Ultrametric Cluster Hierarchies: I Want ‘em All!

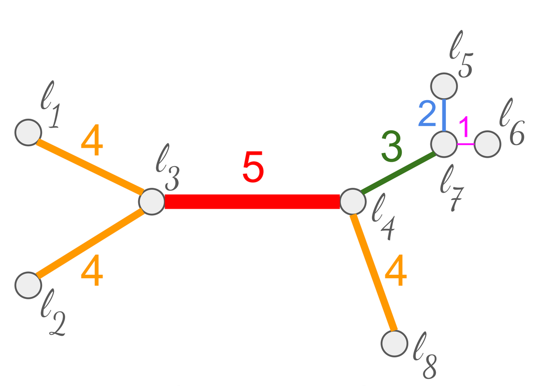
Andrew Draganov* [1,2], **Pascal Weber*** [3,4,5], Rasmus Jørgensen [1], Anna Beer [3], Claudia Plant [3,5], Ira Assent [1,2]

Similarity

All ultrametrics are hierarchical

An ultrametric is a metric which also satisfies the *strong triangle inequality*: ***d(x, z) ≤ max {d(x, y), d(y, z)}***

Every ultrametric can be represented by a **hierarchy**.

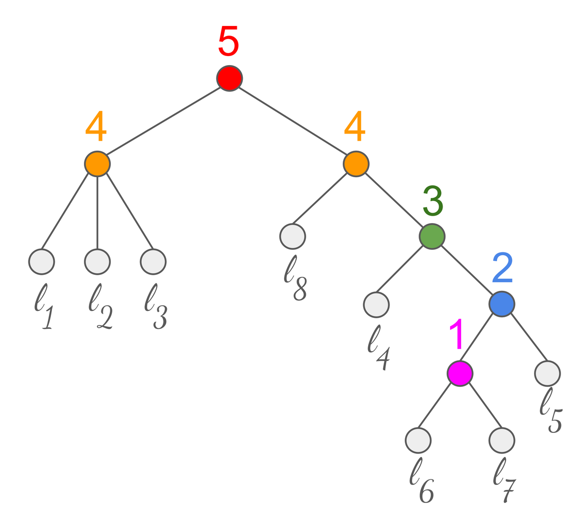


Minimax distance as ultrametric

$$
\begin{array}{c|cccccccc}
 & \ell_1 & \ell_2 & \ell_3 & \ell_4 & \ell_5 & \ell_6 & \ell_7 & \ell_8 \\
\hline
\ell_1 & 0 & 4 & 4 & 5 & 5 & 5 & 5 & 5 \\
\ell_2 & 4 & 0 & 4 & 5 & 5 & 5 & 5 & 5 \\
\ell_3 & 4 & 4 & 0 & 5 & 5 & 5 & 5 & 5 \\
\ell_4 & 5 & 5 & 5 & 0 & 3 & 3 & 3 & 4 \\
\ell_5 & 5 & 5 & 5 & 3 & 0 & 2 & 2 & 4 \\
\ell_6 & 5 & 5 & 5 & 3 & 2 & 0 & 1 & 4 \\
\ell_7 & 5 & 5 & 5 & 3 & 2 & 1 & 0 & 4 \\
\ell_8 & 5 & 5 & 5 & 4 & 4 & 4 & 4 & 0
\end{array}
$$

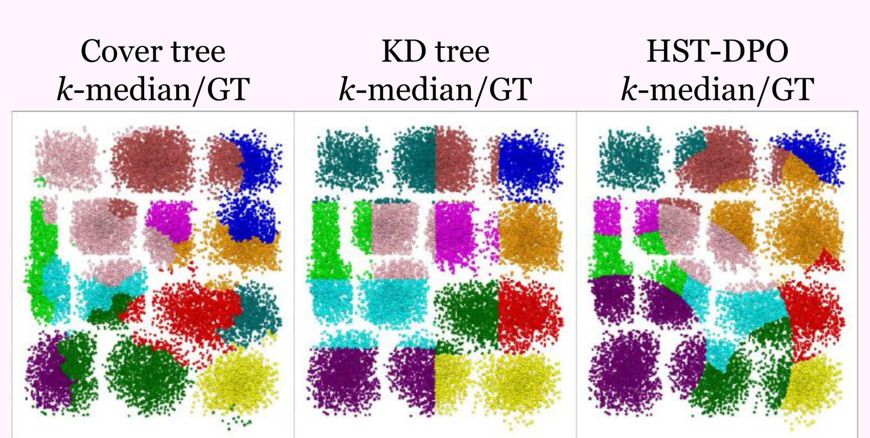
Distance matrix of minimax ultrametric

Every hierarchy with *node values growing along the paths from the leaves to the root* corresponds to an **ultrametric**.



Hierarchical representation

The distance between two nodes is the node value of their lowest common ancestor.



Clusterings of ultrametrics corresponding to Cover tree, KD tree, and another Hierarchically Well-Separated tree

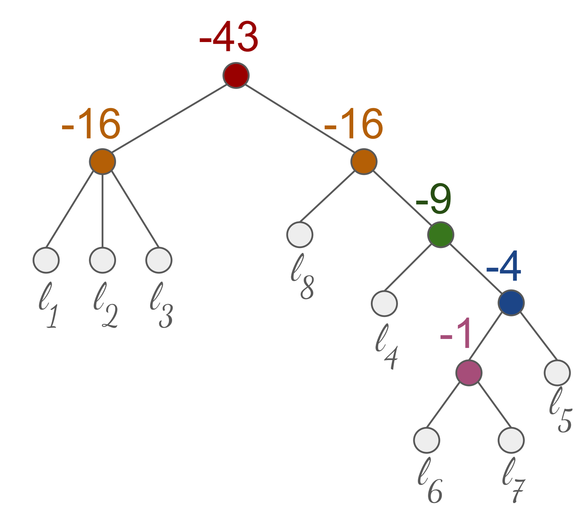
Cover trees, **KD trees**, and other **Hierarchically Well-Separated trees** can all represent data, but they all focus on different characteristics.

Hierarchy

Centroid-based clustering in ultrametrics

It takes **Sort(n) time** to find the optimal k-z (median, mean, etc.) solutions for **all** values of k **in an ultrametric**.

The **optimal solutions** are themselves hierarchical.



Cost decrease for cluster k+1

Clusterings	*Total costs*
k = 1: ℓ_1 - ℓ_2 - ℓ_3 - ℓ_8 - ℓ_4 - ℓ_6 - ℓ_7 - ℓ_5	105 = (3x5² + 4² + 3² + 2² + 1²)
k = 2: ℓ_1 - ℓ_2 - ℓ_3 | ℓ_8 - ℓ_4 - ℓ_6 - ℓ_7 - ℓ_5	62 = (2x4²) + (4² + 3² + 2² + 1²)
k = 3: ℓ_1 | ℓ_2 - ℓ_3 | ℓ_8 - ℓ_4 - ℓ_6 - ℓ_7 - ℓ_5	46 = (4²) + (4² + 3² + 2² + 1²)
k = 4: ℓ_1 | ℓ_2 | ℓ_3 | ℓ_8 - ℓ_4 - ℓ_6 - ℓ_7 - ℓ_5	30 = (4² + 3² + 2² + 1²)
k = 5: ℓ_1 | ℓ_2 | ℓ_3 | ℓ_8 | ℓ_4 - ℓ_6 - ℓ_7 - ℓ_5	14 = (3² + 2² + 1²)
k = 6: ℓ_1 | ℓ_2 | ℓ_3 | ℓ_8 | ℓ_4 | ℓ_6 - ℓ_7 - ℓ_5	5 = (2² + 1²)
k = 7: ℓ_1 | ℓ_2 | ℓ_3 | ℓ_8 | ℓ_4 | ℓ_6 - ℓ_7 | ℓ_5	1 = (1²)
k = 8: ℓ_1 | ℓ_2 | ℓ_3 | ℓ_8 | ℓ_4 | ℓ_6 | ℓ_7 | ℓ_5	0

Cluster centers in bold | The sum of k-means costs

Partitioning

Extract a clustering from a hierarchy

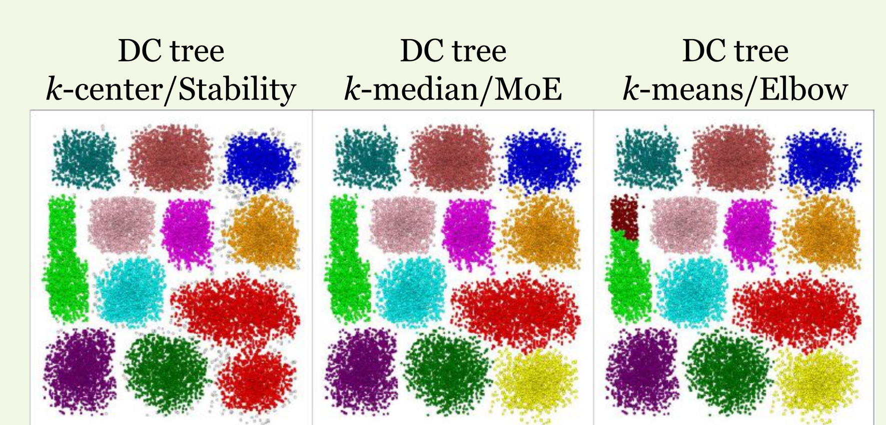
Partitioning a hierarchy can be achieved extremely fast (**O(n) time**). Possible strategies are:

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

DC tree captures density connectivity.

➡ Centroid **hierarchies** can split density-connected clusters

➡ Some **partitioning** methods find noise or determine the number of clusters automatically



Clusterings when using the DC tree as similarity, and different centroid hierarchies and partitioning methods

Our proposed **SHiP clustering framework**
(1) fits an ultrametric (similarity)
(2) computes a centroid-based hierarchy
(3) extracts a partitioning

✓ C++ Code on Github
✓ pip package of the Python interface

Github Repo

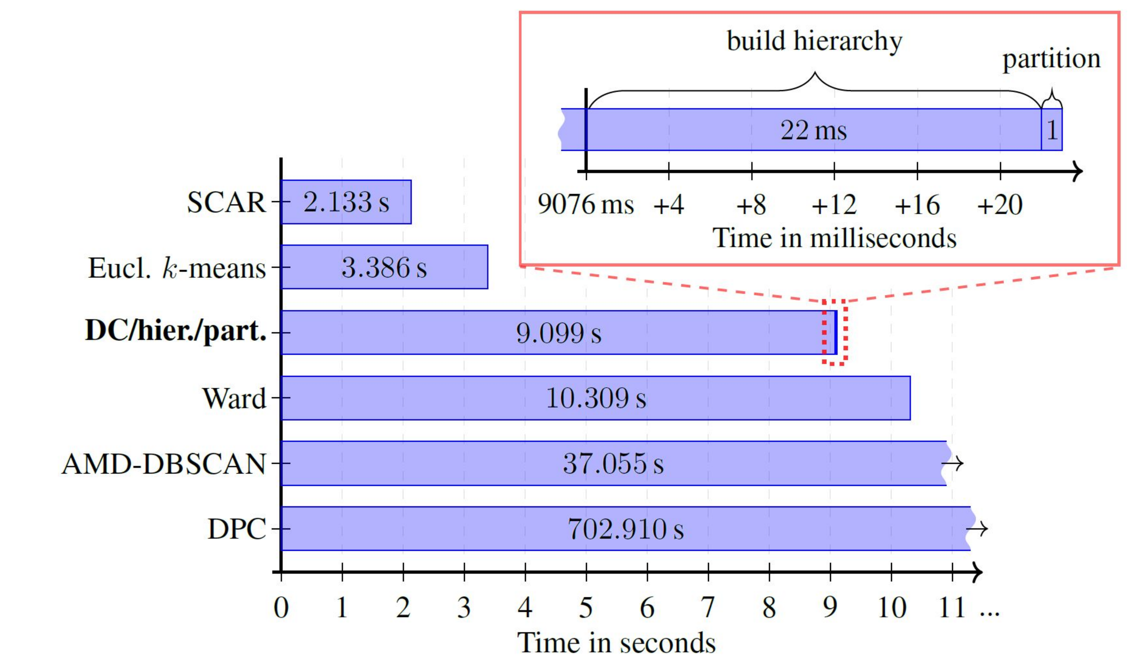
pip package

Runtimes

High efficiency through single ultrametric preprocessing

Our framework requires only a **single upfront ultrametric computation**, after that we can generate **multiple different clusterings** with **negligible additional runtime**.

Faster methods compute the clustering for only one single parameter setting.



Clustering runtimes for different algorithms of the letterrec. dataset. Our method finds clusterings for **all** possible k.

Clustering Quality

Flexibility enables deeper insights

There is no combination of hierarchy and partitioning that generally works “best”. Although DC tree/k-means/Elbow performs good in many cases, it is not universally superior to the other combinations; different pairings excel on different datasets. Our framework allows to rapidly switch between different hierarchies and partitioning methods.

		DC tree			**Cover tree**			**competitors**				
	Dataset	k-center Stability	k-median MoE	k-means Elbow	k-center Stability	k-median MoE	k-means Elbow	Eucl. k-means	SCAR	Ward	AMD-DBSCAN	DPC
Tabular Data	Boxes	90.1	**99.3**	<u>97.9</u>	2.6	42.1 ± 4.7	24.2 ± 1.6	93.5 ± 4.3	0.1 ± 0.1	95.8	63.9	25.9
	D31	79.7	42.7	82.9	46.5 ± 1.8	62.0 ± 5.4	67.7 ± 3.2	**92.0 ± 2.7**	41.7 ± 5.4	**92.0**	<u>86.4</u>	18.5
	airway	38.0	**65.9**	58.8	0.8	18.2 ± 2.4	12.0 ± 1.4	39.9 ± 2.0	−0.9 ± 0.5	43.7	31.7	<u>65.1</u>
	lactate	41.0	41.0	<u>67.5</u>	0.1	4.1 ± 0.6	1.7 ± 0.2	28.6 ± 1.1	1.5 ± 1.0	27.7	**71.5**	0.0
	HAR	30.0	46.9	**52.8**	14.7 ± 8.8	14.2 ± 4.7	9.6 ± 2.2	46.0 ± 4.5	5.5 ± 3.2	<u>49.1</u>	0.0	33.2
	letterrec.	12.1	<u>16.6</u>	**17.9**	5.8 ± 0.2	7.2 ± 0.6	6.2 ± 0.3	12.9 ± 0.6	0.4 ± 0.1	14.7 ± 0.9	7.9	0.0
	PenDigits	66.4	<u>73.1</u>	**75.4**	8.0 ± 0.8	12.0 ± 0.6	8.9 ± 0.5	55.3 ± 3.2	0.9 ± 0.3	55.2	55.6	28.8 ± 1.1
Image Data	COIL20	**81.2**	<u>72.8</u>	72.6	46.4 ± 4.4	46.6 ± 2.1	47.7 ± 2.0	58.2 ± 2.8	33.5 ± 2.0	68.6	39.2	35.9 ± 0.1
	COIL100	**80.1**	66.8	<u>70.0</u>	44.6 ± 4.2	46.6 ± 1.5	50.1 ± 1.2	56.1 ± 1.4	16.7 ± 0.8	61.4	14.2	0.2
	cmu_faces	60.2	56.6	**66.5**	8.6 ± 3.1	37.1 ± 4.1	34.2 ± 2.1	53.2 ± 4.7	38.5 ± 2.9	<u>61.6</u>	0.7	0.6
	OptDigits	55.3	**77.0**	**77.0**	40.9 ± 3.5	20.9 ± 2.3	18.1 ± 2.4	61.3 ± 6.6	14.4 ± 4.1	<u>74.6 ± 2.4</u>	63.2	0.0
	USPS	33.7	29.3	29.3	12.0 ± 1.7	8.7 ± 1.0	11.2 ± 1.5	<u>52.3 ± 1.7</u>	2.9 ± 0.9	**63.9**	0.0	21.0
	MNIST	19.7	41.7	<u>46.0</u>	11.1 ± 1.7	5.4 ± 0.6	5.4 ± 0.6	36.9 ± 1.0	1.3 ± 0.4	**52.7**	0.0	-

ARI values for the SHiP framework on the DC tree and Cover tree ultrametrics and competitors. Euclidean k-means, SCAR, and Ward are given the ground-truth value k.

References

Anna Beer, Andrew Draganov, et al. (2023). “Connecting the dots – density-connectivity distance unifies DBSCAN, k-center and Spectral Clustering”. In: SIGKDD Conference on Knowledge Discovery and Data Mining: p. 80–92.

Yuxiang Zeng, Yongxin Tong, and Lei Chen. (2021). “HST+: An efficient index for embedding arbitrary metric spaces”. In: IEEE 37th International Conference on Data Engineering: p. 648–659.

NEURAL INFORMATION PROCESSING SYSTEMS

[1]Department of Computer Science, Aarhus University, Aarhus, Denmark; [2]IAS-8: Data Analytics and Machine Learning, Forschungszentrum Jülich GmbH, Jülich, Germany; [3]Faculty of Computer Science, University of Vienna, Vienna, Austria; [4]UniVie Doctoral School Computer Science, University of Vienna, Vienna, Austria; [5]Data Science @ Uni Vienna, University of Vienna, Vienna, Austria