Comparison-Based Learning: Hierarchical Clustering and Classification

Michaël Perrot

Max Planck Institute for Intelligent Systems,
Tübingen, Germany

February 21, 2020

MAX-PLANCK-GESELLSCHAFT

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Example: comparing cars


$w\Big($  ,  $\Big) = ?$

Example: comparing cars

$w\Big($  ,  $\Big) = \ ?$

Example: comparing cars

$w\Big($  ,  $\Big) \lessgtr w\Big($  ,  $\Big)$

Example: comparing cars




$w\Big($ ,  $\Big) \lessgtr w\Big($ ,  $\Big)$

Ordinal Comparisons

Assumptions

- $\mathcal{X} = \{x_i\}_{i=1}^{N}$ a set of N examples,
- $w : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ an unknown similarity function ($w_{ij} = w(x_i, x_j)$),
- $\mathcal{T} = \{(x_i, x_j, x_k) : w_{ij} \geq w_{ik}$ with $i, j, k \in [N]$ and $j \neq k\}$ the set of all triplets associated with $\mathcal{X}$,
- $\mathcal{Q} = \{(x_i, x_j, x_k, x_l) : w_{ij} \geq w_{kl}$ with $i, j, k, l \in [N]$ and $j \neq l\}$ the set of all quadruplets associated with $\mathcal{X}$.

Ordinal Comparisons

Assumptions

- $\mathcal{X} = \{x_i\}_{i=1}^{N}$ a set of N examples,
- $w : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ an unknown similarity function ($w_{ij} = w(x_i, x_j)$),
- $\mathcal{T} = \{(x_i, x_k, x_j) : w_{ij} \geq w_{ik}$ with $i, j, k \in [N]$ and $j \neq k\}$ the set of all (noisy) triplets associated with $\mathcal{X}$,
- $\mathcal{Q} = \{(x_k, x_l, x_i, x_j) : w_{ij} \geq w_{kl}$ with $i, j, k, l \in [N]$ and $j \neq l\}$ the set of all (noisy) quadruplets associated with $\mathcal{X}$.

Ordinal Comparisons

Assumptions

- $\mathcal{X} = \{x_i\}_{i=1}^N$ a set of N examples,
- $w : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ an unknown similarity function ($w_{ij} = w(x_i, x_j)$),
- $\mathcal{T} = \{(x_i, x_k, x_j) : w_{ij} \geq w_{ik}$ with $i, j, k \in [N]$ and $j \neq k\}$ the set of all (noisy) triplets associated with $\mathcal{X}$,
- $\mathcal{Q} = \{(x_k, x_l, x_i, x_j) : w_{ij} \geq w_{kl}$ with $i, j, k, l \in [N]$ and $j \neq l\}$ the set of all (noisy) quadruplets associated with $\mathcal{X}$.

Is it possible to solve standard **machine learning** problems using **only** comparisons?

Comparison-Based Machine Learning

Ordinal Embedding

- **Idea:** Embed the examples in $\mathbb{R}^D$ such that the comparisons are respected and then apply standard machine learning methods.
- Works for a wide range of problems.
- Difficult to derive guarantees because of the two steps process.

Comparison-Based Machine Learning

Ordinal Embedding

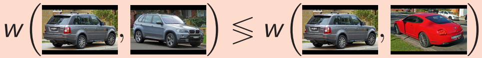
- **Idea:** Embed the examples in $\mathbb{R}^D$ such that the comparisons are respected and then apply standard machine learning methods.
- Works for a wide range of problems.
- Difficult to derive guarantees because of the two steps process.

Learning from Comparisons

- **Idea:** Design new machine learning methods able to directly handle ordinal comparisons.
- Each new problem requires the development of a new method.
- Easier to derive theoretical results.

Obtaining the Comparisons

Actively

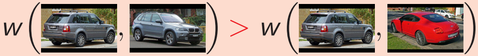


Algorithm 1 TripletBoost: boosting with triplet classifiers

Input: $S = \{(x_i, y_i)\}_{i=1}^N$ a set of N examples, $T = \{(x_i, x_j, x_k) : x_j \neq x_k\}$ a set of m triplets.

Output: $H(\cdot)$ a strong classifier.

```
Let 𝒲_1 be the empirical uniform distribution: ∀(x_i, y_i) ∈ S, ∀y ∈ 𝒴, w_{1,x_i,y} = 1/(N|𝒴|).
for c = 1, ..., C do
    Choose a triplet classifier h_c according to 𝒲_c.
    Compute the weight α_c of h_c according to its performance on 𝒲_c.
    Update the weights of the examples to obtain a new distribution 𝒲_{c+1}.
end for
return H(·) = argmax_{y∈𝒴} ( Σ_{c=1}^{C} α_c 𝕀_{h_c(·)≠ϑ ∧ y∈h_c(·)} )
```



Obtaining the Comparisons

Passively

Algorithm 1 TripletBoost: boosting with triplet classifiers

Input: $S = \{(x_i, y_i)\}_{i=1}^{N}$ a set of N examples, $T = \{(x_i, x_j, x_k) : x_j \neq x_k\}$ a set of m triplets.

Output: $H(\cdot)$ a strong classifier.

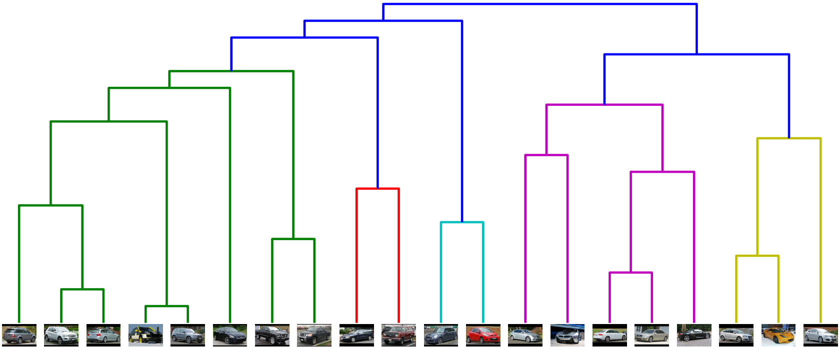
Let $\mathcal{W}_1$ be the empirical uniform distribution: $\forall (x_i, y_i) \in S, \forall y \in \mathcal{Y}, w_{1,x_i,y} = \frac{1}{N|\mathcal{Y}|}$.
for $c = 1, \ldots, C$ **do**
    Choose a triplet classifier h_c according to $\mathcal{W}_c$.
    Compute the weight α_c of h_c according to its performance on $\mathcal{W}_c$.
    Update the weights of the examples to obtain a new distribution $\mathcal{W}_{c+1}$.
end for
return $H(\cdot) = \arg\max_{y \in \mathcal{Y}} \left(\sum_{c=1}^{C} \alpha_c \mathbb{I}_{h_c(\cdot) \neq \vartheta \wedge y \in h_c(\cdot)} \right)$

$w\Big($  ,  $\Big) > w\Big($  ,  $\Big)$

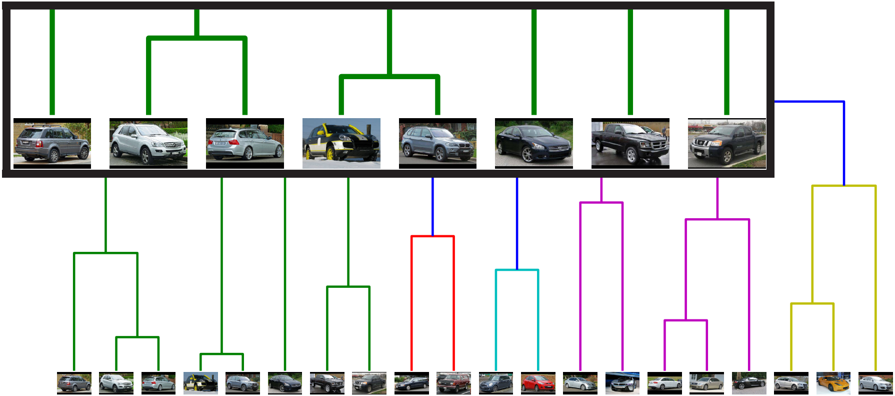
Focus: Hierarchical Clustering

Joint work with **Debarghya Ghoshdastidar** and **Ulrike von Luxburg**.
Accepted to NeurIPS 2019.

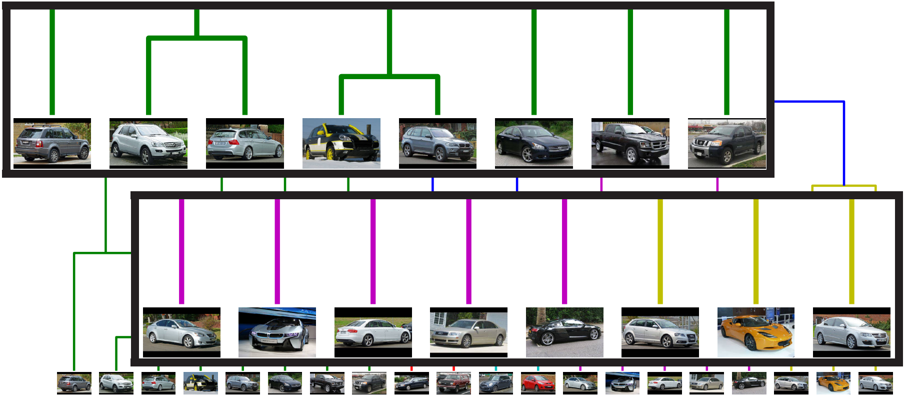
Example: Cars Dendrogram



Example: Cars Dendrogram



Example: Cars Dendrogram



Hierarchical Clustering: Bottom-Up Approach

Algorithm

- Start with clusters containing only **one example**,
- At each iteration, greedily merge the two clusters which are most similar with respect to a linkage function,
- Stop when all the examples are in the same cluster.

Hierarchical Clustering: Bottom-Up Approach

Algorithm

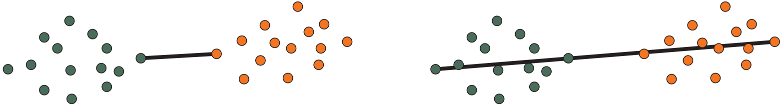
- Start with clusters containing only **one example**,
- At each iteration, greedily merge the two clusters which are most similar with respect to a linkage function,
- Stop when all the examples are in the same cluster.

Linkage Functions

A function $W : 2^{\mathcal{X}} \times 2^{\mathcal{X}} \rightarrow \mathbb{R}$. Given two clusters G_p and G_q:

- Single linkage (SL): $W(G_p, G_q) = \max_{x_i \in G_p, x_j \in G_q} w_{ij}$
- Complete linkage (CL): $W(G_p, G_q) = \min_{x_i \in G_p, x_j \in G_q} w_{ij}$
- Average linkage (AL): $W(G_p, G_q) = \frac{1}{|G_p||G_q|} \sum_{x_i \in G_p, x_j \in G_q} w_{ij}$

Hierarchical Clustering: Bottom-Up Approach



Linkage Functions

A function $W : 2^{\mathcal{X}} \times 2^{\mathcal{X}} \to \mathbb{R}$. Given two clusters G_p and G_q:

- Single linkage (SL): $W(G_p, G_q) = \max_{x_i \in G_p, x_j \in G_q} w_{ij}$
- Complete linkage (CL): $W(G_p, G_q) = \min_{x_i \in G_p, x_j \in G_q} w_{ij}$
- Average linkage (AL): $W(G_p, G_q) = \frac{1}{|G_p||G_q|} \sum_{x_i \in G_p, x_j \in G_q} w_{ij}$

Comparison-Based Hierarchical Clustering

Objective: Comparison-Based Hierarchical Clustering

- Comparison-based linkage functions
- Theoretical results on a planted hierarchical model

Comparison-Based Hierarchical Clustering

Objective: Comparison-Based Hierarchical Clustering

- Comparison-based linkage functions
- Theoretical results on a planted hierarchical model

Assumptions

- $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ a set of N examples,
- $w : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ an unknown similarity function ($w_{ij} = w(x_i, x_j)$),
- $\mathcal{Q} = \{(x_i, x_j, x_k, x_l) : w_{ij} \geq w_{kl}$ with $i, j, k, l \in [N]$ and $j \neq l\}$ a set of quadruplets associated with $\mathcal{X}$.

Comparison-Based Single/Complete Linkage

Given K clusters $G_1, \ldots, G_K$, the two merged clusters are chosen as $G, G' = \underset{G_p, G_q}{\arg\max}\, W(G_p, G_q)$.

Standard single linkage (SL): $W(G_p, G_q) = \max_{x_i \in G_p, x_j \in G_q} w_{ij}$

Standard complete linkage (CL): $W(G_p, G_q) = \min_{x_i \in G_p, x_j \in G_q} w_{ij}$

Comparison-Based Single/Complete Linkage

Given K clusters $G_1, \ldots, G_K$, the two merged clusters are chosen as $G, G' = \underset{G_p, G_q}{\arg\max}\, W(G_p, G_q)$.

Standard single linkage (SL): $W(G_p, G_q) = \max_{x_i \in G_p, x_j \in G_q} w_{ij}$

Standard complete linkage (CL): $W(G_p, G_q) = \min_{x_i \in G_p, x_j \in G_q} w_{ij}$

Idea: Assume that the similarity w is **transitive**, finding the two clusters with maximum similarity only requires quadruplet comparisons.

Comparison-Based Single/Complete Linkage

Given K clusters $G_1, \ldots, G_K$, the two merged clusters are chosen as $G, G' = \arg\max_{G_p, G_q} W(G_p, G_q)$.

Standard single linkage (SL): $W(G_p, G_q) = \max_{x_i \in G_p, x_j \in G_q} w_{ij}$

Standard complete linkage (CL): $W(G_p, G_q) = \min_{x_i \in G_p, x_j \in G_q} w_{ij}$

Idea: Assume that the similarity w is **transitive**, finding the two clusters with maximum similarity only requires quadruplet comparisons.

Single linkage (SL) and complete linkage (CL) are **inherently based on comparisons**.

Comparison-Based Average Linkage

Standard average linkage (AL): $W(G_p, G_q) = \frac{1}{|G_p|\,|G_q|} \sum_{x_i \in G_p, x_j \in G_q} w_{ij}.$

Comparison-Based Average Linkage

Standard average linkage (AL): $W(G_p, G_q) = \frac{1}{|G_p|\,|G_q|} \sum_{x_i \in G_p, x_j \in G_q} w_{ij}.$

Idea: Using quadruplets, estimate the relative similarity between two pairs of clusters.

Quadruplets-Based Average Linkage: 4–AL

$$\mathbb{W}_{\mathcal{Q}}\left(G_1, G_2 \| G_3, G_4\right) = \sum_{x_i \in G_1} \sum_{x_j \in G_2} \sum_{x_k \in G_3} \sum_{x_l \in G_4} \frac{\mathbb{I}_{(x_i, x_j, x_k, x_l) \in \mathcal{Q}} - \mathbb{I}_{(x_k, x_l, x_i, x_j) \in \mathcal{Q}}}{|G_1|\,|G_2|\,|G_3|\,|G_4|},$$

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

Comparison-Based Average Linkage

Standard average linkage (AL): $W(G_p, G_q) = \frac{1}{|G_p|\,|G_q|} \sum_{x_i \in G_p, x_j \in G_q} w_{ij}$.

Idea: Using quadruplets, derive a proxy for the similarity score w.

Quadruplets Kernel Average Linkage: 4K–AL

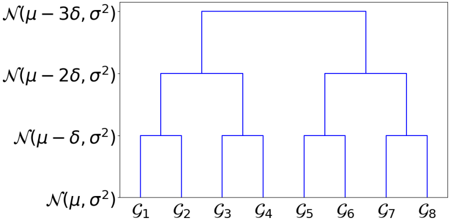
- **Active comparisons:** $w_{i_0 j_0}$ is a reference similarity and S is a set of landmarks:

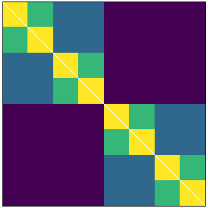
$$\hat{w}_{ij} = \sum_{k \in S \setminus \{i,j\}} \left(\mathbb{I}_{(w_{ik} > w_{i_0 j_0})} - \mathbb{I}_{(w_{ik} < w_{i_0 j_0})}\right)\left(\mathbb{I}_{(w_{jk} > w_{i_0 j_0})} - \mathbb{I}_{(w_{jk} < w_{i_0 j_0})}\right).$$

- **Passive comparisons:**

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

Planted Hierarchical Model





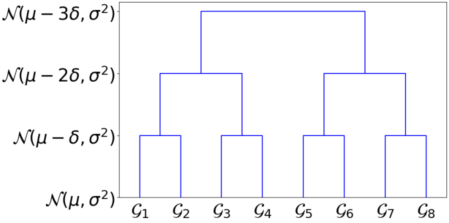
A hierarchy with L levels and N_0 objects per cluster: $N = 2^L N_0$.

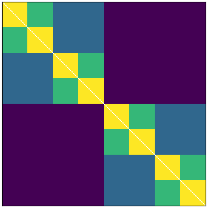
The similarities $\{w_{ij}\}_{1\leq i<j\leq N}$ are **random**, **mutually independent**, and,

- Normally distributed, $w_{ij} \sim \mathcal{N}\left(\mu_{ij}, \sigma^2\right)$,
- Symmetric, $w_{ji} = w_{ij}$,
- $w_{ii} = \infty$.

The hierarchy is introduced by specifying the means $\mu_{ij} = \mu - (L-\ell)\delta$.

Planted Hierarchical Model





A hierarchy with L levels and N_0 objects per cluster: $N = 2^L N_0$.

The similarities $\{w_{ij}\}_{1 \leq i < j \leq N}$ are **random**, **mutually independent**, and,

- Normally distributed, $w_{ij} \sim \mathcal{N}\left(\mu_{ij}, \sigma^2\right)$,
- Symmetric, $w_{ji} = w_{ij}$,
- $w_{ii} = \infty$.

The hierarchy is introduced by specifying the means $\mu_{ij} = \mu - (L - \ell)\delta$.

Objective: Obtain some **sufficient conditions** under which the different comparison-based algorithms **exactly recover** the hierarchy.

Theoretical Results: Summary

Recovery Guarantees $(L = \mathcal{O}(1))$

Method	Queries	# queries	Sufficient conditions	Remarks
SL	Active	$\Omega\left(N^2\right)$	$\frac{\delta}{\sigma} = \Omega\left(\sqrt{\ln N}\right)$	Tight!
CL	Active	$\Omega\left(N^2\right)$	$\frac{\delta}{\sigma} = \Omega\left(\sqrt{\ln N}\right)$	
4K–AL	Active	$\mathcal{O}(N \ln N)$	$\frac{\delta}{\sigma} = \mathcal{O}(1)$	Near-optimal # queries.
4K–AL	Passive	$\mathcal{O}\left(N^{\frac{7}{2}} \ln N\right)$	$\frac{\delta}{\sigma} = \mathcal{O}(1)$	
4–AL	Passive	$\Omega\left(N^3 \ln N\right)$	$\frac{\delta}{\sigma} = \mathcal{O}(1)$	Initial clusters: $\Omega(N_0)$.

Planted Model: Setup

Goal: Empirically verify that the proposed approaches are able to recover the planted hierarchy.

Planted model parameters:

- Mean: $\mu = 0.8$,
- Standard deviation: $\sigma = 0.1$,
- Number of levels: $L = 3$,
- Size of clusters: $N_0 = 30$,
- Separation: $\delta \in \{0.01, 0.02, \ldots, 0.2\}$,
- Proportion of quadruplets: $p \in \{0.01, \ldots, 0.1, 1\}$.

Measure of performance: Given two hierarchies $\mathcal{C}$ and $\mathcal{C}'$, let $\mathcal{C}^\ell$ and $\mathcal{C}'^\ell$ be the partitions at level ℓ. The Averaged Adjusted Rand Index (AARI) is:

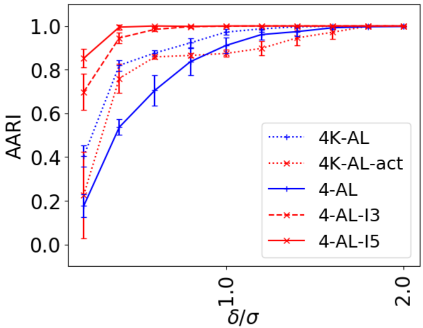
$$\text{AARI}\left(\mathcal{C}, \mathcal{C}'\right) = \frac{1}{L} \sum_{\ell \in \{1, \ldots, L\}} \text{ARI}\left(\mathcal{C}^\ell, \mathcal{C}'^\ell\right)$$

where $\text{ARI}\left(\mathcal{C}^\ell, \mathcal{C}'^\ell\right)$ is the Adjusted Rand Index [Hubert and Arabie, 1985].

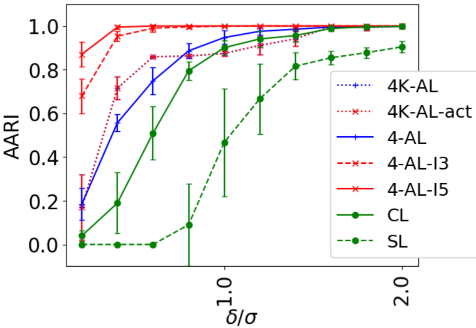
Planted Model: Results

Planted model parameters: $\mu = 0.8$, $\sigma = 0.1$, $L = 3$, $N_0 = 30$, $\delta \in \{0.01, 0.02, \ldots, 0.2\}$.

$p = 0.1$



$p = 1$



Toy Datasets: Experimental Setup

Comparisons: Generated using the cosine similarity,

$$w_{ij} = \frac{\langle x_i, x_j \rangle}{\|x_i\| \, \|x_i\|}.$$

Baselines: Ordinal embedding followed by standard average linkage,

- FORTE [Jain et al., 2016],
- tSTE [Van Der Maaten and Weinberger, 2012].

Measure of performance: A cost function for hierarchies proposed by Dasgupta [2015]:

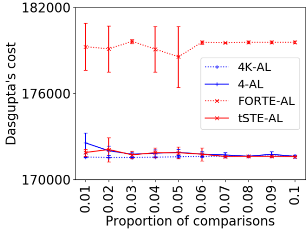
$$\text{cost}(\mathcal{C}, w) = \sum_{x_i, x_j \in \mathcal{X}} w_{ij} \left| \mathcal{C}^{lca}(x_i, x_j) \right|$$

where $\mathcal{C}^{lca}(x_i, x_j)$ is the smallest cluster containing both x_i and x_j.

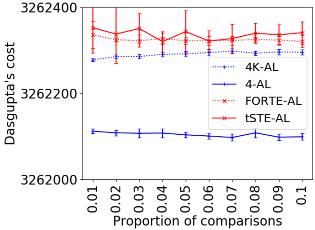
Toy Datasets: Results

Ordinal embedding parameters: $D = 2$.

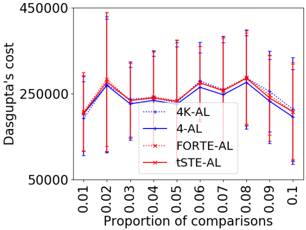
Zoo (100 objects, 16 features)



Glass (214 objects, 9 features)



20news (200 objects, 100 features)



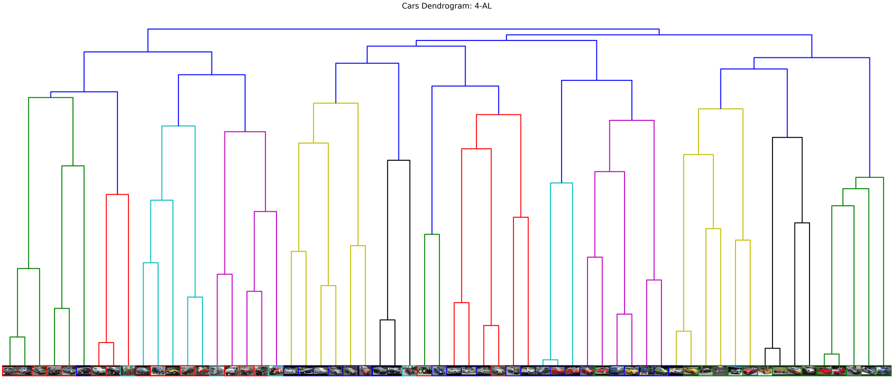
Car Dataset: Experimental Setup

Created by Kleindessner and von Luxburg [2017]:

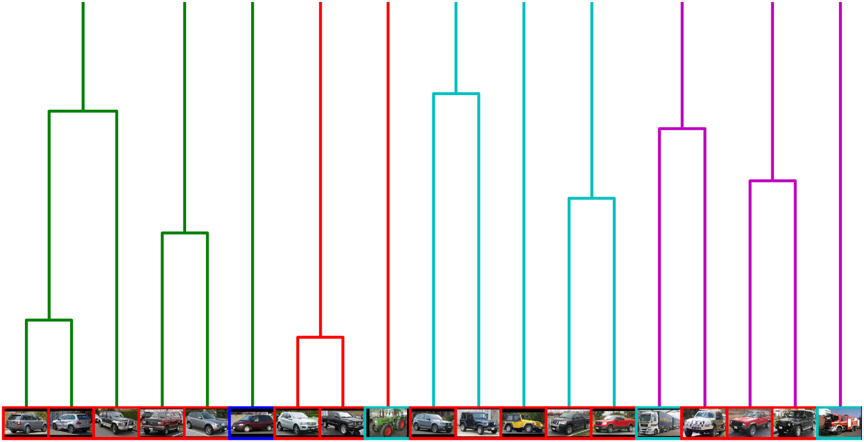
- 60 car images,
- 6056 statements of the form x_i **is most central among** (x_i, x_j, x_k).

In our setting, it corresponds to 12112 quadruplets: $w_{ij} > w_{jk}$ and $w_{ik} > w_{jk}$.

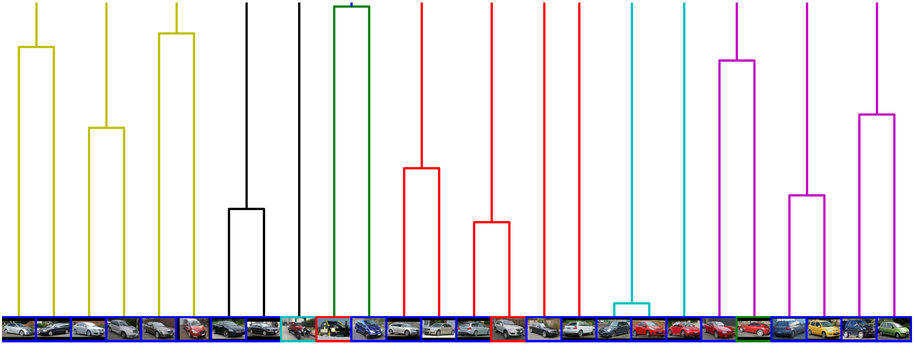
Car Dataset: Results



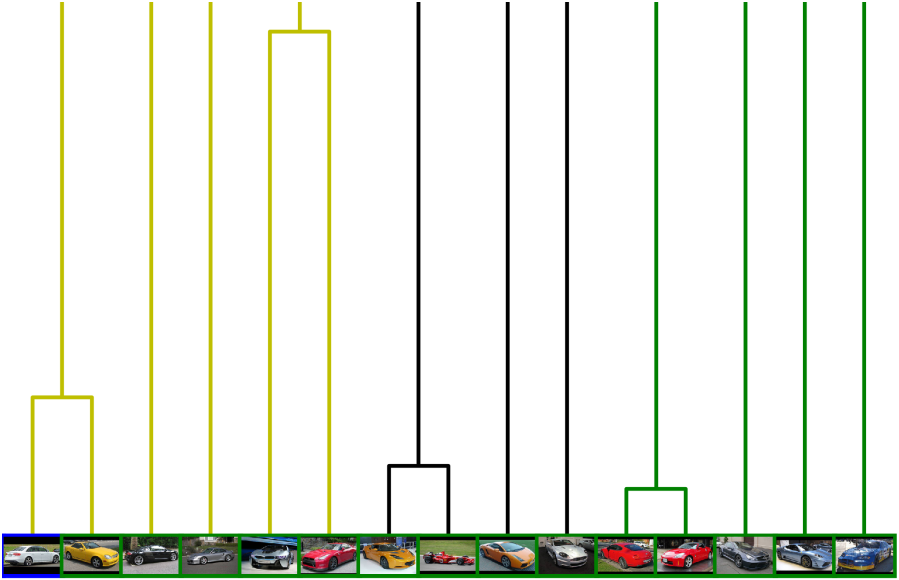
Car Dataset: Results



Car Dataset: Results



Car Dataset: Results



Conclusion

Comparison-Based Hierarchical Clustering

- Single linkage and complete linkage are inherently comparison-based,
- Several linkage functions for comparison-based average linkage,
- Recovery guarantees for a planted hierarchical model,
- Empirically well-behaved.

Conclusion

Comparison-Based Hierarchical Clustering

- Single linkage and complete linkage are inherently comparison-based,
- Several linkage functions for comparison-based average linkage,
- Recovery guarantees for a planted hierarchical model,
- Empirically well-behaved.

Main limits

- No necessary conditions (apart for SL),
- Limited to quadruplets,
- Noise only in the similarities.

Conclusion

Comparison-Based Hierarchical Clustering

- Single linkage and complete linkage are inherently comparison-based,
- Several linkage functions for comparison-based average linkage,
- Recovery guarantees for a planted hierarchical model,
- Empirically well-behaved.

Main limits

- No necessary conditions (apart for SL),
- Limited to quadruplets,
- Noise only in the similarities.

ComparisonHC on **GitHub**:
`https://github.com/mperrot/ComparisonHC`

Focus: Classification

Joint work with **Ulrike von Luxburg**.
Distinguished Paper Award at IJCAI 2019.

Comparison-Based Classification

Objective: Comparison-Based Classification

- Boosting algorithm using comparisons
- Theoretical guarantees (generalization, number of triplets)

Comparison-Based Classification

Objective: Comparison-Based Classification

- Boosting algorithm using comparisons
- Theoretical guarantees (generalization, number of triplets)

Assumptions

- $(\mathcal{X}, w)$ a general metric space, $\mathcal{Y}$ a finite label space,
- $S = \{(x_i, y_i)\}_{i=1}^{N}$ a set of N examples,
- $T = \{(x_i, x_j, x_k) : w_{ij} > w_{ik}, x_j \neq x_k\}$ a set of m (noisy) triplets.

Comparison-Based Classification

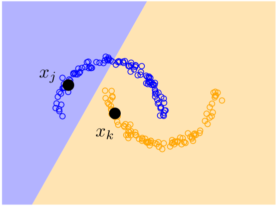
Objective: Comparison-Based Classification

- Boosting algorithm using comparisons
- Theoretical guarantees (generalization, number of triplets)

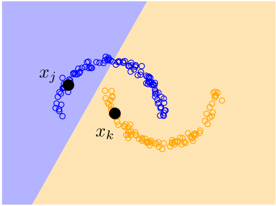
Assumptions

- $(\mathcal{X}, w)$ a general metric space, $\mathcal{Y}$ a finite label space,
- $S = \{(x_i, y_i)\}_{i=1}^{N}$ a set of N examples,
- $T = \{(x_i, x_k, x_j) : w_{ij} > w_{ik}, x_j \neq x_k\}$ a set of m (noisy) triplets.

Triplet classifier



Triplet classifier

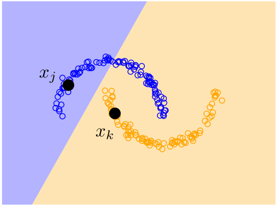


Definition

Let o_j and o_k be sets of labels and ϑ represent abstention,

$$h_{j,k}(x) = \begin{cases} o_j & \text{if } (x, x_j, x_k) \in T, \\ o_k & \text{if } (x, x_k, x_j) \in T, \\ \vartheta & \text{otherwise.} \end{cases}$$

Triplet classifier



Definition

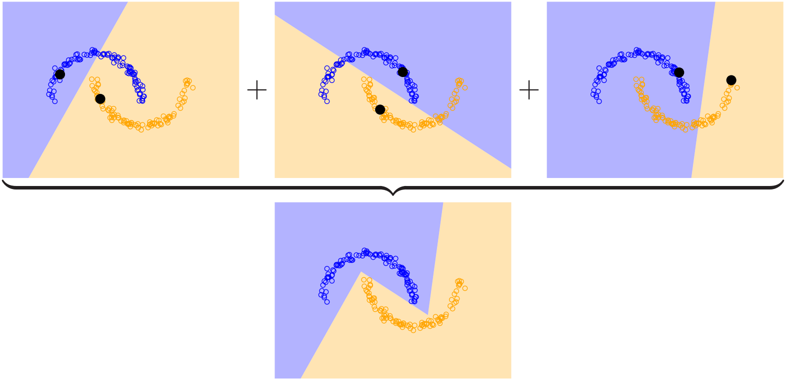
Let o_j and o_k be sets of labels and ϑ represent abstention,

$$h_{j,k}(x) = \begin{cases} o_j & \text{if } (x, x_j, x_k) \in T, \\ o_k & \text{if } (x, x_k, x_j) \in T, \\ \vartheta & \text{otherwise.} \end{cases}$$

Key property for boosting: Individual triplet classifiers are **slightly better than random classifiers**.

TripletBoost

Idea: Combine individual triplet classifiers to obtain a strong classifier.



Strong classifier:

$$H(x) = \arg\max_{y \in \mathcal{Y}} \left(\sum_{c=1}^{C} \alpha_c \mathbb{I}_{h_c(x) \neq \vartheta \wedge y \in h_c(x)} \right)$$

Theoretical guarantees

Strong classifier: $$H(x) = \arg\max_{y \in \mathcal{Y}} \left(\sum_{c=1}^{C} \alpha_c \mathbb{I}_{h_c(x) \neq \vartheta \wedge y \in h_c(x)} \right)$$

Boosting based guarantees (upper bounds)

- **Reduction of the training error** at each meaningful iteration,
- **Generalization guarantees** based on the margin theory, error drops in $\mathcal{O}\left(\sqrt{\frac{\log N}{N\theta^2}}\right)$.

Theoretical guarantees

Strong classifier: $$H(x) = \underset{y \in \mathcal{Y}}{\arg\max} \left(\sum_{c=1}^{C} \alpha_c \mathbb{I}_{h_c(x) \neq \vartheta \wedge y \in h_c(x)} \right)$$

Boosting based guarantees (upper bounds)

- **Reduction of the training error** at each meaningful iteration,
- **Generalization guarantees** based on the margin theory, error drops in $\mathcal{O}\left(\sqrt{\frac{\log N}{N\theta^2}}\right)$.

Triplets based guarantee (lower bound)

- At least $\Omega\left(N\sqrt{N}\right)$ **passively obtained triplets** are necessary to avoid random guessing.

Experiments: MovieLens

MovieLens dataset [Harper and Konstan, 2016]:

- 6040 users,
- 3706 movies,
- 1 million ratings,
- Movies have 1 or several genres (18 in total).

Experiments: MovieLens

MovieLens dataset [Harper and Konstan, 2016]:

- 6040 users,
- 3706 movies,
- 1 million ratings,
- Movies have 1 or several genres (18 in total).

Goal: classification with respect to the genres

- Use the **ratings** to generate triplets,
- Use **TripletBoost** to learn a classifier for the genres,
- Predict the genre of **new movies**.

Experiments: MovieLens

Movie		Genres
They Made Me a Criminal (1939)	True	Crime, Drama
	Pred	Drama, Comedy, Thriller, Romance, Crime
The Man Who Knew Too Little (1997)	True	Comedy, Mystery
	Pred	Comedy, Romance, Mystery, War, Crime
Heaven and Earth (1993)	True	Action, Drama, War
	Pred	Drama, Romance, Thriller, War, Crime
Planet of the Apes (1968)	True	Action, Sci-Fi
	Pred	Sci-Fi, Action, War, Adventure, Comedy
Fire Down Below (1997)	True	Action, Drama, Thriller
	Pred	Action, Thriller, Adventure, Drama, Mystery

Precision@1: ∼83.1%, Recall@5: ∼92.9%

Conclusion

TripletBoost

- A new comparison-based algorithm for classification,
- Works with general metric spaces,
- Uses passively obtained noisy triplets,
- Theoretical guarantees (generalization, number of triplets),
- Behaves well empirically (MovieLens dataset).

Conclusion

TripletBoost

- A new comparison-based algorithm for classification,
- Works with general metric spaces,
- Uses passively obtained noisy triplets,
- Theoretical guarantees (generalization, number of triplets),
- Behaves well empirically (MovieLens dataset).

Limits

- No matching upper bound for the number of necessary of triplets,
- Needs sufficiently many triplets to work well in pratice.

Conclusion

TripletBoost

- A new comparison-based algorithm for classification,
- Works with general metric spaces,
- Uses passively obtained noisy triplets,
- Theoretical guarantees (generalization, number of triplets),
- Behaves well empirically (MovieLens dataset).

Limits

- No matching upper bound for the number of necessary of triplets,
- Needs sufficiently many triplets to work well in pratice.

TripletBoost on **GitHub**:
`https://github.com/mperrot/TripletBoost`

References I

S. Dasgupta. A cost function for similarity-based hierarchical clustering. *arXiv preprint arXiv:1510.05043*, 2015.

F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19, 2016.

L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2 (1):193–218, 1985.

L. Jain, K. G. Jamieson, and R. Nowak. Finite sample prediction and recovery bounds for ordinal embedding. In *Neural Information Processing Systems*, pages 2711–2719, 2016.

M. Kleindessner and U. von Luxburg. Lens depth function and k-relative neighborhood graph: versatile tools for ordinal data analysis. *The Journal of Machine Learning Research*, 18(1):1889–1940, 2017.

L. Van Der Maaten and K. Weinberger. Stochastic triplet embedding. In *Machine Learning for Signal Processing*, pages 1–6, 2012.

TripletBoost: Algorithm

Algorithm 1 TripletBoost: boosting with triplet classifiers

Input: $S = \{(x_i, y_i)\}_{i=1}^{N}$ a set of N examples, $T = \{(x_i, x_j, x_k) : x_j \neq x_k\}$ a set of m triplets.

Output: $H(\cdot)$ a strong classifier.

1: Let $\mathcal{W}_1$ be the empirical uniform distribution: $\forall (x_i, y_i) \in S, \forall y \in \mathcal{Y}, w_{1,x_i,y} = \frac{1}{N|\mathcal{Y}|}$.
2: **for** $c = 1, \ldots, C$ **do**
3: Choose a triplet classifier h_c according to $\mathcal{W}_c$.
4: Compute the weight α_c of h_c according to its performance on $\mathcal{W}_c$.
5: Update the weights of the examples to obtain a new distribution $\mathcal{W}_{c+1}$.
6: **end for**
7: **return** $H(\cdot) = \arg\max_{y \in \mathcal{Y}} \left(\sum_{c=1}^{C} \alpha_c \mathbb{I}_{h_c(\cdot) \neq \vartheta \wedge y \in h_c(\cdot)} \right)$

Experiments: MovieLens

- Let $r_{u,i}$ be the rating of user u on movie m_i,
- Let $r_{u,i,j} = |r_{u,i} - r_{u,j}|$,
- Let $U_{i,j,k}$ be the set of users that rated all three movies.

$$T = \left\{ (m_i, m_j, m_k) : \left(\sum_{u \in U_{i,j,k}} \frac{\mathbb{I}_{r_{u,i,j} < r_{u,i,k}} - \mathbb{I}_{r_{u,i,j} > r_{u,i,k}}}{|U_{i,j,k}|} \right) > 0 \right\}$$

- Each user has only rated a small number of movies,
- Each user might give a high, respectively low, rating to a movie with a genre that he usually rates lower, respectively higher.

We only have access to a **noisy subset** of all the possible triplets.