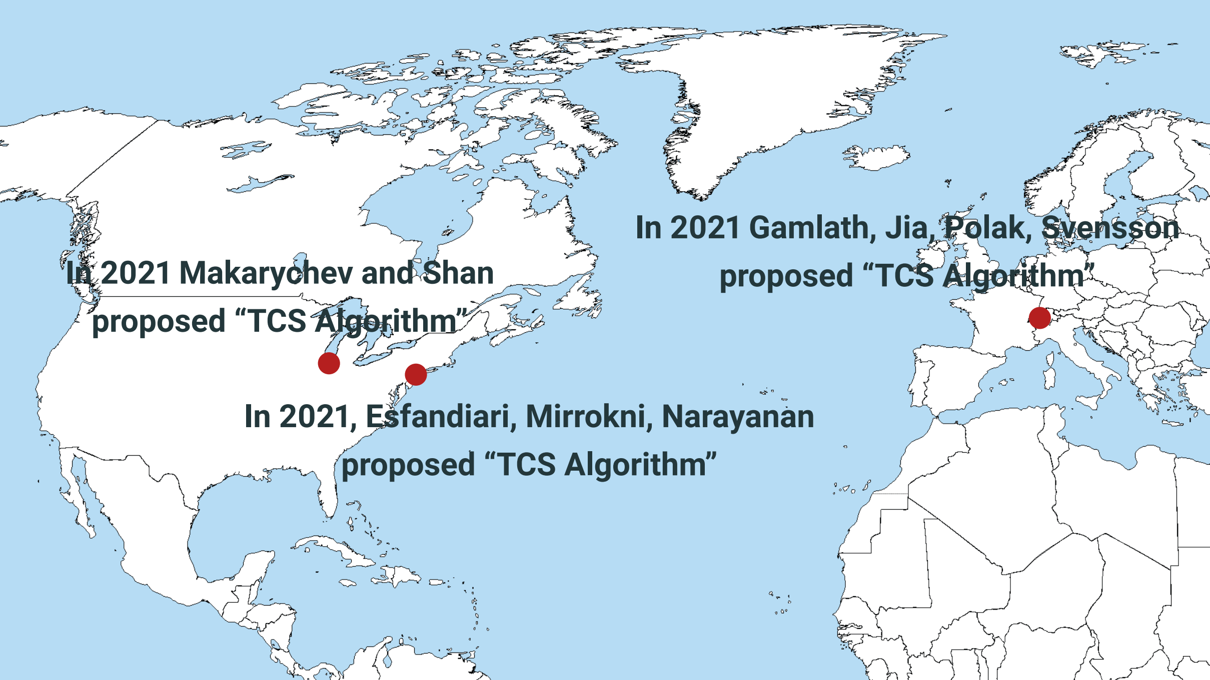
cost increase at each level $\leqslant OPT$

Price of explainability is at most **height of the tree**, and hence **at most** $\boldsymbol{k}$.

Also, there are instances where the price of explainability is **at least** $\boldsymbol{\log k}$.


In 2021 Makarychev and Shan
proposed “TCS Algorithm”


In 2021 Makarychev and Shan
proposed "TCS Algorithm"
In 2021 Gamlath, Jia, Polak, Svensson
proposed "TCS Algorithm"


In 2021 Gamlath, Jia, Polak, Svensson
proposed “TCS Algorithm”
In 2021 Makarychev and Shan
proposed “TCS Algorithm”
In 2021, Esfandiari, Mirrokni, Narayanan
proposed “TCS Algorithm”

Moshkovitz–Dasgupta–Rashtchian–Frost algorithm

While there is a leaf with more than one center, **select a min-cut**

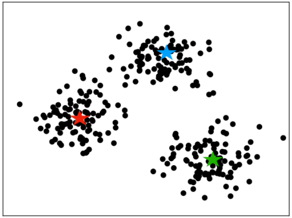
The TCS algorithm

While there is a leaf with more than one center, ~~**select a min-cut**~~ **select a cut uniformly at random**

The TCS algorithm

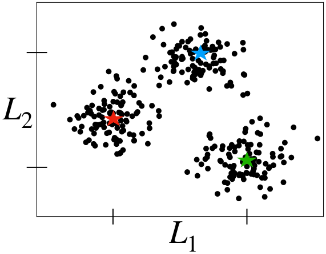
While there is a leaf with more than one center, ~~**select a min-cut**~~

select a cut uniformly at random



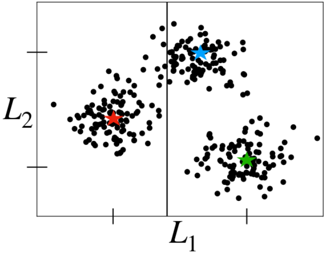
The TCS algorithm

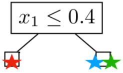
While there is a leaf with more than one center, ~~**select a min-cut**~~
select a cut uniformly at random



The TCS algorithm

While there is a leaf with more than one center, ~~**select a min-cut**~~
select a cut uniformly at random

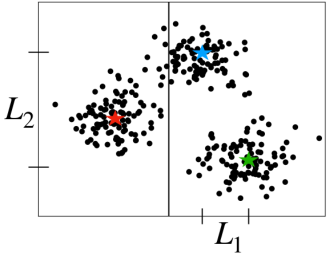


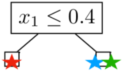


The TCS algorithm

While there is a leaf with more than one center, **~~select a min-cut~~**

select a cut uniformly at random

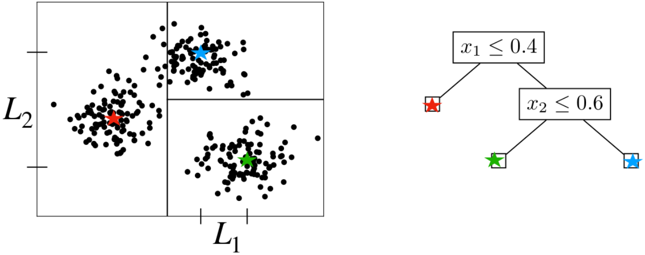




The TCS algorithm

While there is a leaf with more than one center, ~~**select a min-cut**~~

select a cut uniformly at random



The independent works in 2021

Makarychev and Shan:
$O(\log k \log \log k)$

Gamlath, Jia, Polak, Svensson:
$O(\log^2 k)$

Esfandiari, Mirrokni, Narayanan:
$O(\min(\log k \log \log k, d \log d))$

Finally, in 2023

Gupta, Pitty, Svensson, Yuan:

$O(\log k)$

Open problems

What is price of explainability for k-means?

It is between k and $k \log k$.

Open problems

What is price of explainability for k-means?

It is between k and $k \log k$.

What if we allows more than one dimension in threshold cuts?

Open problems

What is price of explainability for k-means?

It is between k and $k \log k$.

What if we allows more than one dimension in threshold cuts?

Under what **natural clusterability assumptions** we could obtain a **lower price** of explainability?

Thank you!