The hard problem of ranking

SéPag seminar - 22nd March 2023

Speaker: Adrien Pavão (adrien.pavao@gmail.com)

- Introduction
- Ranking functions
- The hard problem of ranking
- What to do?
- Conclusion

The problem of ranking

	Judge 1	Judge 2	Judge 3	Judge 4
Candidate 1	2	1	2	2
Candidate 2	1	2	1	3
Candidate 3	3	3	3	1

?
?
?

The problem of ranking

	Judge 1	Judge 2	Judge 3	Judge 4
Candidate 1	2	1	2	2
Candidate 2	1	2	1	3
Candidate 3	3	3	3	1

Ranking method

?
?
?

The problem of ranking

	Judge 1	Judge 2	Judge 3	Judge 4
Candidate 1	3.5	8	8	5
Candidate 2	10	6.5	10	3.5
Candidate 3	2	1	0	7

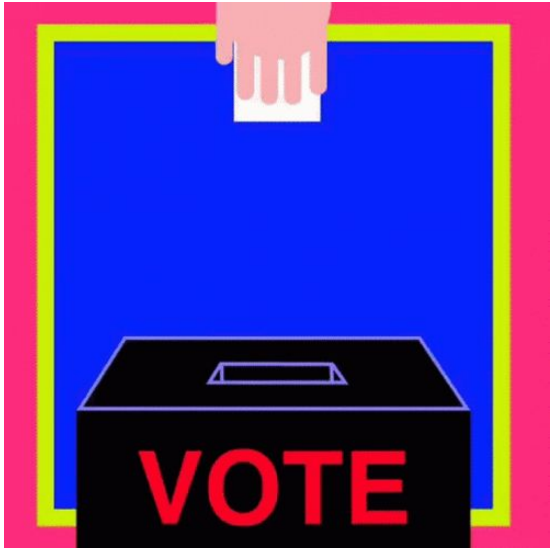
Ranking method

?
?
?

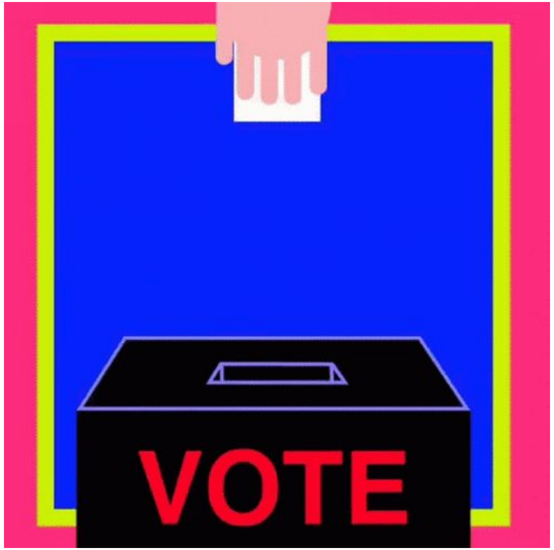
Remark: here the judges provide rankings but they could provide scores

Real world examples

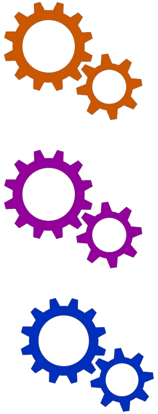
Real world examples



Real world examples

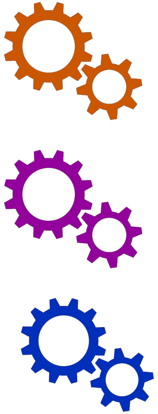


In machine learning



Models **Scores**

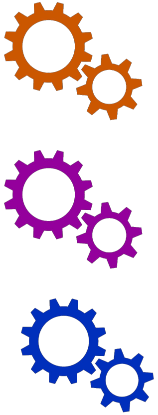
In machine learning



Task 1	Task 2	Task 3	Task 4	Task 5

Models **Scores**

In machine learning

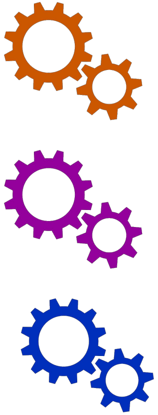


ACC	ROC AUC	F1 score	RMSE	Log Loss
Task 1	Task 2	Task 3	Task 4	Task 5

Models **Scores**

In machine learning

Fold 1	Fold 2	Fold 3	Fold 4	…
ACC	ROC AUC	F1 score	RMSE	Log Loss
Task 1	Task 2	Task 3	Task 4	Task 5

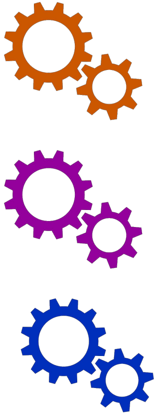


Models

Scores

In machine learning

Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
Fold 1	Fold 2	Fold 3	Fold 4	…
ACC	ROC AUC	F1 score	RMSE	Log Loss
Task 1	Task 2	Task 3	Task 4	Task 5

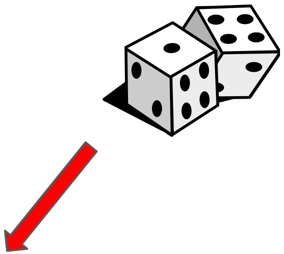


Models

Scores

- Introduction
- **Ranking functions**
- The hard problem of ranking
- What to do?
- Conclusion

Random Dictator



	j_1	j_2	j_3	j_4
c_1	0.4	0.8	0.2	0.2
c_2	0.8	0.7	0.9	0.7
c_3	0.7	0.7	0.8	1.0

M

$\Rightarrow$

0.8
0.7
0.7

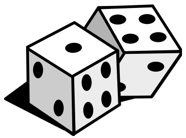
f(M)

$\Rightarrow$

1
2.5
2.5

rank(f(M))

No re-run?



Random Dictator

Only one task?

No bootstraps / cross-validation?

Ranking using **only one score ⇒ Random Dictator!** *(in many cases)*

Mean

	j_1	j_2	j_3	j_4
c_1	0.4	0.8	0.2	0.2
c_2	0.8	0.7	0.9	0.7
c_3	0.7	0.7	0.8	1.0

M

$\Rightarrow$

0.4
0.775
0.8

f(M)

$\Rightarrow$

3
2
1

rank(f(M))

Median

	j_1	j_2	j_3	j_4
c_1	0.4	0.8	0.2	0.2
c_2	0.8	0.7	0.9	0.7
c_3	0.7	0.7	0.8	1.0

M

0.3
0.75
0.75

f(M)

3
1.5
1.5

rank(f(M))

Average rank

Studied in previous works [3]

$$f(M) = \frac{1}{m} \sum_{\mathbf{j} \in \mathcal{J}} \mathtt{rank}(\mathbf{j})$$

	j_1	j_2	j_3	j_4
c_1	0.4	0.8	0.2	0.2
c_2	0.8	0.7	0.9	0.7
c_3	0.7	0.7	0.8	1.0

M

⇒

2.5
1.625
1.875

f(M)

⇒

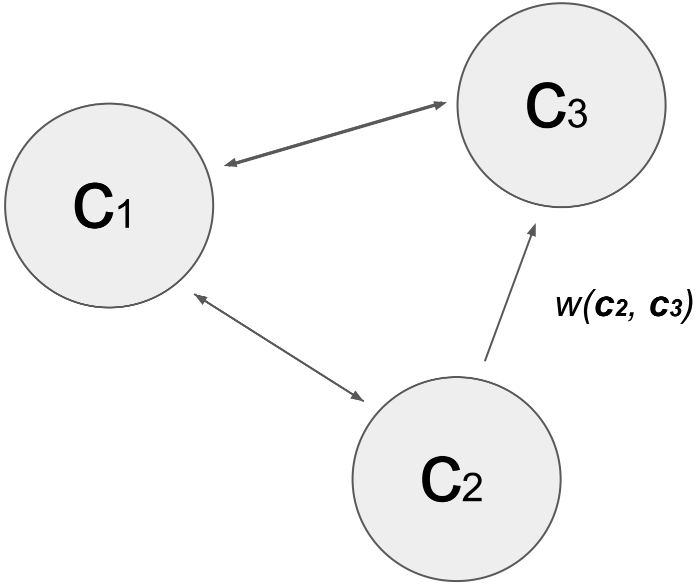
3
1
2

rank(f(M))

Ranking functions

Pairwise comparisons

$$f(M) = \left(\frac{1}{(n-1)} \sum_{j \neq i} w(\mathbf{c_i}, \mathbf{c_j}) \right)_{1 \leq i \leq n}$$



Ranking functions

Pairwise comparisons

$$f(M) = \left(\frac{1}{(n-1)} \sum_{j \neq i} w(\mathbf{c_i}, \mathbf{c_j}) \right)_{1 \leq i \leq n}$$

Copeland's method

$$w(\mathbf{u}, \mathbf{v}) = \begin{cases} \textbf{1} \text{ if the candidate } \textbf{u} \text{ is more frequently better than the candidate } \textbf{v} \text{ across all judges,} \\ \textbf{0.5} \text{ in case of a tie,} \\ \textbf{0} \text{ otherwise.} \end{cases}$$

	j_1	j_2	j_3	j_4
c_1	0.4	0.8	0.2	0.2
c_2	0.8	0.7	0.9	0.7
c_3	0.7	0.7	0.8	1.0

M

$\Rightarrow$

0.0
1.0
0.5

f(M)

$\Rightarrow$

3
1
2

rank(f(M))

Ranking functions

Pairwise comparisons

$$f(M) = \left(\frac{1}{(n-1)} \sum_{j \neq i} w(\mathbf{c_i}, \mathbf{c_j}) \right)_{1 \leq i \leq n}$$

Copeland's method

$$w(\mathbf{u}, \mathbf{v}) = \begin{cases} \textbf{\textit{1}} \textit{ if the candidate } \textbf{\textit{u}} \textit{ is more frequently better than the candidate } \textbf{\textit{v}} \textit{ across all judges,} \\ \textbf{\textit{0.5}} \textit{ in case of a tie,} \\ \textbf{\textit{0}} \textit{ otherwise.} \end{cases}$$

C_2 beats all other candidates

	j_1	j_2	j_3	j_4
c_1	0.4	0.8	0.2	0.2
c_2	0.8	0.7	0.9	0.7
c_3	0.7	0.7	0.8	1.0

M

→

f(M)
0.0
1.0
0.5

f(M)

C_2 is the "Condorcet winner"

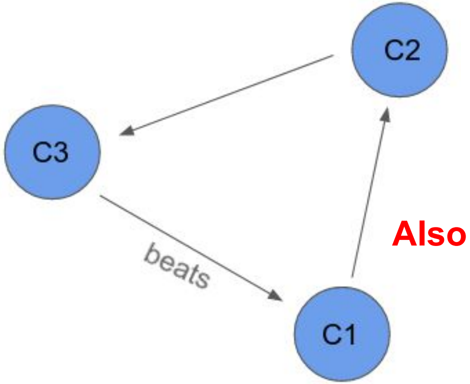
→

rank(f(M))
3
1
2

rank(f(M))

Ranking functions

Pairwise comparisons



A cycle!

Also called "Condorcet paradox" [2]

Copeland's method

$$w(\mathbf{u},\mathbf{v}) = \begin{cases} \textbf{1} \text{ if the candidate } \textbf{u} \text{ is more frequently better than the candidate } \textbf{v} \text{ across all judges,} \\ \textbf{0.5} \text{ in case of a tie,} \\ \textbf{0} \text{ otherwise.} \end{cases}$$

	j_1	j_2	j_3	j_4
c_1	0.8	0.5	0.7	0.5
c_2	0.6	0.9	0.4	0.5
c_3	0.4	0.7	0.8	0.5

M

$\Rightarrow$

1
1
1

f(M)

$\Rightarrow$

2
2
2

rank(f(M))

Ranking functions

Pairwise comparisons

$$f(M) = \left(\frac{1}{(n-1)} \sum_{j \neq i} w(\mathbf{c_i}, \mathbf{c_j}) \right)_{1 \leq i \leq n}$$

Success rate

$$w(\mathbf{u}, \mathbf{v}) = \frac{1}{m} \sum_{k=1}^{m} \mathbb{1}_{u_k > v_k}$$

	j_1	j_2	j_3	j_4
c_1	0.4	0.8	0.2	0.2
c_2	0.8	0.7	0.9	0.7
c_3	0.7	0.7	0.8	1.0

M

⇒

0.25
0.625
0.5

f(M)

⇒

3
1
2

rank(f(M))

Ranking functions

Pairwise comparisons

$$f(M) = \left(\frac{1}{(n-1)} \sum_{j \neq i} w(\mathbf{c_i}, \mathbf{c_j}) \right)_{1 \leq i \leq n}$$

Relative difference

$$w(\mathbf{u}, \mathbf{v}) = \frac{1}{m} \sum_{k=1}^{m} \frac{u_k - v_k}{u_k + v_k}$$

	j_1	j_2	j_3	j_4
c_1	0.4	0.8	0.2	0.2
c_2	0.8	0.7	0.9	0.7
c_3	0.7	0.7	0.8	1.0

M

$\Rightarrow$

-0.1553...
0.1381...
0.0172...

f(M)

$\Rightarrow$

3
2
1

rank(f(M))

Using optimization

$$\mathbf{s}^* = argmin_{\mathbf{s}} \sum_{i=1}^{m} \rho(\mathbf{j}_i, \mathbf{s})$$

Using optimization

$$\mathbf{s}^* = argmin_{\mathbf{s}} \sum_{i=1}^{m} \rho(\mathbf{j}_i, \mathbf{s})$$

Any metric…

Spearman correlation
(=> average rank [9])

Kendall tau

- Introduction
- Ranking functions
- **The hard problem of ranking**
- What to do?
- Conclusion

Theoretical criteria

Sum up

- Majority criterion
- Condorcet criterion

Winner

- Consistency
- Participation criterion

Judge perturbation

- Independence of irrelevant alternatives (IIA)
- Local IIA
- Clone-proof

Candidate perturbation

… and more

Theoretical criteria

To characterize the **behavior of the ranking functions**

Example 1: **participation criterion**

	j_1	j_2	j_3	j_4	j_5
c_1	0.9	0.9	0.5	0.5	0.2
c_2	0.7	0.7	0.2	0.2	0.7

M

$\Rightarrow$

0.5
0.7

median(M)

$\Rightarrow$

2
1

rank(median(M))

Theoretical criteria

To characterize the **behavior of the ranking functions**

Example 1: **participation criterion**

$$\begin{array}{ccc} & j_1 \quad j_2 & \begin{array}{ccc} j_3 & j_4 & j_5 \end{array} \\ c_1 & & \begin{array}{|c|c|c|} \hline 0.5 & 0.5 & 0.2 \\ \hline \end{array} \\ c_2 & & \begin{array}{|c|c|c|} 0.2 & 0.2 & 0.7 \\ \hline \end{array} \end{array} \Rightarrow \begin{array}{|c|} \hline 0.5 \\ \hline 0.2 \\ \hline \end{array} \Rightarrow \begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline \end{array} \; !$$

M $\qquad$ *median(M)* $\qquad$ *rank(median(M))*

Median does NOT satisfies the participation criterion, while most methods do.

Theoretical criteria

To characterize the **behavior of the ranking functions**

Example 2: **independence of irrelevant alternatives (IIA) criterion**

	j_1	j_2	j_3
c_1	0.6	0.6	0
c_2	0.4	0.4	1
c_3	1	0	0.4

M

2
2
2

AvRank(M)

2
2
2

rank(AvRank(M))

Theoretical criteria

To characterize the **behavior of the ranking functions**

Example 2: **independence of irrelevant alternatives (IIA) criterion**

	j_1	j_2	j_3
c_1	0.6	0.6	0
c_2	0.4	0.4	1
c_3			

M

$\Rightarrow$

1.333…
1.667…

AvRank(M)

$\Rightarrow$

1
2

!

rank(AvRank(M))

Average rank does NOT satisfies the IIA criterion, while median does.

No method is perfect

No method is perfect

Gibbard's theorem* [3]:

Any deterministic ranking method holds **at least one** of the following three *(unwanted)* properties:

1. The process is **dictatorial**
2. The ranking is **limited to only two candidates**
3. The process is open to **"tactical voting"**: the preferences of a judge may not best defend their interest.

**Generalization of Arrow's theorem [4]*

No method is perfect

Gibbard's theorem* [3]:

Any deterministic ranking method holds **at least one** of the following three *(unwanted)* properties:

1. The process is **dictatorial**
2. The ranking is **limited to only two candidates**
3. The process is open to **"tactical voting"**: the preferences of a judge may not best defend their interest

In practice, this imply incompatibilities between the desired properties of ranking functions

https://en.wikipedia.org/wiki/Comparison_of_electoral_systems

**Generalization of Arrow's theorem [4]*

Sum up

	Majority	Condorcet	Consistency	Participation	IIA	LLIA	Clone proof
Random Dictator 			✓	✓	✓	✓	✓
Mean			✓	✓	✓	✓	✓
Median					✓	✓	✓
Average Rank			✓	✓			
Copeland's method	✓	✓					
Success Rate			✓	✓			
Relative Difference			✓	✓			
Kemeny-Young	✓	✓				✓	

- Introduction
- Ranking functions
- The hard problem of ranking
- **What to do?**
- Conclusion

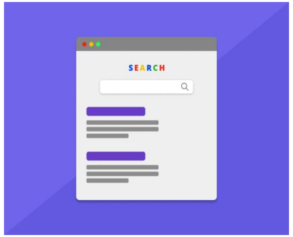
What to do?

In practice, the choice of the ranking functions may depend on the problem

Multiple tasks/datasets

Cross-validation

Multiple samples



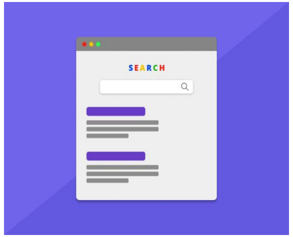
What to do?

In practice, the choice of the ranking functions may depend on the problem

Multiple tasks/datasets

Cross-validation

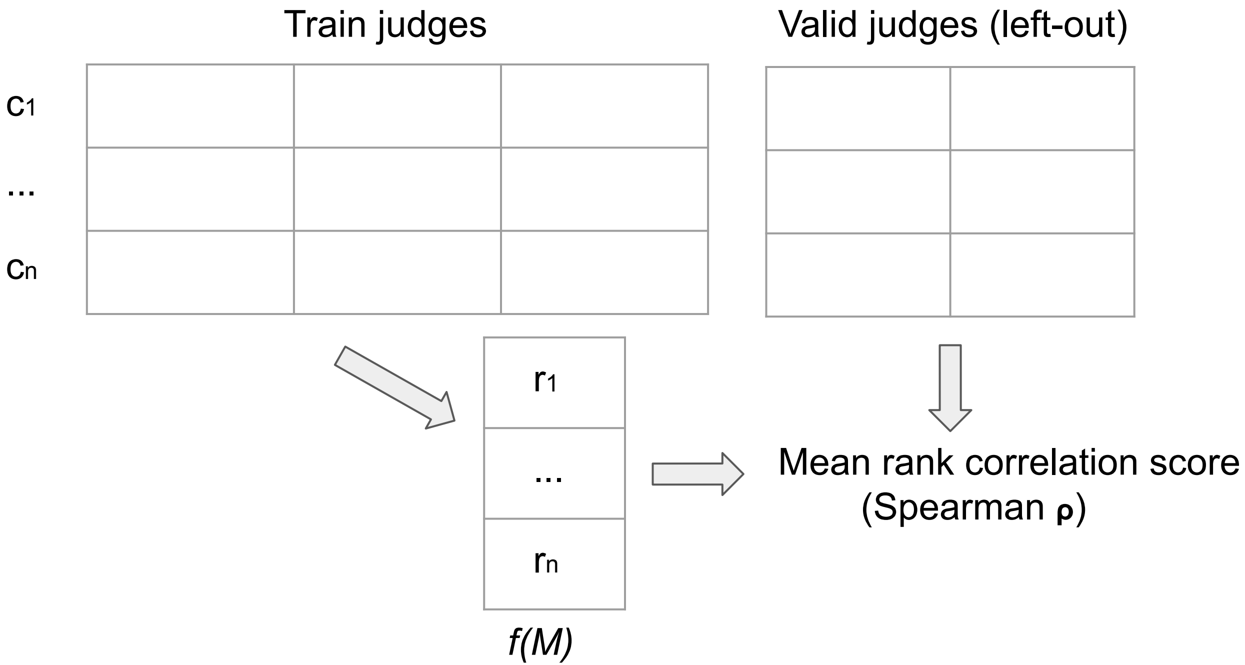
Multiple samples



Let’s try to find out **empirically**

Empirical criteria

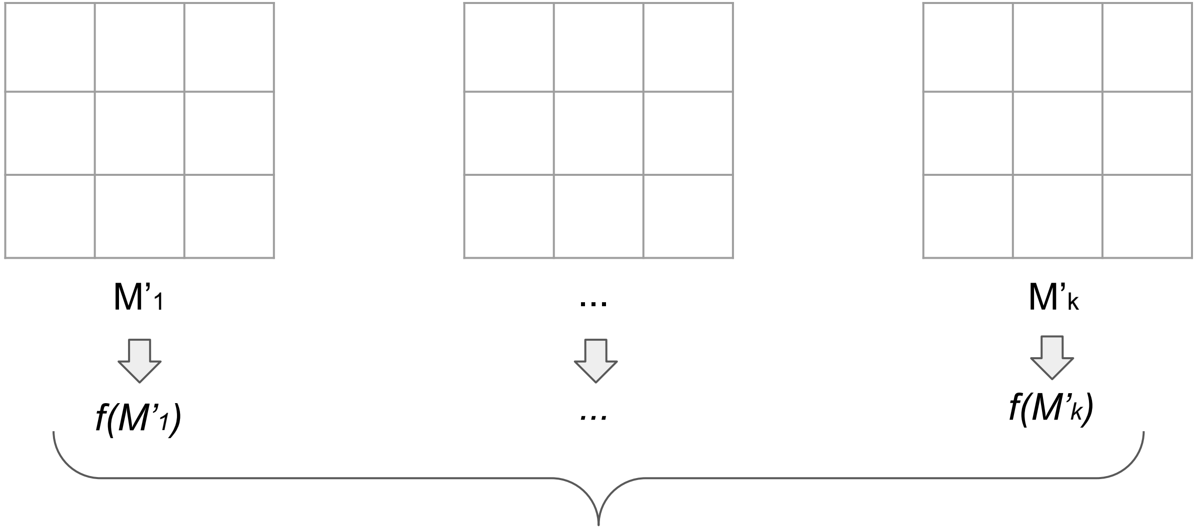
$$\text{generalization}(f) = \sum_{\mathbf{j} \in \mathcal{J}^{valid}} \frac{1}{m} \sigma(f(\mathcal{J}^{train}), \ \mathtt{rank}(\mathbf{j}))$$



Empirical criteria

$$\texttt{stability}(f) = \frac{1}{m(m-1)} \sum_{i \neq j} \sigma(X_i, X_j)$$

Where X is a matrix whose columns are the rankings f(M’) produced on several variation M’ of the score matrix M.
Variation can be on candidates, judges, or both.



Mean correlation between all ranking produced
(Spearman **ρ**)

Empirical criteria

Criteria relative to the elected winner (ranked first)

- **The average rank of the winner** is the average rank across all input judges of the candidate ranked first in *f(M)*.

- **The Condorcet rate** is the rate of ranking the Condorcet winner first when one exists.

  Remark: This rate need to be evaluated on a set of score matrices

Experimental setting (case judges = datasets, inline with [1])

Globally normalized?

Benchmarks

	# Datasets	# Algorithms	Metric	W	Norm	Source
AutoDL-AUC	66	13	AUC	0.38	No	AutoDL [9]
AutoDL-ALC	66	13	ALC	0.60	No	
AutoML	30	17	BAC or R^2	0.27	Yes	AutoML [6]
Artificial	50	20	*None*	0.00	Yes	Authors of [13]
OpenML	76	292	Accuracy	0.32	Yes	Alors [10] website
Statlog	22	24	Error rate	0.27	Yes	Statlog in UCI repository

10,000 repeat trials based on bootstraps

Concordance between judges

Experimental setting

Globally normalized?

Benchmarks

	# Datasets	# Algorithms	Metric	W	Norm	Source
AutoDL-AUC	66	13	AUC	0.38	No	AutoDL [5]
AutoDL-ALC	66	13	ALC	0.60	No	
AutoML	30	17	BAC or R^2	0.27	Yes	AutoML [6]
Artificial	50	20	*None*	0.00	Yes	Authors of [7]
OpenML	76	292	Accuracy	0.32	Yes	Alors [8] website
Statlog	22	24	Error rate	0.27	Yes	Statlog in UCI repository

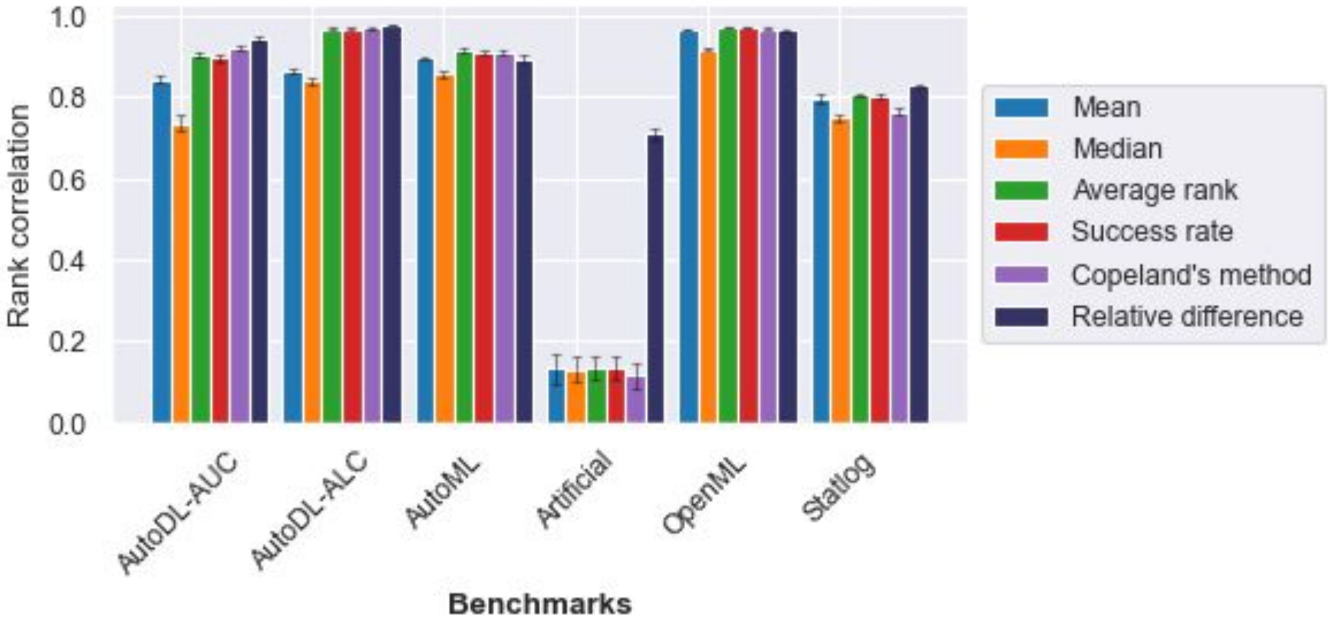
10,000 repeat trials based on bootstraps

Concordance between judges

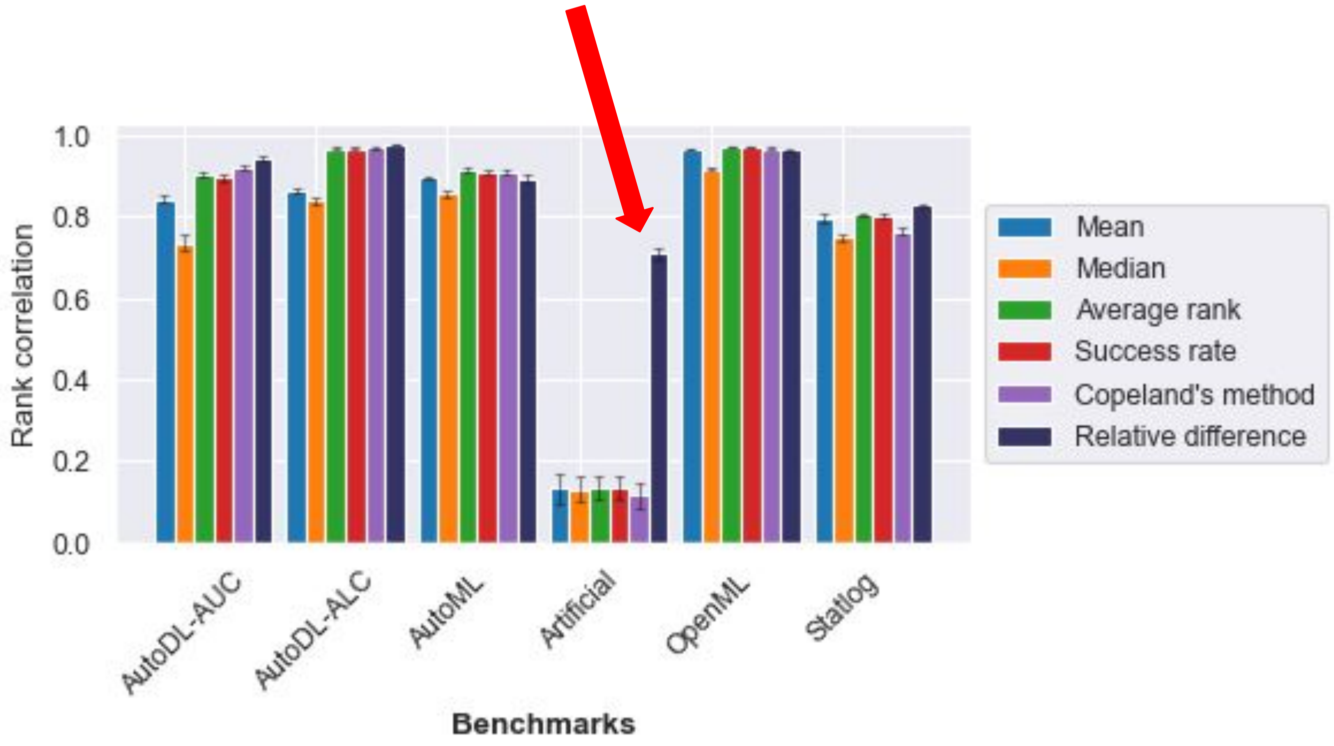
Experimental results (case judges = datasets, inline with [1])

	Theoretical properties							Empirical properties				
	Winner		Judge		Candidate			Winner		Judge		Candidate
	Maj.	Condorcet	Consist.	Particip.	IIA	LIIA	Clone-proof	Winner rank	Condorcet rate	Generalization	Stability (judge)	Stability (candidate)
Mean	0	0	1	1	1	1	1	0.68	0.4	0.36	0.753	1.000
Median	0	0	0	0	1	1	1	0.70	0.5	0.37	0.702	1.000
Average rank	0	0	1	1	0	0	0	0.74	0.8	0.41	0.780	0.954
Success rate	0	0	1	1	0	0	0	0.73	0.8	0.40	0.777	0.839
Relative diff.	0	0	1	1	0	0	0	0.73	0.8	0.41	0.884	0.941
Copeland	1	1	0	0	0	0	0	0.73	1.0	0.41	0.771	0.965

Experimental results - “Judge stability”



Experimental results - “Judge stability”



Discussion time! 🤔

We are trying to rank the ranking functions…

…how do we solve this meta-problem?

- Introduction
- Ranking functions
- The hard problem of ranking
- What to do?
- **Conclusion**

Conclusion

The problem of ranking candidates from multiple scores is **hard**

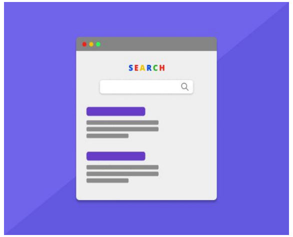
Need empirical studies

In practice, the choice of the ranking functions may depend on the problem

Multiple tasks/datasets

Cross-validation

Multiple samples



Thank you!

- Any question?

- Feel free to reach me later, I’ll be happy to discuss this topic with you!

References

[1] Pavel B. Brazdil and Carlos Soares, 2000. “A Comparison of Ranking Methods for Classification Algorithm Selection”.

[2] William V. Gehrlein, 1997. “Condorcet's Paradox and the Condorcet Efficiency of Voting Rules”.

[3] Gibbard Allan, 1973. “Manipulation of voting schemes: A general result”. *Econometrica*.

[4] Arrow, Kenneth J.,1950. "A Difficulty in the Concept of Social Welfare". *Journal of Political Economy*.

[5] Zhengying Liu, Adrien Pavao, Zhen Xu, Sergio Escalera, Fabio Ferreira, Isabelle Guyon, Sirui Hong, Frank Hutter, Rongrong Ji, Júlio C. S. Jacques Júnior, Ge Li, Marius Lindauer, Zhipeng Luo, Meysam Madadi, Thomas Nierhoff, Kangning Niu, Chunguang Pan, Danny Stoll, Sébastien Treguer, Jin Wang, Peng Wang, Chenglin Wu, Youcheng Xiong, Arber Zela, and Yang Zhang. Winning solutions and post-challenge analyses of the chalearn autodl challenge 2019. IEEE Trans. Pattern Anal. Mach. Intell., 43(9):3108–3125, 2021.

[6] I. Guyon et al. Analysis of the AutoML Challenge Series 2015–2018, pages 177–219. Springer International Publishing, Cham, 2019.

[7] L. Sun-Hosoya, I. Guyon, and M. Sebag. Activmetal: Algorithm recommendation with active meta learning. In Wshp on Interactive Adaptive Learning @ ECML-PKDD, 2018.

[8] M. Misir and M. Sebag. Alors: An algorithm recommender system. Artif. Intell., 244:291–314, 2017.

[9] Kendall, M. G. and Gibbons, J. D. (1990) pg. 125. Rank Correlation Methods. *5th ed. London: Griffin.*