Learning MR-Sort rules with coalitional veto

Olivier Sobrie[1,2] Vincent Mousseau[1] Marc Pirlot[2]

[1] Université Paris-Saclay - CentraleSupélec
[2] Université de Mons - Faculté polytechnique

November 7, 2016

Sorting problem

Settings

- **Assignment** of alternatives in categories
- Categories are **ordered**
- Alternatives are evaluated on **monotone criteria**

Example of sorting problem

- Assignment of hotels in two categories : “Bad” and “Good”




distance to the beach	600m	300m	50m	200m	...
distance to the center	500m	100m	600m	300m	...
price	150€	130€	90€	80€	...
size	45m^2	35m^2	30m^2	25m^2	...
rating	★★★★★	★★★★	★★★	★★	...

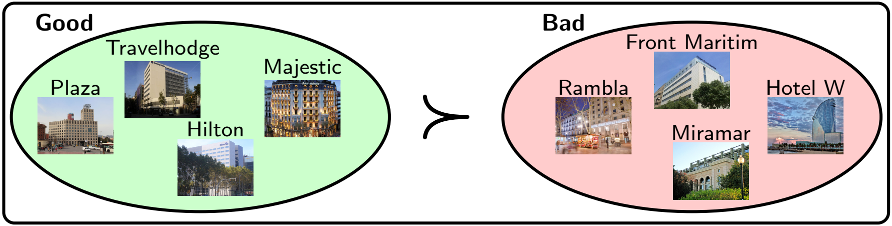
Sorting problem

Settings

- **Assignment** of alternatives in categories
- Categories are **ordered**
- Alternatives are evaluated on **monotone criteria**

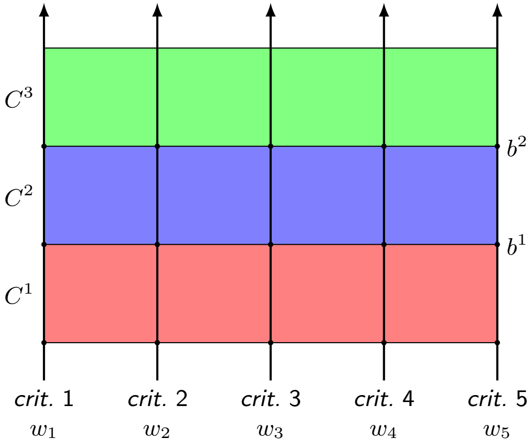
Example of sorting problem

- Assignment of hotels in two categories : "Bad" and "Good"



Majority rule sorting model

- Sorting model (p **ordered categories**, i.e. $C^p \succ C^{p-1} \succ \ldots \succ C^1$)
- Axiomatized by Bouyssou and Marchant (2007a,b)



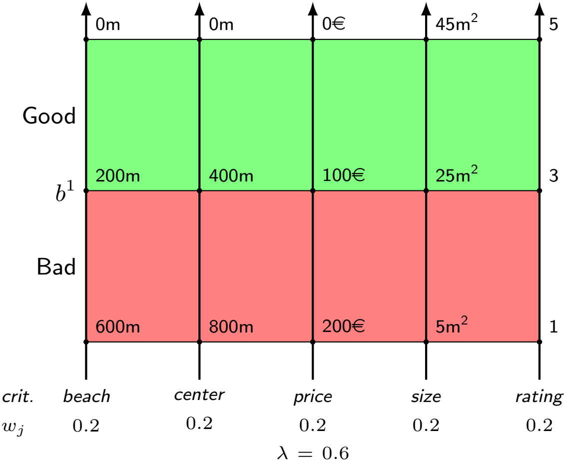
- n **weights** $(w_1, \ldots, w_n)$
- 1 **majority threshold** (λ)
- $p-1$ **profiles** $(b^1, \ldots, b^{p-1})$

Assignment rule

$$a \in C^h \Leftrightarrow \sum_{j: a_j \geq b_j^{h-1}} w_j \geq \lambda \text{ and } \sum_{j: a_j \geq b_j^{h}} w_j < \lambda$$

MR-Sort applied to the introductory example

- **Sorting** accommodations in two categories : **Good** and **Bad**

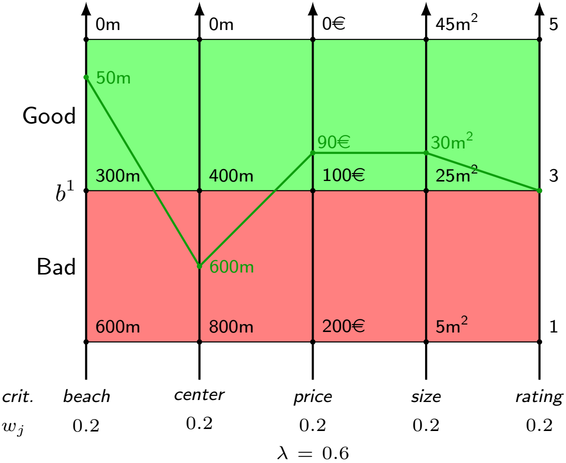


Assignment rule

$$\text{hotel} \in \text{Good} \Leftrightarrow \sum_{j:a_j \geq b_j^1} w_j \geq \lambda$$

MR-Sort applied to the introductory example

- **Sorting** accommodations in two categories : **Good** and **Bad**



Assignment rule

$$\text{hotel} \in \text{Good} \Leftrightarrow \sum_{j:a_j \geq b_j^1} w_j \geq \lambda$$

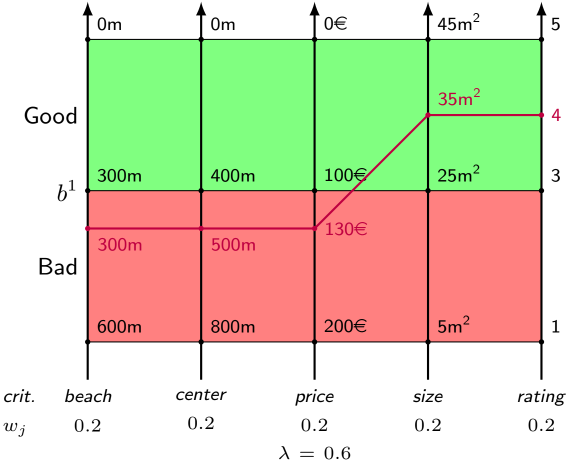
Hilton

$\in$ Good

$$\sum_{j:a_j \geq b_j^1} w_j = 0.8$$

MR-Sort applied to the introductory example

- **Sorting** accommodations in two categories : **Good** and **Bad**



Assignment rule

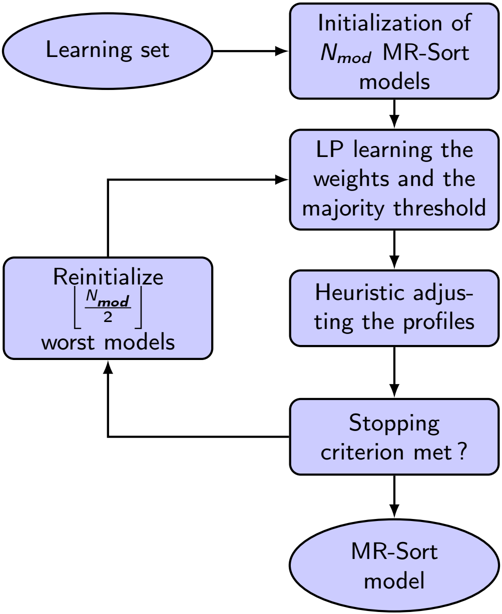
$$\text{hotel} \in \text{Good} \Leftrightarrow \sum_{j : a_j \geq b_j^1} w_j \geq \lambda$$

Plaza

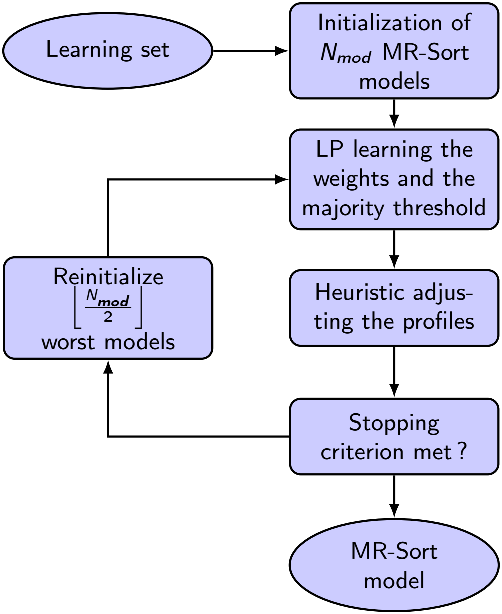
$\in$ Bad

$$\sum_{j : a_j \geq b_j^1} w_j = 0.4$$

Heuristic algorithm for learning a MR-Sort model

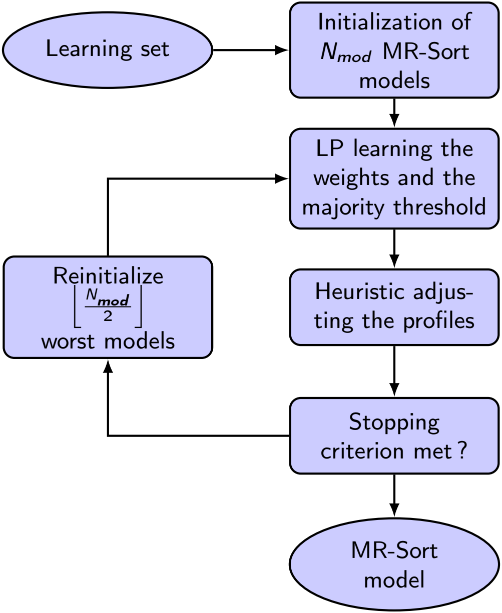


Heuristic algorithm for learning a MR-Sort model



Profiles initialized with a heuristic with some randomness

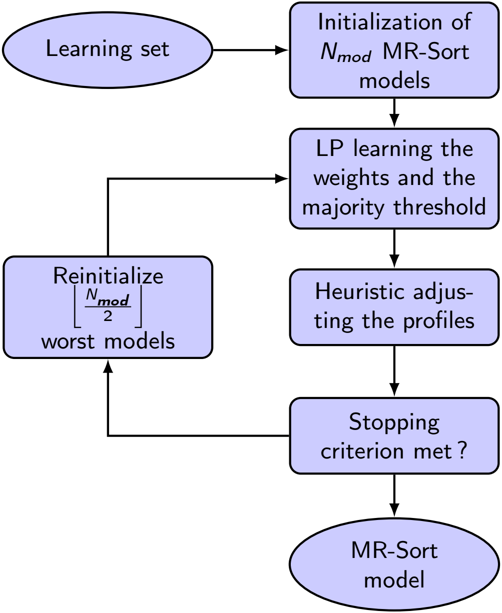
Heuristic algorithm for learning a MR-Sort model



Profiles initialized with a heuristic with some randomness

Fixed profiles
Maximization of the CA

Heuristic algorithm for learning a MR-Sort model

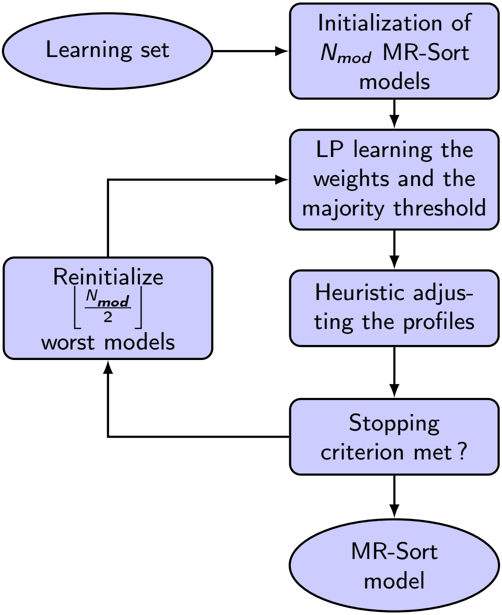


Profiles initialized with a heuristic with some randomness

Fixed profiles
Maximization of the CA

Fixed weights and majority threshold
Maximization of the CA

Heuristic algorithm for learning a MR-Sort model



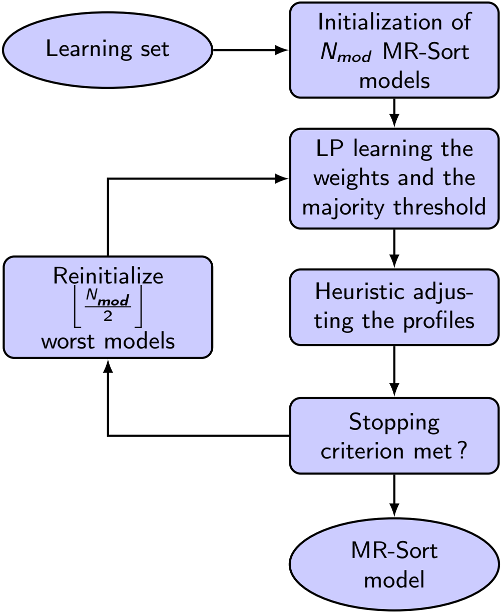
Profiles initialized with a heuristic with some randomness

Fixed profiles
Maximization of the CA

Fixed weights and majority threshold
Maximization of the CA

Once a model restores all the assignment examples or after N_{it} iterations

Heuristic algorithm for learning a MR-Sort model



Profiles initialized with a heuristic with some randomness

Fixed profiles
Maximization of the CA

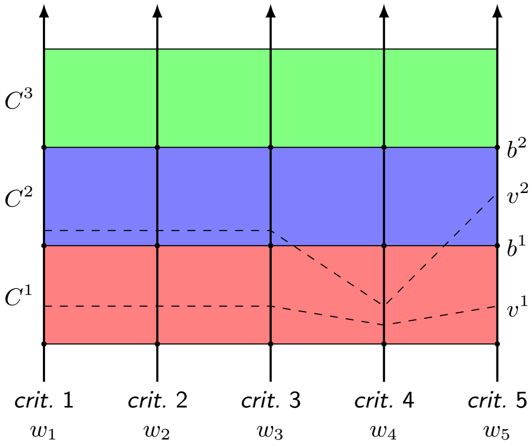
Fixed weights and majority threshold
Maximization of the CA

Once a model restores all the assignment examples
or after N_{it} iterations

The best model regarding CA or AUC is returned

MR-Sort with binary veto rule

- Sorting model (p **ordered categories**, i.e. $C^p \succ C^{p-1} \succ \ldots \succ C^1$)
- Veto if alternative worse than the **veto profile** on **any criterion**



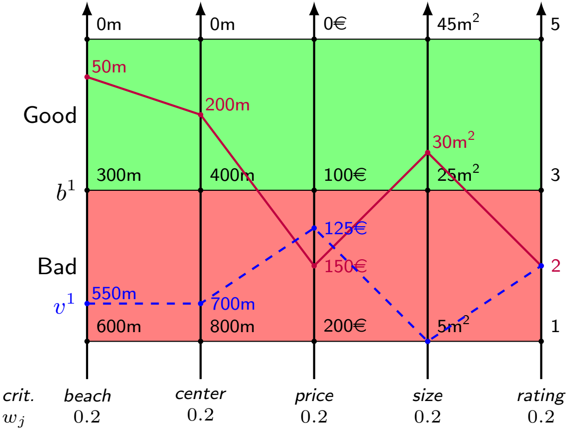
- n **weights** $(w_1, \ldots, w_n)$
- 1 **majority threshold** (λ)
- $p-1$ **profiles** $(b^1, \ldots, b^{p-1})$
- $p-1$ **veto profiles** $(v^1, \ldots, v^{p-1})$

Assignment rule

$$a \in C^h$$
$$\Leftrightarrow$$
$$\sum_{j:a_j \geq b_j^{h-1}} w_j \geq \lambda \text{ and } \nexists j : a_j \leq v_j^{h-1}$$
AND
$$\sum_{j:a_j \geq b_j^{h}} w_j < \lambda \text{ or } \exists j : a_j \leq v_j^{h}$$

MR-Sort with binary veto rule

- Veto if alternative worse than the **veto profile** on **any criterion**



$\lambda = 0.6$

Assignment rule

hotel $\in$ Good

$$\Leftrightarrow$$

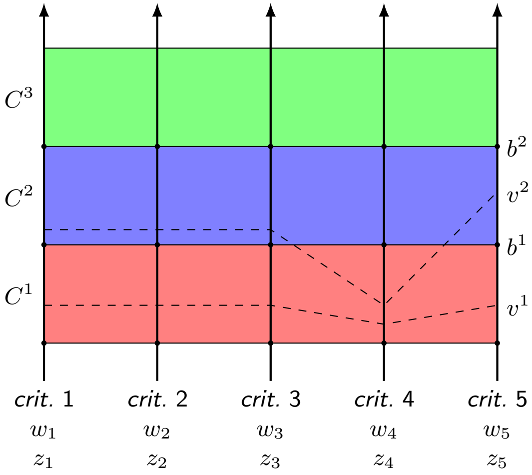
$$\sum_{j : a_j \geq b_j^1} w_j \geq \lambda \text{ and } \nexists j : a_j \leq v_j^1$$

Rambla

$\in$ **Bad**

MR-Sort with coalitional veto rule

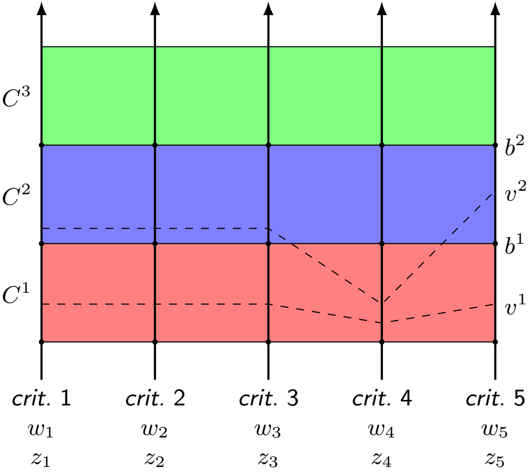
- Sorting model (p **ordered categories**, i.e. $C^p \succ C^{p-1} \succ \ldots \succ C^1$)
- Veto if alternative worse than the **veto profile** on a **subset of criteria**



- n **weights** $(w_1, \ldots, w_n)$
- n **veto weights** $(z_1, \ldots, z_n)$
- 1 **majority threshold** (λ)
- 1 **veto threshold** (Λ)
- $p-1$ **profiles** $(b^1, \ldots, b^{p-1})$
- $p-1$ **veto profiles** $(v^1, \ldots, v^{p-1})$

MR-Sort with coalitional veto rule

- Sorting model (p **ordered categories**, i.e. $C^p \succ C^{p-1} \succ \ldots \succ C^1$)
- Veto if alternative worse than the **veto profile** on a **subset of criteria**

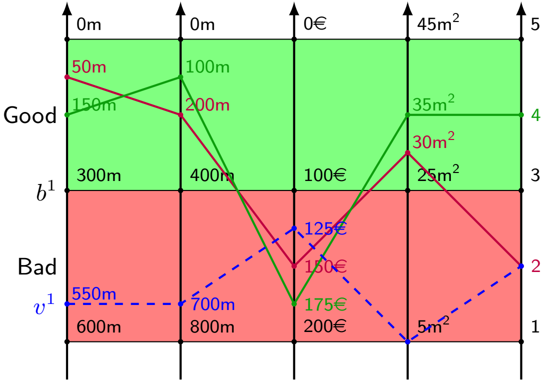


Assignment rule

$$a \in C^h$$
$$\Leftrightarrow$$
$$\sum_{j : a_j \geq b_j^{h-1}} w_j \geq \lambda \text{ and } \sum_{j : a_j \leq v_j^h} z_j < \Lambda$$
AND
$$\sum_{j : a_j \geq b_j^{h}} w_j < \lambda \text{ or } \sum_{j : a_j \leq v_j^h} z_j \geq \Lambda$$

MR-Sort with coalitional veto rule

- Veto if alternative worse than the **veto profile** on a **subset of criteria**



crit.	beach	center	price	size	rating
w_j	0.2	0.2	0.2	0.2	0.2
z_j	0.2	0.2	0.2	0.2	0.2

$\lambda = 0.6$ $\Lambda = 0.4$

Assignment rule

$$\text{hotel} \in \text{Good} \Leftrightarrow \sum_{j: a_j \geq b_j^1} w_j \geq \lambda \text{ and } \sum_{j: a_j \leq v_j^1} z_j < \Lambda$$

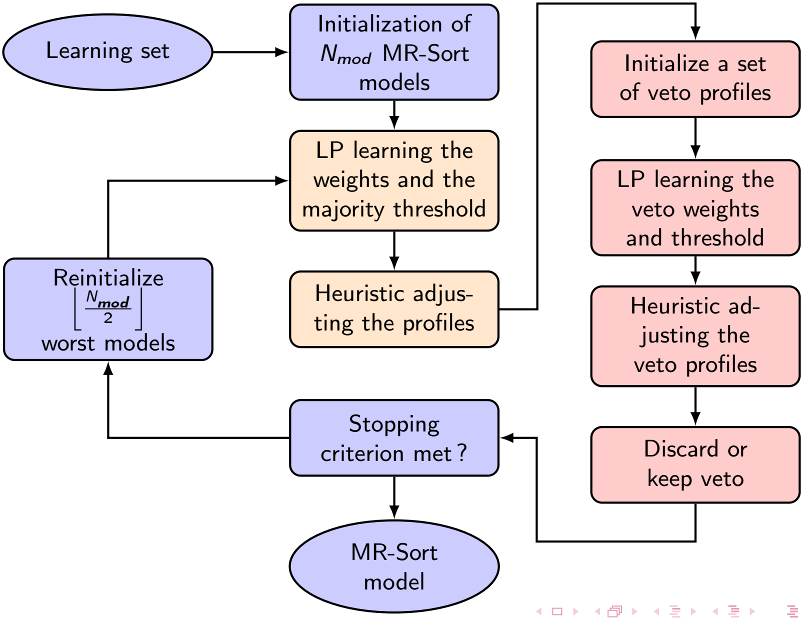
Rambla

$\in$ **Bad**

Majestic

$\in$ **Good**

Heuristic algo. for learning a MR-SortCV model



Experimental results I

- Datasets used in Tehrani et al. (2012); Sobrie et al. (2015)
- **120 to 1728 instances**
- **4 to 8 attributes**
- **2 to 36 categories**

Data set	#instances	#attributes	#categories
DBS	120	8	2
CPU	209	6	4
BCC	286	7	2
MPG	392	7	36
ESL	488	4	9
MMG	961	5	2
ERA	1000	4	4
LEV	1000	4	5
CEV	1728	6	4

Experimental results II

- Categories have been **binarized**
- Datasets split in **twofold 50/50 partition** : a learning and a test set (operation repeated 100 times)
- Average classification accuracy of the **test set** :

Data set	MR-Sort	MR-SortCV
DBS	0.8377 ± 0.0469	0.8390 ± 0.0476
CPU	0.9325 ± 0.0237	0.9429 ± 0.0244
BCC	0.7250 ± 0.0379	0.7044 ± 0.0299
MPG	0.8219 ± 0.0237	0.8240 ± 0.0391
ESL	0.8996 ± 0.0185	0.9024 ± 0.0179
MMG	0.8268 ± 0.0151	0.8267 ± 0.0119
ERA	0.7944 ± 0.0173	0.7959 ± 0.0270
LEV	0.8408 ± 0.0122	0.8551 ± 0.0171
CEV	0.8516 ± 0.0091	0.8516 ± 0.0665

- **MR-SortCV doesn't improve** the performances

Experimental results III

- Results obtained with the **original datasets**
- Datasets split in **twofold 50/50 partition** : a learning and a test set (operation repeated 100 times)
- Average classification accuracy of the **test set** :

Dataset	# cat.	MR-Sort	MR-SortCV
CPU	4	0.8039 ± 0.0354	0.8469 ± 0.0426
ERA	4	0.5123 ± 0.0233	0.5230 ± 0.0198
LEV	5	0.5662 ± 0.0258	0.5734 ± 0.0213
CEV	4	0.7664 ± 0.0193	0.7832 ± 0.0130

- **MR-SortCV performs better** with more than 2 categories

Experimental results IV

- Tests with **artificial datasets**
- **Learning set** composed of **1000 alternatives** assigned to **2 categories** by a **random generated MR-SortCV model** composed of **4 to 7** criteria
- **Test set** composed of **10000 alternatives**
- The learning set is used as input of the heuristic algorithm learning a MR-SortCV model

# criteria	Learning set	Test set
4	0.9908 ± 0.01562	0.98517 ± 0.01869
5	0.9904 ± 0.01447	0.98328 ± 0.01677
6	0.9860 ± 0.01560	0.97547 ± 0.02001
7	0.9827 ± 0.01766	0.96958 ± 0.02116

- The learned models restore on average $\sim$ 99% of the examples
- **Good performances in generalization**

Conclusion

- New and **general form** of veto condition
- **"Reversed"** MR-Sort (concordance) rule
- **No significant improvements**
- Veto adds **limited descriptive ability** to the MR-Sort model
- It confirms the results obtained by **Olteanu and Meyer (2014)**

Vielen Dank für Ihre Aufmerksamkeit

References I

Bouyssou, D. and Marchant, T. (2007a). An axiomatic approach to noncompensatory sorting methods in MCDM, I : The case of two categories. *European Journal of Operational Research*, 178(1) :217–245.

Bouyssou, D. and Marchant, T. (2007b). An axiomatic approach to noncompensatory sorting methods in MCDM, II : More than two categories. *European Journal of Operational Research*, 178(1) :246–276.

Olteanu, A. L. and Meyer, P. (2014). Inferring the parameters of a majority rule sorting model with vetoes on large datasets. In *DA2PL 2014 : From Multicriteria Decision Aid to Preference Learning*, pages 87–94.

Sobrie, O., Gillis, N., Mousseau, V., and Pirlot, M. (2015). Using polynomial marginal utility functions in UTADIS. In *27th European Conference on Operational Research*, Glasgow, Scotland.

Tehrani, A. F., Cheng, W., Dembczyński, K., and Hüllermeier, E. (2012). Learning monotone nonlinear models using the Choquet integral. *Machine Learning*, 89(1–2) :183–211.