Max Planck Institute for Intelligent Systems, Tübingen, Germany

Foundations of Comparison-Based Hierarchical Clustering

Debarghya Ghoshdastidar, **Michaël Perrot**, and Ulrike von Luxburg

Objective: Study hierarchical clustering when only similarity comparisons are available, that is without features nor explicit similarities.

Comparison-Based Machine Learning

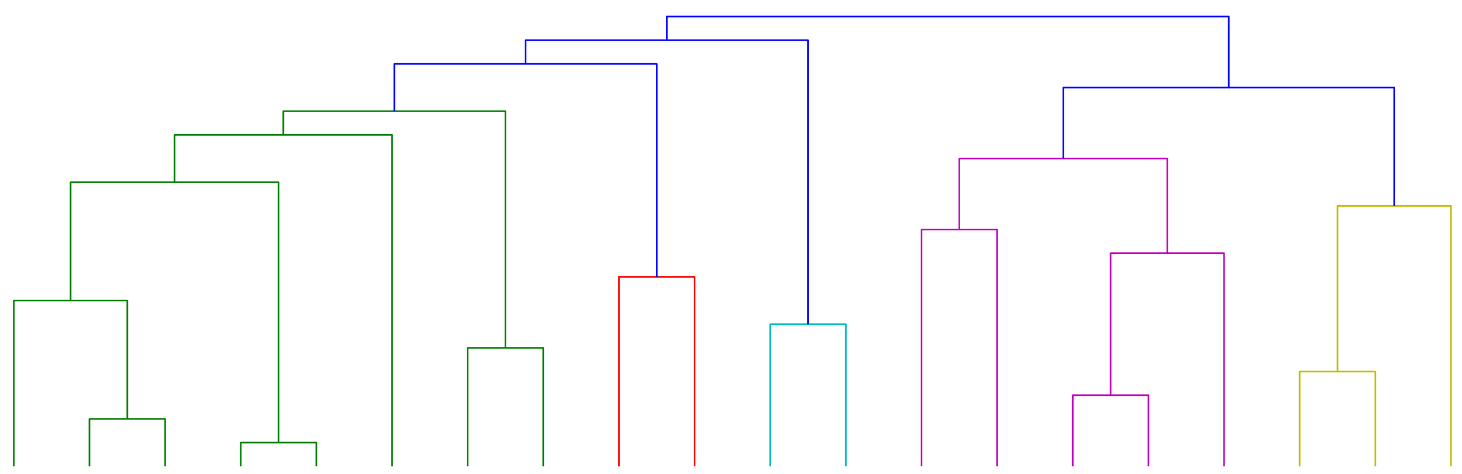
Humans are bad at giving unbiased, quantitative information. Better at giving *relative information*.
Example: The left vehicles are *more similar* to each other than the right vehicles.

Given an unknown similarity function w, the corresponding quadruplet is

$$w(\text{SUV left}, \text{SUV right}) > w(\text{Sport car}, \text{Tractor}).$$

Challenging problem: No features (coordinates), not even distances!
Given a list of quadruplets, can we solve standard machine learning tasks such as *clustering*?

Example: Let $X = \{x_i\}_{i=1}^{N}$ be a set of N cars. Can we build a dendrogram that reflects their similarities using only a limited set of quadruplets Q?



Existing solutions:

– *Embedding based methods*: Retrieve a Euclidean representation of the objects that respects the quadruplets, then use standard machine learning methods.

– *Direct methods*: Algorithms that directly handle the quadruplets to solve a specific task.

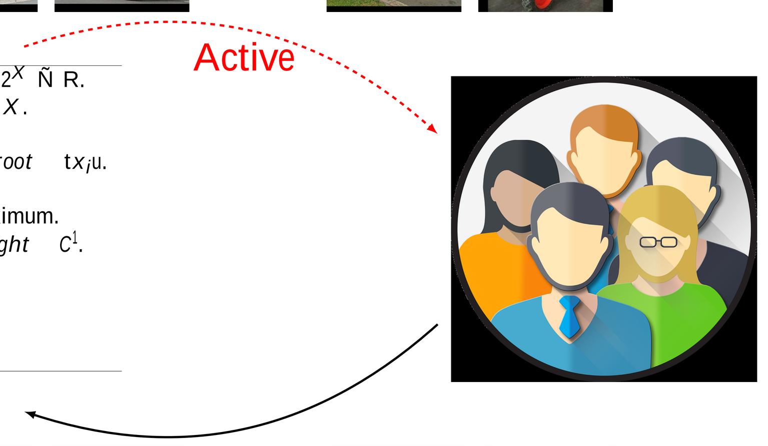
Obtaining the comparisons:

– *Actively*: quadruplets chosen by the algorithm.

– *Passively*: quadruplets given to the algorithm with no way to make new queries.

input : Set of objects $X = \{x_1, \dots, x_N\}$; Cluster-level similarity $W : 2^X \times 2^X \to \mathbb{R}$.
output: Binary tree, or dendrogram, representing a hierarchical clustering of X.
begin
  Let B be a collection of N singleton trees $C_1, \dots, C_N$ with root nodes $C_i.root = \{x_i\}$.
  while $|B| > 1$ **do**
    Let C, C' be the pair of trees in B for which $W(C.root, C'.root)$ is maximum.
    Create C'' with $C''.root = C.root \cup C'.root$, $C''.left = C$, and $C''.right = C'$.
    Add C'' to the collection B, and remove C, C'.
  end
  return The surviving element in B.
end

Algorithm 1: Agglomerative Hierarchical Clustering.



Contributions:
New algorithms for hierarchical clustering that *directly* use quadruplets.
Sufficient conditions to guarantee exact recovery of a planted model.

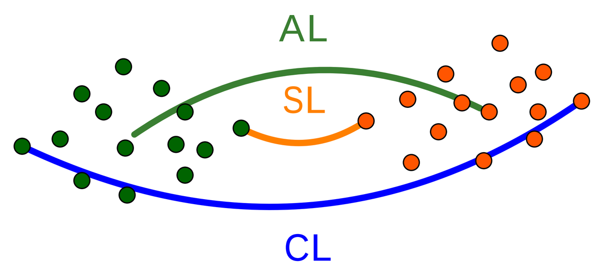
Setting

Hierarchical Clustering: Iteratively group clusters using a linkage function. Given G and G':

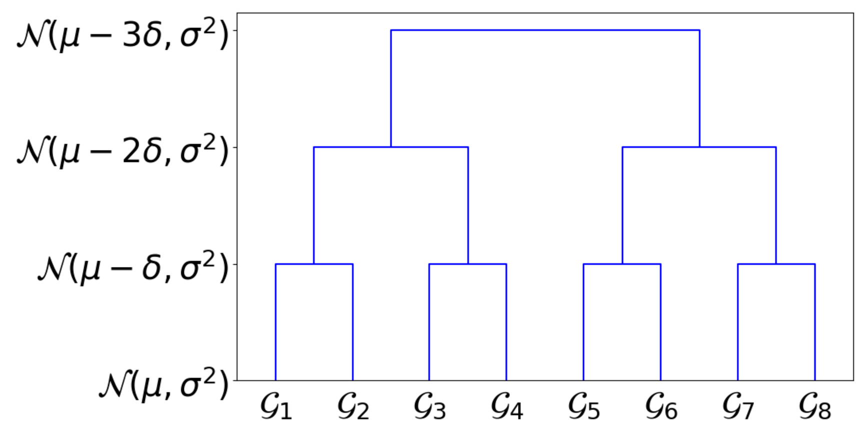
– Single Linkage (SL): $W(G, G') = \max_{x_i \in G, x_j \in G'} w_{ij}$,

– Complete Linkage (CL): $W(G, G') = \min_{x_i \in G, x_j \in G'} w_{ij}$,

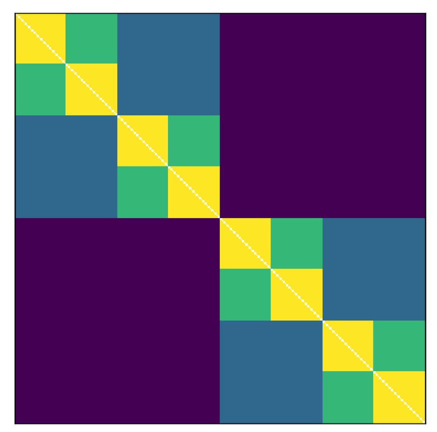
– Average Linkage (AL): $W(G, G') = \sum_{x_i \in G, x_j \in G'} \frac{w_{ij}}{|G||G'|}$.



Planted Model: A noisy hierarchical block matrix with L levels, 2^L pure clusters of size N_0 and signal to noise δ.



Hierarchical structure.



Expected similarities.

Quadruplets Kernel Average Linkage (4K–AL)

Summary: Use the quadruplets to derive a proxy for the similarities between the examples.

Kernel function: Two similar objects should behave similarly with respect to any third object.

– **Active comparisons:** Let $w_{i_0 j_0}$ be a reference similarity and S be a set of landmarks:

$$K_{ij} = \sum_{k \in S \setminus \{i,j\}} \left(\mathbb{I}(w_{ik} > w_{i_0 j_0}) - \mathbb{I}(w_{ik} < w_{i_0 j_0})\right)\left(\mathbb{I}(w_{jk} > w_{i_0 j_0}) - \mathbb{I}(w_{jk} < w_{i_0 j_0})\right).$$

– **Passive comparisons:** Use all the similarities as references and all the examples as landmarks:

$$K_{ij} = \sum_{k,l=1, k<l}^{N} \sum_{r=1}^{N} \left(\mathbb{I}_{(i,r,k,l) \in Q} - \mathbb{I}_{(k,l,i,r) \in Q}\right)\left(\mathbb{I}_{(j,r,k,l) \in Q} - \mathbb{I}_{(k,l,j,r) \in Q}\right)$$

Quadruplets-Based Average Linkage (4–AL)

Summary: Use passive comparisons to define a cluster-level similarity function.

Cluster-level similarity: Clusters G_1, G_2 are more similar to each other than G_3, G_4 if their objects are, on average, more similar to each other than the objects of G_3 and G_4:

$$\mathbb{W}_Q(G_1, G_2 \| G_3, G_4) = \sum_{x_i \in G_1} \sum_{x_j \in G_2} \sum_{x_k \in G_3} \sum_{x_l \in G_4} \frac{\mathbb{I}_{(i,j,k,l) \in Q} - \mathbb{I}_{(k,l,i,j) \in Q}}{|G_1|\,|G_2|\,|G_3|\,|G_4|}.$$

Averaging over all cluster pairs gives rise to the following linkage function:

$$W(G_p, G_q) = \sum_{r,s=1, r \neq s}^{K} \frac{\mathbb{W}_Q(G_p, G_q \| G_r, G_s)}{K(K-1)}.$$

Theory

Summary: 4K–AL and 4–AL have better guarantees than SL and CL and use less quadruplets.

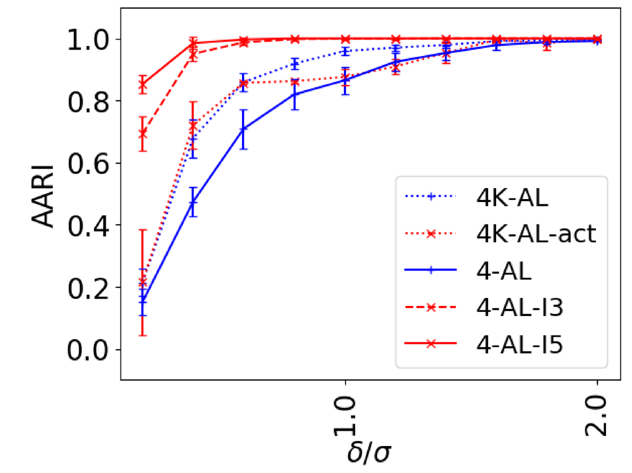
Recovery Guarantees ($L = O(1)$)

Method	Queries	Number of queries	Sufficient conditions	Remarks
SL	Active	N^2	$\delta = \Omega(\sqrt{\ln N})$	Tight!
CL	Active	N^2	$\delta = \Omega(\sqrt{\ln N})$	
4K–AL	Active	$O(N \ln N)$	$\delta = O(1)$	Near-optimal number of queries.
4K–AL	Passive	$O(N^{\frac{7}{2}} \ln N)$	$\delta = O(1)$	
4–AL	Passive	$N^3 \ln N$	$\delta = O(1)$	Needs initial clusters of size (N_0).

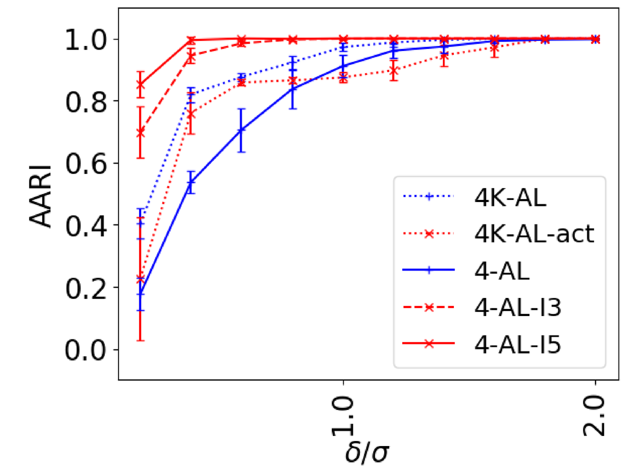
Experiments

Planted Model: SL and CL only recover the hierarchy for large signal to noise ratios while 4K–AL and 4–AL exactly recover the hierarchy for smaller signal to noise ratios.

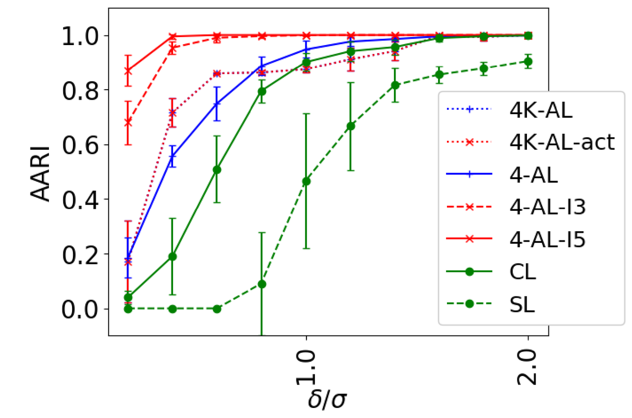
Evaluation: Average Adjusted Rand Index (AARI, higher is better).



Proportion of quadruplets $p = 0.01$.



Proportion of quadruplets $p = 0.1$.

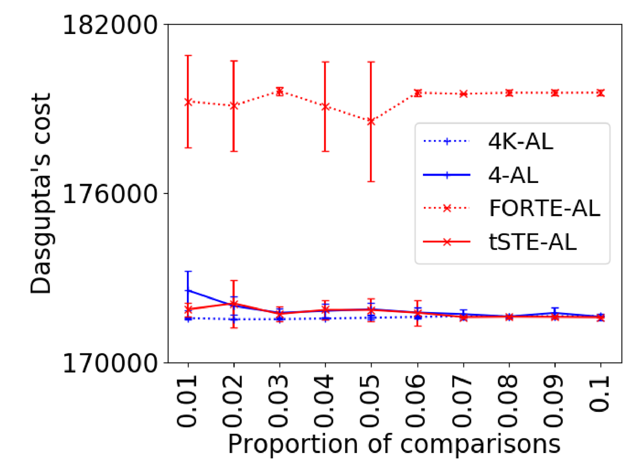


Proportion of quadruplets $p = 1$.

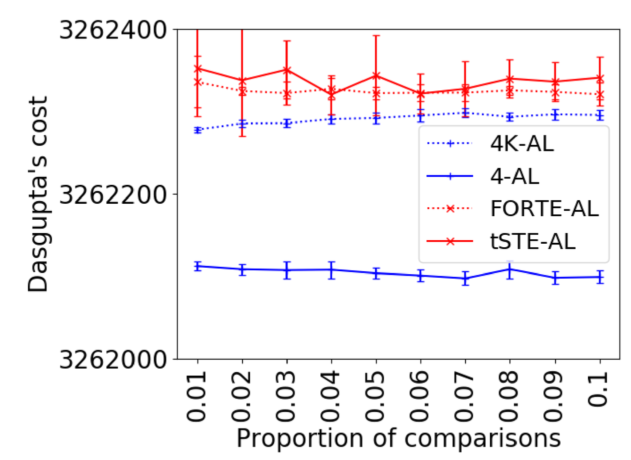
Code available online!

Standard Datasets: 4K–AL and 4–AL are on average better than embedding based methods.

Evaluation: Dasgupta's cost (lower is better).



Zoo (100 examples, 16 features).



Glass (214 examples, 9 features).