Learning the parameters of a Non-Compensatory Sorting Model

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

[1]CentraleSupélec - Laboratoire de Génie Industriel
[2]University of Mons - Faculty of engineering

September 28, 2015

Introductory example

- Admission/Refusal of student.
- Students are evaluated in 4 courses.
- Admission condition : score above 10/20 in all the courses of one the minimal winning coalitions.

Minimal winning coalitions

- {math, physics}
- {math, chemistry}
- {chemistry, history}

Maximal loosing coalitions

- {math, history}
- {physics, chemistry}
- {physics, history}

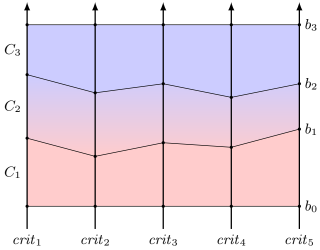
	Math	Physics	Chemistry	History	A/R
James	15	15	5	5	A
Marc	15	5	15	5	A
Robert	5	5	15	15	A
John	15	5	5	15	R
Paul	5	15	5	15	R
Pierre	5	15	15	5	R

Majority rule sorting model (MR-Sort) I

Characteristics

- Allows to sort alternatives in ordered classes on basis of their performances on monotone criteria.
- MCDA method based on outranking relations.
- Simplified version of ELECTRE TRI.

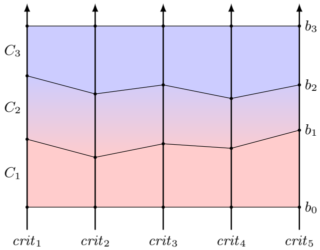
Parameters



- Profiles performances ($b_{h,j}$ for $h = 1, ..., p-1; j = 1, ..., n$).
- Criteria weights ($w_j \geq 0$ for $n = 1, ..., n$, $\sum_{j=1}^{n} w_j = 1$).
- Majority threshold (λ).

Majority rule sorting model (MR-Sort) II

Parameters



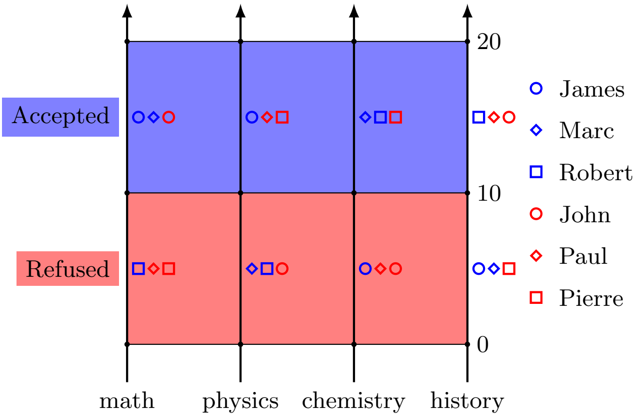
- Profiles performances ($b_{h,j}$ for $h = 1, ..., p-1; j = 1, ..., n$).
- Criteria weights ($w_j \geq 0$ for $n = 1, ..., n$, $\sum_{j=1}^{n} w_j = 1$).
- Majority threshold (λ).

Assignment rule

$$a \in C_h \iff \sum_{j: a_j \geq b_{h-1,j}} w_j \geq \lambda \text{ and } \sum_{j: a_j \geq b_{h,j}} w_j < \lambda$$

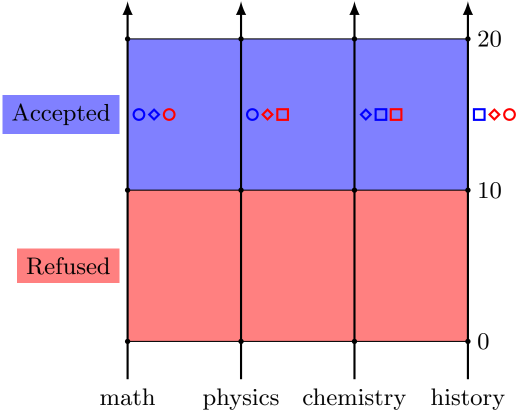
MR-Sort applied to the introductory example

- Student a accepted $\iff \sum_{j:a_j \geq 10} w_j \geq \lambda$



MR-Sort applied to the introductory example

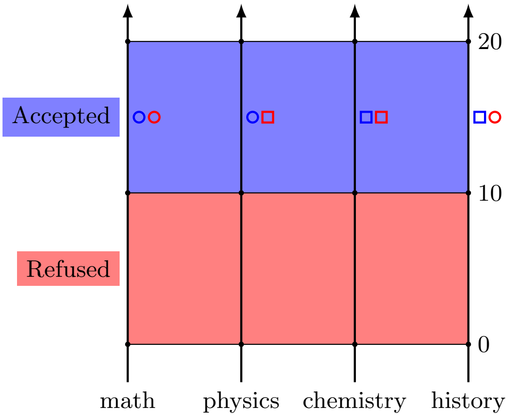
- Student a accepted $\iff \sum_{j: a_j \geq 10} w_j \geq \lambda$



○ (blue)	James	$w_{\text{math}} + w_{\text{physics}} \geq \lambda$
◇ (blue)	Marc	$w_{\text{math}} + w_{\text{chemistry}} \geq \lambda$
□ (blue)	Robert	$w_{\text{chemistry}} + w_{\text{history}} \geq \lambda$
○ (red)	John	$w_{\text{math}} + w_{\text{history}} < \lambda$
◇ (red)	Paul	$w_{\text{physics}} + w_{\text{history}} < \lambda$
□ (red)	Pierre	$w_{\text{physics}} + w_{\text{chemistry}} < \lambda$

MR-Sort applied to the introductory example

- Student a accepted $\iff \sum_{j : a_j \geq 10} w_j \geq \lambda$



- ○ James $w_{\text{math}} + w_{\text{physics}} \geq \lambda$
- □ Robert $w_{\text{chemistry}} + w_{\text{history}} \geq \lambda$
- ○ John $w_{\text{math}} + w_{\text{history}} < \lambda$
- □ Pierre $w_{\text{physics}} + w_{\text{chemistry}} < \lambda$

$\lambda \leq \frac{1}{2}$ $\lambda > \frac{1}{2}$

- Impossible to represent all the examples with MR-Sort.

Non-compensatory sorting model (NCSM)

Characteristic

- Characterized by [Bouyssou and Marchant, 2007].
- Improvement of the expressivity of the model.
- Take criteria interactions into account.

Capacity

- $F = \{1, ..., n\}$: set of criteria
- A capacity is a function $\mu : 2^F \rightarrow [0, 1]$ such that :
  - $\mu(B) \geq \mu(A)$, for all $A \subseteq B \subseteq F$ (monotonicity) ;
  - $\mu(\emptyset) = 0$ and $\mu(F) = 1$ (normalization).

New assignment rule

$$a \in C_h \iff \mu(\{j \in F : a_j \geq b_{h-1,j}\}) \geq \lambda \quad \text{and} \quad \mu(\{j \in F : a_j \geq b_{h,j}\}) < \lambda$$

Learning a NCSM model - MIP I

Mixed Integer Programming

- Input : Examples of assignments and their associated vectors of performances.
- Objective : Finding a model compatible with as much example as possible.
- MIP to learn an MR-Sort model in [Leroy et al., 2011].
- Limitation to 2-additive capacities.
- For NCSM, more constraints and binary variable are required :

Table – Max number of constraints

	MIP MR-Sort	MIP NCSM
# binary variables	$n(2m+1)$	$n(2m+1+2m(m+1))$
# constraints	$2n(5m+1)+n(p-3)+1$	$2n(5m+1)+n(p-3)+1+2m(n^2+1)+n^2$

- Too much variables and constraints to be used with large datasets.

Learning a NCSM model - MIP II

Application to the introductory example

- Admission condition : score above 10/20 in all the courses of one these coalitions :
  - {math, physics}
  - {math, chemistry}
  - {chemistry, history}
- MIP is able to find a model matching all the rules

J	$m(J)$
{math}	0
{physics}	0
{chemistry}	0
{history}	0

$\lambda = 0.3$

J	$m(J)$
{math, physics}	0.3
{math, chemistry}	0.3
{math, history}	0
{physic, chemistry}	0
{physic, history}	0
{chemistry, history}	0.4

Learning a NCSM model - Meta I

Metaheuristic to learn a NCSM model

- Input : Examples of assignments and their associated vectors of performances.
- Objective : Finding a model compatible with as much example as possible.
- Being able to handle large datasets.
- Metaheuristic to learn parameters of a MR-Sort model in [Sobrie et al., 2012, Sobrie et al., 2013].

Learning a NCSM model - Meta II

Recall : Metaheuristic to learn a MR-Sort model

- Principle (genetic algorithm) :
  - Initialize a population of MR-Sort models
  - Evolve the population by iteratively
    - Optimizing weights (profiles fixed) with a LP
    - Improving profiles (weights fixed) with a heuristic
    - Selecting the best models and reinitializing the others
  - ... to get a "good" MR-Sort model in the population
- Stopping criteria :
  - If one of the models restores all examples
  - Or after N iterations

Learning a NCSM model - Meta II

Recall : Metaheuristic to learn a MR-Sort model

- Principle (genetic algorithm) :
  - Initialize a population of MR-Sort models
  - Evolve the population by iteratively
    - Optimizing weights (profiles fixed) with a LP
    - Improving profiles (weights fixed) with a heuristic
    - Selecting the best models and reinitializing the others
  - ... to get a "good" MR-Sort model in the population
- Stopping criteria :
  - If one of the models restores all examples
  - Or after N iterations

Metaheuristic to learn a NCSM model

- Adaptation of the LP to learn capacities and adaptation of the heuristic

Learning a NCSM model - Meta III

Linear Program to learn the capacities and the majority threshold

- Learning of capacities based on fixed profiles.
- Expression of the capacities with the Möbius transform.
- Limitation to 2-additive capacities to limit the number of variables and constraints.

Heuristic to adjust the profiles

- Same principles as in [Sobrie et al., 2013], adapted for capacities instead of weights.
- Multiple iterations per profile and per criteria.
- Profile moved in order to increase the number of correct assignments.

Experimentations I

Dataset	#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

- Instances split in two parts : learning set and test set.
- Binarization of the categories.

Source : [Tehrani et al., 2012]

Experimentations II

Average Classification Accuracy

Dataset	META MR-Sort	META NCSM
DBS	0.8400 ± 0.0456	0.8306 ± 0.0466
CPU	0.9270 ± 0.0294	0.9203 ± 0.0315
BCC	0.7271 ± 0.0379	0.7262 ± 0.0377
MPG	0.8174 ± 0.0290	0.8167 ± 0.0468
ESL	0.8992 ± 0.0195	0.9018 ± 0.0172
MMG	0.8303 ± 0.0154	0.8318 ± 0.0121
ERA	0.6905 ± 0.0192	0.6927 ± 0.0165
LEV	0.8454 ± 0.0221	0.8445 ± 0.0223
CEV	0.9217 ± 0.0067	0.9187 ± 0.0153

- 50% of the dataset used as learning set
- Results are not convincing, overfitting ?

Experimentations II

Average Classification Accuracy

Dataset	META MR-Sort	META NCSM
DBS	0.9318 ± 0.0036	0.9247 ± 0.0099
CPU	0.9761 ± 0.0000	0.9694 ± 0.0072
BCC	0.7737 ± 0.0013	0.7700 ± 0.0077
MPG	0.8418 ± 0.0000	0.8418 ± 0.0000
ESL	0.9180 ± 0.0000	0.9180 ± 0.0000
MMG	0.8491 ± 0.0011	0.8508 ± 0.0005
ERA	0.7142 ± 0.0028	0.7158 ± 0.0004
LEV	0.8650 ± 0.0000	0.8650 ± 0.0000
CEV	0.9225 ± 0.0000	0.9225 ± 0.0000

- Full dataset used as learning set.
- Results are not convincing.

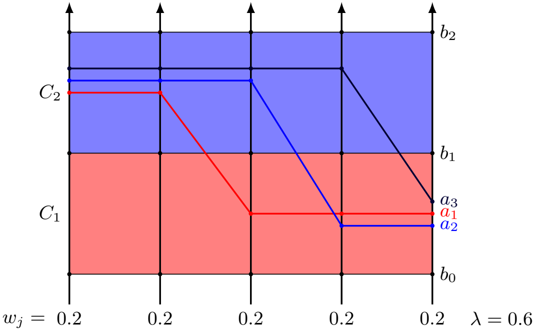
Comments

What to conclude after the experiments ?

- Algorithm not well adapted ?
- Expressivity of the model is not so much improved ?
- To what extent MR-Sort approximates non-additive learning sets ?

Non-additive set approximation with MR-Sort

- Boolean function : function $f : \{0, 1\}^n \rightarrow \{0, 1\}$.
- MBF : $f(x_1, x_2, \ldots, x_n) \geq f(y_1, y_2, \ldots, y_n)$ if $x_i \geq y_i$ for $i = \{1, ..., n\}$.
- The weights and cut threshold of one MR-Sort model define one MBF.



$(1 \quad 1 \quad 0 \quad 0 \quad 0) \rightarrow 0 \quad (\sum_{j=1}^{5} w_j = 0.4 < \lambda)$

$(1 \quad 1 \quad 1 \quad 0 \quad 0) \rightarrow 1 \quad (\sum_{j=1}^{5} w_j = 0.6 = \lambda)$

$(1 \quad 1 \quad 1 \quad 1 \quad 0) \rightarrow 1 \quad (\sum_{j=1}^{5} w_j = 0.8 > \lambda)$

Non-additive set approximation with MR-Sort

- Boolean function : function $f : \{0,1\}^n \rightarrow \{0,1\}$.
- MBF : $f(x_1, x_2, \ldots, x_n) \geq f(y_1, y_2, \ldots, y_n)$ if $x_i \geq y_i$ for $i = \{1, ..., n\}$.
- The weights and cut threshold of one MR-Sort model define one MBF.
- Number of MBFs (Dedekind number) :

n	$D(n)$
0	2
1	3
2	6
3	20
4	168
5	7 581
6	7 828 354
7	2 414 682 040 998
8	56 130 437 228 687 557 907 788
9	? ? ?

Non-additive set approximation with MR-Sort

- How many MBFs are not additive, i.e. cannot be represented with a MR-Sort model ?
- For MBFs that are not representable with MR-Sort : how many assignments are wrong ?
- Generation of all MBFs for $n \leq 6$.
- For each MBF :
  1. Generation of 2^n different binary vectors of performances and assignment of these vectors according to the MBF.
  2. Learning of a MR-Sort model with a MIP that minimize the 0/1 loss.

n	$D(n)$	% non-additive	0/1 loss		
			min.	max.	avg.
4	168	11 %	1/16	1/16	1/16
5	7 581	57 %	1/32	3/32	1.26/32
6	7 828 354	97 %	1/64	8/64	2.73/64

- Few alternatives are incorrectly assigned

Conclusion

- For problems involving small number of criteria (< 7), we don't win so much in expressivity with NCSM
- Metaheuristic can be improved to better deal with interactions
- Tests with datasets in which there exist interactions between criteria

Thank you for your attention !

References I

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

Leroy, A., Mousseau, V., and Pirlot, M. (2011).
Learning the parameters of a multiple criteria sorting method.
In Brafman, R., Roberts, F., and Tsoukiàs, A., editors, *Algorithmic Decision Theory*, volume 6992 of *Lecture Notes in Artificial Intelligence*, pages 219–233. Springer Berlin / Heidelberg.

References II

Sobrie, O., Mousseau, V., and Pirlot, M. (2012).
Learning the parameters of a multiple criteria sorting method from large sets of assignment examples.
In *DA2PL 2012 Workshop From Multiple Criteria Decision Aid to Preference Learning*, pages 21–31.
Mons, Belgique.

Sobrie, O., Mousseau, V., and Pirlot, M. (2013).
Learning a majority rule model from large sets of assignment examples.
In Perny, P., Pirlot, M., and Tsoukiás, A., editors, *Algorithmic Decision Theory*, Lecture Notes in Artificial Intelligence, pages 336–350.
Springer.

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

References III