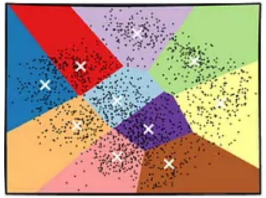
Ultrametric Cluster Hierarchies: I Want 'em All!

*Andrew Draganov**, ***Pascal Weber****,
Rasmus Jørgensen, Anna Beer, Claudia Plant, Ira Assent

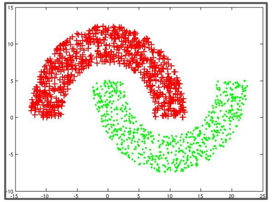
December 3rd, 2025: 11 a.m. – 2 p.m.
Exhibit Hall C,D,E

Different **Clustering** Algorithm Types

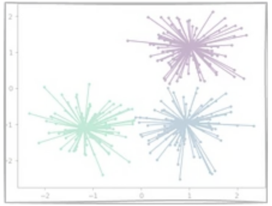
Centroid-based



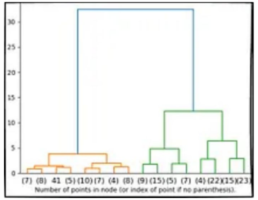
Density-based



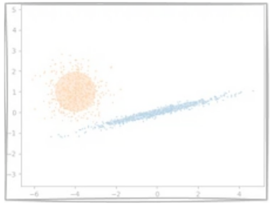
Graph-based



Hierarchical

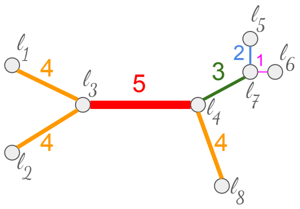


Distribution-based



Ultrametric Cluster Hierarchies

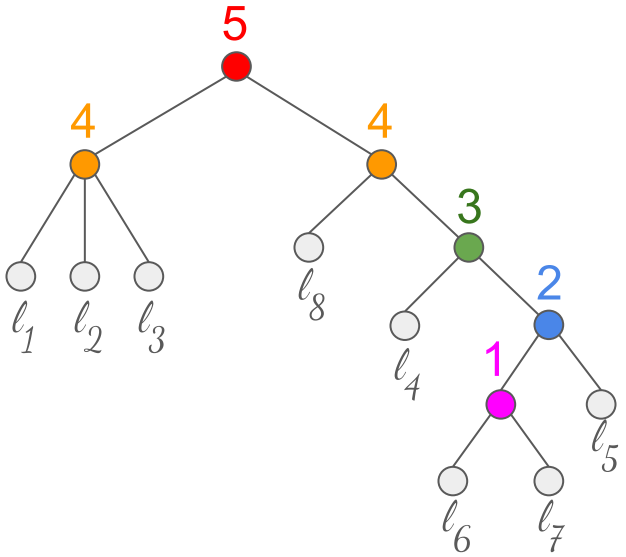
All ultrametrics are hierarchical



$$d(x, z) \leq \max \{d(x, y), d(y, z)\}$$

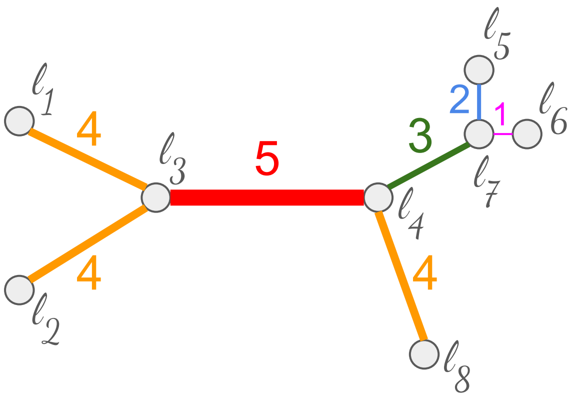
	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5	ℓ_6	ℓ_7	ℓ_8
ℓ_1	0	4	4	5	5	5	5	5
ℓ_2	4	0	4	5	5	5	5	5
ℓ_3	4	4	0	5	5	5	5	5
ℓ_4	5	5	5	0	3	3	3	4
ℓ_5	5	5	5	3	0	2	2	4
ℓ_6	5	5	5	3	2	0	1	4
ℓ_7	5	5	5	3	2	1	0	4
ℓ_8	5	5	5	4	4	4	4	0

Distance between two leaves is the value in their lowest common ancestor (LCA):

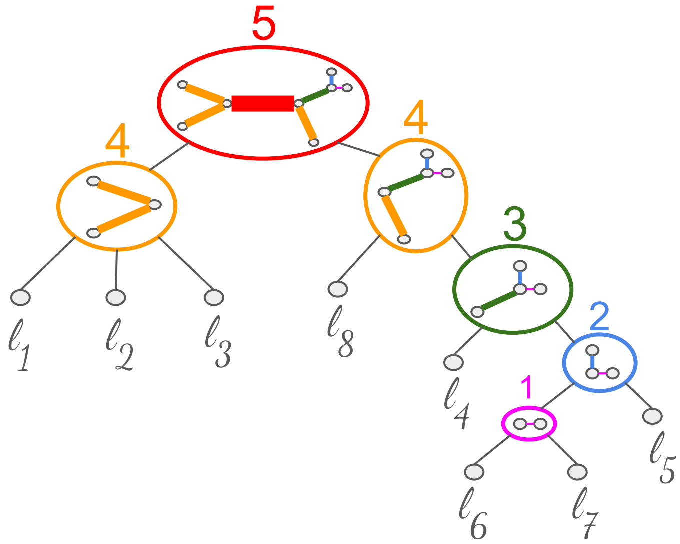


All ultrametrics are hierarchical

Minimax distance is the largest step in the minimax path between two nodes, which is always on a minimum spanning tree (MST)



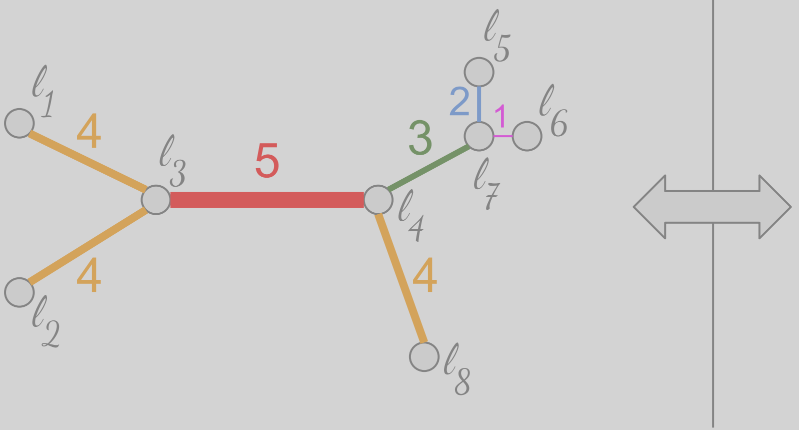
Distance between two leaves is the value in their lowest common ancestor (LCA):

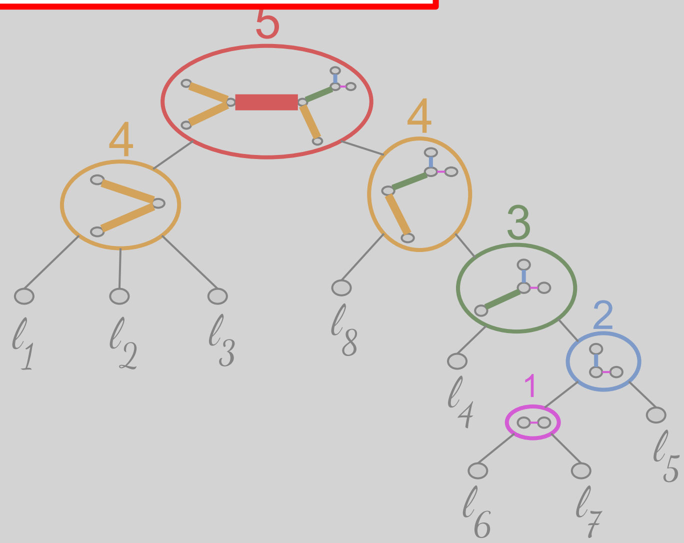


All ultrametrics are hierarchical

Minimax distance is t… ves is the value
minimax path betwee… ncestor (LCA):
always on a minimum spanning tree (MST)

Density-based clustering
corresponds to thresholding an **ultrametric**



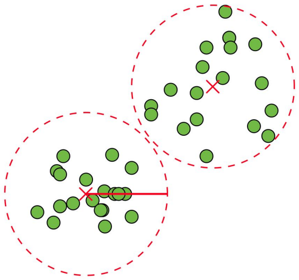


What happens if we do centroid-based clustering in ultrametrics?

- The following are NP-hard in standard metric spaces
  but can be **solved optimally in ultrametrics**:

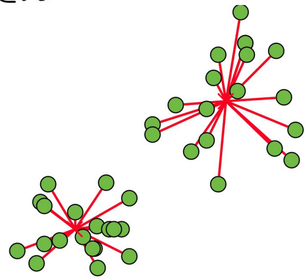
k-center

$$\max_{x \in \mathcal{X}} \min_{c \in C} d(x, c)$$



k-means

$$\sum_{x \in \mathcal{X}} \min_{c \in C} d(x, c)^2$$



What happens if we do centroid-based clustering in ultrametrics?

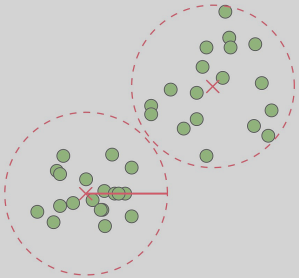
- The followi

  but can be

It takes `Sort(n)` time to find the optimal *k*-means solutions for all values of *k* in an ultrametric

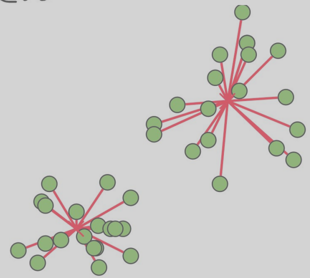
k-center

$$\max_{x \in \mathcal{X}} \min_{c \in C} d(x, c)$$



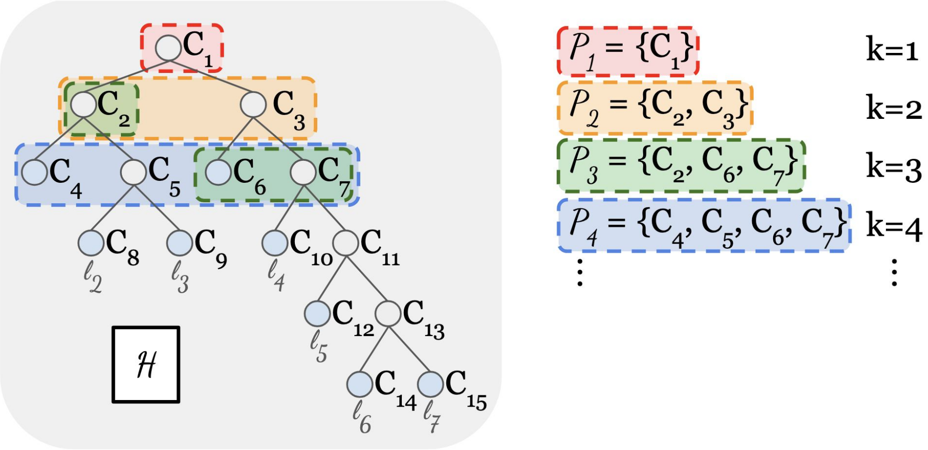
k-means

$$\sum_{x \in \mathcal{X}} \min_{c \in C} d(x, c)^2$$



The optimal *k*-means solutions are *themselves* hierarchical

The optimal *k*-means solution and the optimal (*k*+1)-means solution in an ultrametric are <u>identical</u> except that a single cluster was split in two

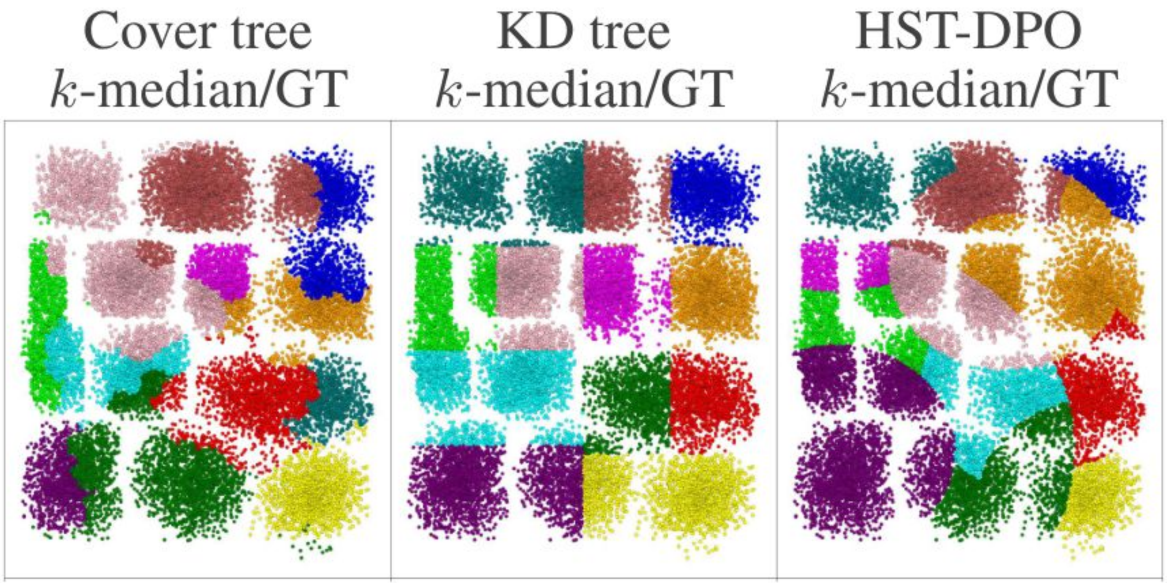


Implications

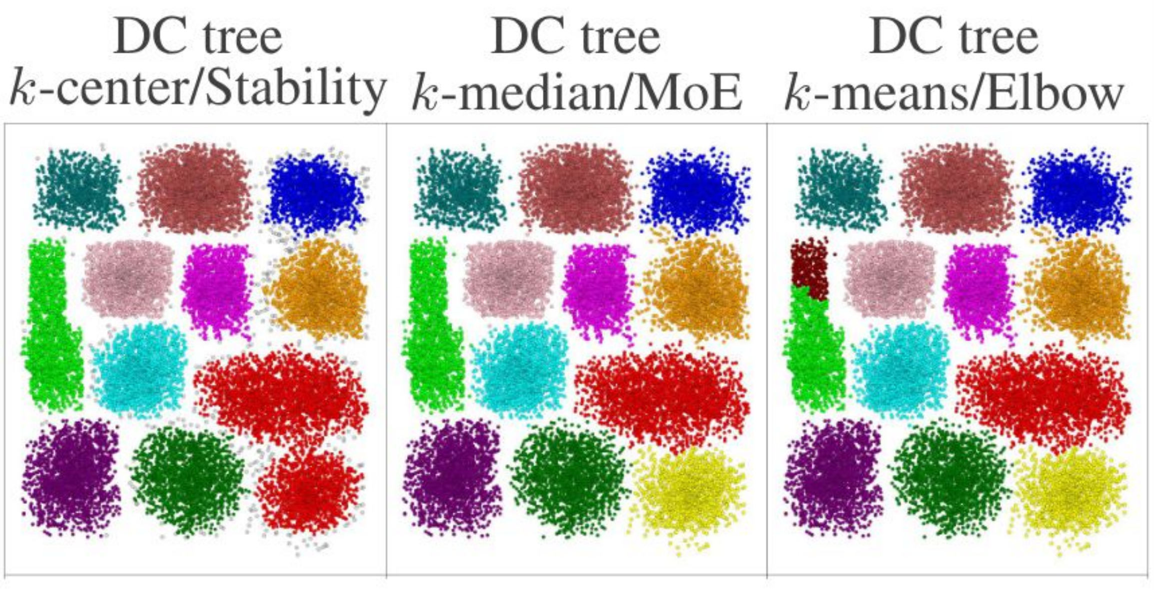
1. Solve *any* center-based clustering task in *any* ultrametric *optimally* in `Sort(n)` time
    a. Takes this time for *all* values of *k*
    b. These solutions are hierarchical

2. One can pick any clustering from this hierarchy extremely quickly (in O(n) time)
    a. Threshold the values in the tree (DBSCAN)
    b. Pick the “best” clustering by a function (HDBSCAN)
    c. Optimal clustering for a user-specified value of *k*
    d. Elbow method

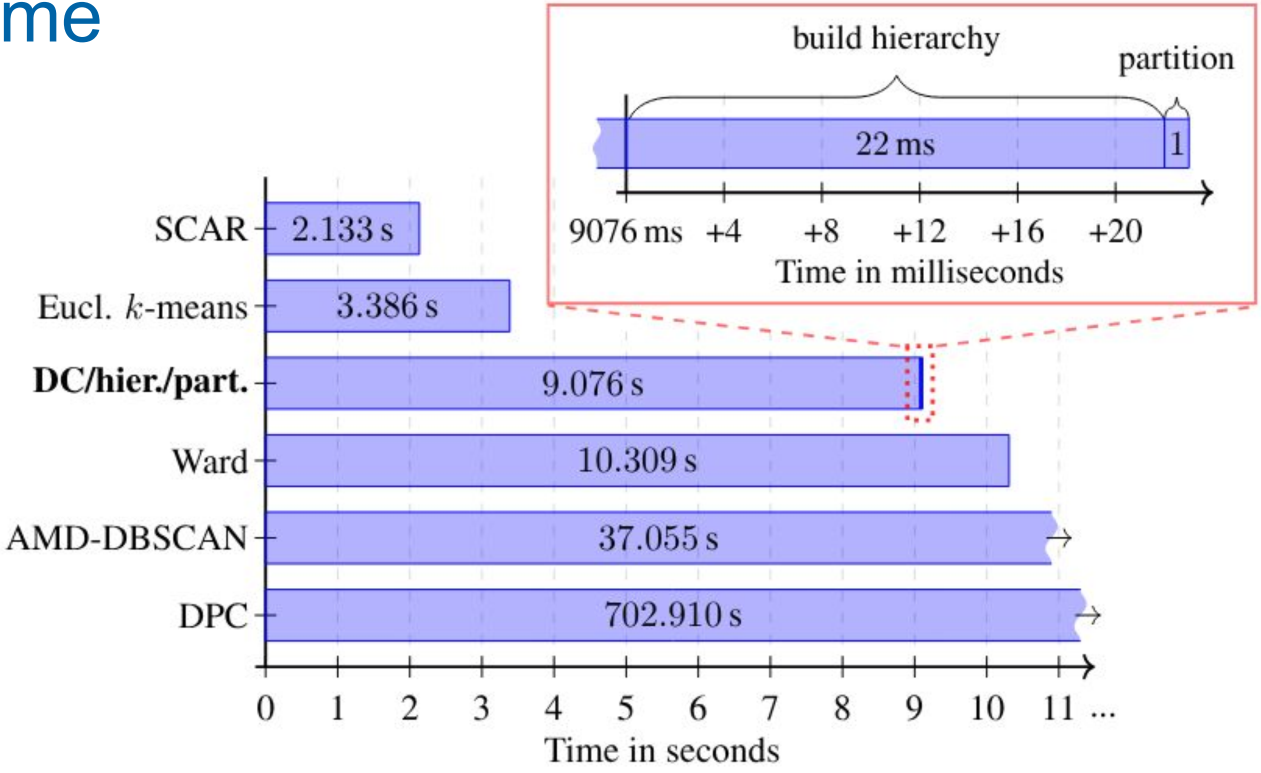
Different Ultrametrics



Different Hierarchies



Runtime


build hierarchy
partition
22 ms
1
9076 ms
+4
+8
+12
+16
+20
Time in milliseconds
SCAR
2.133 s
Eucl. k-means
3.386 s
DC/hier./part.
9.076 s
Ward
10.309 s
AMD-DBSCAN
37.055 s
DPC
702.910 s
0
1
2
3
4
5
6
7
8
9
10
11 ...
Time in seconds

Summary

- It takes **Sort(n) time** to find the optimal *k*-means solutions for **all** values of *k* **in an ultrametric**
- Outperforms other clustering algorithms in clustering accuracy

- Ultrametric Cluster Hierarchies – Implementation:

  Similarity **Hi**erarchy **P**artitioning Framework (SHiP framework)

Github Repo

pip package