Learning Majority rule and non-compensentory sorting

Mustapha AJEGHRIR | Asmae KHALD | Nouamane TAZI

CentraleSupélec – Département Informatique

1er février 2022

Formulation of MR sort : no noise

$$
\begin{aligned}
&\textit{Maximize} \quad \alpha \\
&\textit{subject to} \sum_{i \in N} c_{ij}^{h} + x_j + \varepsilon = \lambda && \forall a_j \in C_h, \forall h \in K_{no\_end} \\
&\sum_{i \in N} c_{ij}^{h-1} = \lambda + y_j && \forall a_j \in C_h, \forall h \in \{2, \dots, k\} \\
&\alpha \leq x_j, \alpha \leq y_j && \forall a_j \in C^* \\
&c_{ij}^{l} \leq w_i && \forall i \in N, \forall a_j \in C_h, \forall h \in K, \forall l \in \{h, h-1\} \cap K_{no\_end} \\
&c_{ij}^{l} \leq \delta_{ij}^{l} && \forall i \in N, \forall a_j \in C_h, \forall h \in K, \forall l \in \{h, h-1\} \cap K_{no\_end} \\
&c_{ij}^{l} \geq \delta_{ij}^{l} - 1 + w_i && \forall i \in N, \forall a_j \in C_h, \forall h \in K, \forall l \in \{h, h-1\} \cap K_{no\_end} \\
&M\delta_{ij}^{l} + \varepsilon \geq g_i(a_j) - b_i^{l} && \forall i \in N, \forall a_j \in C_h, \forall h \in K, \forall l \in \{h, h-1\} \cap K_{no\_end} \\
&M(\delta_{ij}^{l} - 1) \leq g_i(a_j) - b_i^{l} && \forall i \in N, \forall a_j \in C_h, \forall h \in K, \forall l \in \{h, h-1\} \cap K_{no\_end} \\
&\sum_{i \in N} w_i = 1, \quad \lambda \in [0.5, 1] \\
&w_i \in [0, 1] && \forall i \in N \\
&c_{ij}^{l} \in [0, 1], \delta_{ij}^{l} \in \{0, 1\} && \forall i \in N, \forall a_j \in C_h, \forall h \in K, \forall l \in \{h, h-1\} \cap K_{no\_end} \\
&x_j, y_j \in \mathbb{R} && \forall a_j \in C^* \\
&\alpha \in \mathbb{R}
\end{aligned}
$$

Formulation of MR sort : noise

$$
\begin{aligned}
\textit{Maximize} \quad & \sum_{a_j \in C^*} \gamma_j \\
\textit{subject to} \; & \sum_{i \in N} c_{ij}^{h} + \varepsilon \leq \lambda + M(1-\gamma_j) && \forall a_j \in C_h, \forall h \in K_{no\_end} \\
& \sum_{i \in N} c_{ij}^{h-1} \geq \lambda - M(1-\gamma_j) && \forall a_j \in C_h, \forall h \in \{2,\dots,k\} \\
& c_{ij}^{l} \leq w_i && \forall i \in N, \forall a_j \in C_h, \forall h \in K, \forall l \in \{h, h-1\} \cap K_{no\_end} \\
& c_{ij}^{l} \leq \delta_{ij}^{l} && \forall i \in N, \forall a_j \in C_h, \forall h \in K, \forall l \in \{h, h-1\} \cap K_{no\_end} \\
& c_{ij}^{l} \geq \delta_{ij}^{l} - 1 + w_i && \forall i \in N, \forall a_j \in C_h, \forall h \in K, \forall l \in \{h, h-1\} \cap K_{no\_end} \\
& M\delta_{ij}^{l} + \varepsilon \geq g_i(a_j) - b_i^{l} && \forall i \in N, \forall a_j \in C_h, \forall h \in K, \forall l \in \{h, h-1\} \cap K_{no\_end} \\
& M(\delta_{ij}^{l} - 1) \leq g_i(a_j) - b_i^{l} && \forall i \in N, \forall a_j \in C_h, \forall h \in K, \forall l \in \{h, h-1\} \cap K_{no\_end} \\
& b_i^{h+1} \geq b_i^{h} && \forall i \in N, \forall h \in K_{no\_no\_end} \\
& \sum_{i \in N} w_i = 1, \quad \lambda \in [0.5, 1] \\
& w_i \in [0,1] && \forall i \in N \\
& c_{ij}^{l} \in [0,1], \delta_{ij}^{l} \in \{0,1\} && \forall i \in N, \forall a_j \in C_h, \forall h \in K, \forall l \in \{h, h-1\} \cap K_{no\_end} \\
& x_j, y_j \in \mathbb{R} && \forall a_j \in C^*
\end{aligned}
$$

Data generation

Data generation method

alternative	criterion 0	criterion 1	criterion 2	criterion 3	class
0	8.02	14.22	9.84	9.03	0
1	8.96	10.78	11.07	9.03	0
2	12.61	10.76	15.98	10.54	2
3	10.13	10.77	9.93	8.92	1

Figure – Example of instances generated

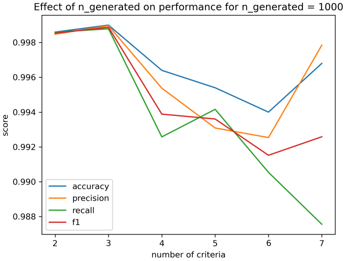
Results : no noise

Generalisation	Predicted Class 0	Predicted Class 1	Predicted Class 2
True Class 0	408	0	0
True Class 1	0	462	0
True Class 2	0	8	122

Reconstruction	Predicted Class 0	Predicted Class 1	Predicted Class 2
True Class 0	429	0	0
True Class 1	0	465	0
True Class 2	0	1	105

Figure – Confusion matrix for the configuration ($n = 3$, $p = 2$, $n_{generated} = 1000$). Inference time : 40s

Results : no noise



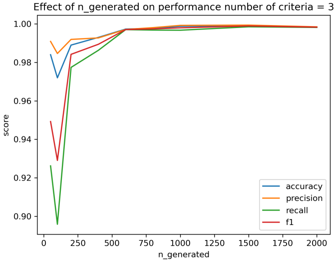


Figure – The MR-sort rules are first being inferred over a training split then tested over a testing split. The two splits are being generated by the same rules with only 2 classes $(p = 1)$, results are the mean of 5 random runs. In the right we can see the effect of varying the number of elements in the training split $n_{generated}$, in the left we can see the effect of varying the number of criteria

Results : no noise

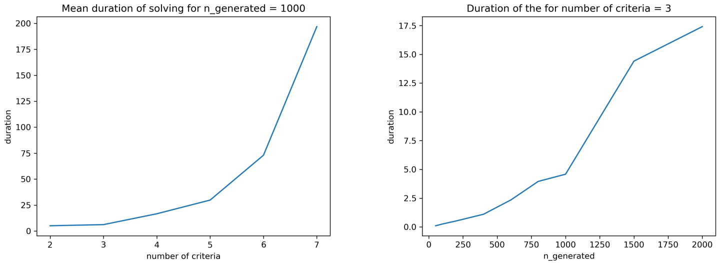


Figure – Duration of the MR-sort inference over the training split. The split is being generated by the same rules with only 2 classes $(p = 1)$, results are the mean of 5 random runs. In the right we can see the effect of varying the number of elements in the training split $n_{generated}$, in the left we can see the effect of varying the number of criteria

Majority rule sorting

Results : noise

Generalisation	Predicted Class 0	Predicted Class 1	Predicted Class 2
True Class 0	80	2	0
True Class 1	3	95	2
True Class 2	0	8	18

Reconstruction	Predicted Class 0	Predicted Class 1	Predicted Class 2
True Class 0	84	3	0
True Class 1	1	90	2
True Class 2	0	1	20

Figure – Confusion matrix for the configuration ($n = 3$, $p = 2$, $n_{generated} = 200$, $N = 0.1$). Inference time : $50s$, real errors 12

Reconstruction rate $= 97\%$

SAT Formulation of NCS

$$
\begin{aligned}
\Phi_\alpha^C &= \phi_\alpha^{C1} \wedge \phi_\alpha^{C2} \wedge \phi_\alpha^{C3} \wedge \phi_\alpha^{C4} \wedge \phi_\alpha^{C5} \wedge \phi_\alpha^{C6} \\
\phi_\alpha^{C1} &= \bigwedge_{i \in \mathcal{N}, k \in [2.p]} \bigwedge_{x' \succsim_i x \in X^*} \left(a_{i,k,x'} \vee \neg a_{i,k,x}\right) \\
\phi_\alpha^{C2} &= \bigwedge_{i \in \mathcal{N}, k < k' \in [2.p], x \in X^*} \left(a_{i,k,x} \vee \neg a_{i,k',x}\right) \\
\phi_\alpha^{C3} &= \bigwedge_{B \subset B'} \subseteq \mathcal{N}, k \in [2.p] \quad \left(t_{B',k} \vee \neg t_{B,k}\right) \\
\phi_\alpha^{C4} &= \bigwedge_{B \subseteq \mathcal{N}, k < k' \in [2.p]} \left(t_{B,k} \vee \neg t_{B,k'}\right) \\
\phi_\alpha^{C5} &= \bigwedge_{B \subseteq \mathcal{N}, k \in [2.p]} \bigwedge_{x \in \alpha^{-1}(C^{k-1})} \quad \left(\bigvee_{i \in B} \neg a_{i,k,x} \vee \neg t_{B,k}\right) \\
\phi_\alpha^{C6} &= \bigwedge_{B \subseteq \mathcal{N}, k \in [2.p]} \bigwedge_{x \in \alpha^{-1}(C^{k})} \quad \left(\bigvee_{i \in B} a_{i,k,x} \vee t_{\mathcal{N} \setminus B,k}\right)
\end{aligned}
$$

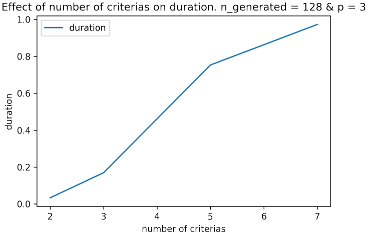
Max-SAT Formulation of NCS

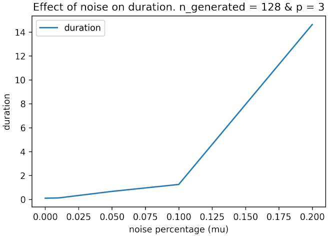
$$\Phi_{\alpha}^{\widetilde{CS}} = \bigwedge_{B \subseteq \mathcal{N}, k \in [2.p]} \bigwedge_{x \in \alpha^{-1}(C^{k-1})} \left(\bigvee_{i \in B} \neg a_{i,k,x} \vee \neg t_{B,k} \vee \neg z_X \right)$$
$$\phi_{\alpha}^{\widetilde{C6}} = \bigwedge_{B \subseteq \mathcal{N}, k \in [2.p]} \bigwedge_{x \in \alpha^{-1}(C^{k})} \left(\bigvee_{i \in B} a_{i,k,x} \vee t_{\mathcal{N} \setminus B,k} \vee \neg z_x \right)$$
$$\phi_{\alpha}^{\text{goal}} = \bigwedge_{x \in \mathbb{X}^*} z_x$$

We added new variable z

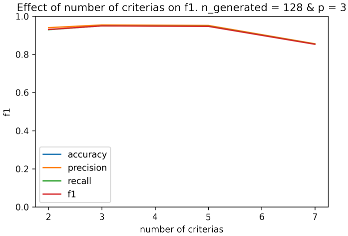
' z ' variables, indexed by an alternative x, represent the set of alternatives properly classified by the inferred model, with the following semantic :
$z_x = 1 \Leftrightarrow \alpha^{-1}(x) = NCS_{\omega}(x)$ i.e. the alternative x is properly classified.

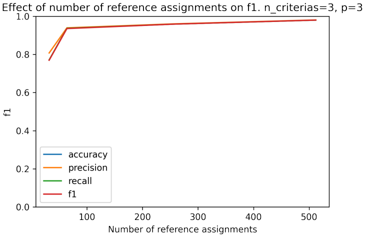
Results : duration

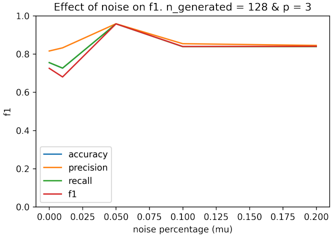




Results : performance







Results : confusion matrix

Generalisation	Predicted Class 0	Predicted Class 1	Predicted Class 2
True Class 0	330	2	0
True Class 1	0	333	1
True Class 2	0	0	334

Figure – Confusion matrix for the configuration 'coalitions' : [[0], [1], [2]], 'criteria' : [0, 1, 2], μ : $0.1,' n\_ground\_truth'$: $1000,' n\_learning\_set'$: $1024,' profiles'$: $array([[3, 1, 10], [6, 12, 16]])$
restoration rate = 100%
f1 score, accuracy, recall, precision = 1

Non compensatory sorting

Max-SAT Formulation of NCS (single-peaked)

To solve the problem for single-peaked, we only had to change this equation.

$$\phi_{\alpha}^{C1} = \bigwedge_{i \in \mathcal{N}, k \in [2.p]} \bigwedge_{x' \succsim_i x" \succsim_i x \in X^*} \left(a_{i,k,x"} \vee \neg a_{i,k,x} \vee \neg a_{i,k,x'} \right)$$

Conclusion

Our experimental results show that the duration of computation evolve exponentially with respect to the number of criteria for MR-Sort, linearly for Max-Sat, and linearly with respect to the size of the learning set. And the generalization index increases with the size of the learning set and decreases with the addition of number of criteria, all while restoring at least $1 - x$ of the data with x the noise percentage.