



CentraleSupélec

Conception et optimisation d'algorithmes d'apprentissage automatique de
préférences

Rapport Parcours Recherche 1A

LEWY NATHAN

20 juin 2024



CentraleSupélec

1 Cadre du projet

1.1 Introduction

Les moteurs de recherche en ligne sont devenus la manière privilégiée de rechercher l'information aujourd'hui. Cependant, il n'y a jamais eu autant de données à traiter puis sélectionner avant de les présenter à l'utilisateur. Que ce soit Youtube, Trivago ou Leboncoin, tous ces moteurs ont une manière de hiérarchiser, suivant les préférences de l'utilisateur, les alternatives qu'ils proposent. Ces préférences utilisateurs peuvent lui être connues ou non. Lorsqu'elles sont inconnues, mais que l'on connaît un ensemble de décisions prises par l'algorithme de classification, on peut tenter d'évaluer ces préférences.

1.2 Objectif

L'objectif est d'étudier les méthodes de classification ainsi que les algorithmes d'apprentissage de préférences qui leurs sont liées, en particulier du modèle NCS (Non Compensatory Sorting).

2 Formulation des problèmes d'agregation de préférences

2.1 Alternatives et préférences

Définition 1 *Un modèle d'évaluation est un ensemble de 5 éléments :*

$$\mathcal{M} = \{\mathcal{A}, \{\mathcal{D}, \mathcal{E}\}, \mathcal{H}, \mathcal{U}, \mathcal{R}\}$$

- \mathcal{A} est l'ensemble des alternatives à classifier
- \mathcal{E} est l'ensemble des critères sur lesquels on évalue une alternative. Les $X_i \in \mathcal{E}$ sont munis d'une structure, souvent d'ordres partiels \lesssim_i . Chaque élément de \mathcal{D} appelé attribut, est une fonction de \mathcal{A} vers X_i
- \mathcal{U} est l'ensemble d'incertitudes sur les attributs ou sur les critères de préférence.
- \mathcal{H} est l'ensemble des critères permettant de modéliser les préférences utilisateurs. Sur \mathcal{H} et \mathcal{D} on peut définir un ensemble \mathcal{R} d'opérateurs d'agregation, permettant de construire des fonctions faisant intervenir chacun des critères.

Par exemple, on peut prendre un ensemble \mathcal{A} de voitures, évaluées sur les critères \mathcal{E} de vitesse, de prix et de freinage, chacun pouvant être muni d'une relation d'ordre. On peut définir l'agregation suivante : si pour une alternative la somme des valeurs de ces critères est supérieure à un seuil fixé par les préférences utilisateur \mathcal{H} , alors la voiture est souhaitable, et à rejeter sinon. On se place ici dans le cas où \mathcal{E} est fini et on possède à disposition n relations d'ordre $p = (\lesssim_1, \dots, \lesssim_n)$.

2.2 Méthodes d'agrégation de préférences

Il est nécessaire de définir une méthode d'agrégation adaptée au problème. En effet : Prenons $n = 2$ et 2 alternatives a, b telles que $a \lesssim_1 b$ et $b \lesssim_2 a$. On n'a pas de manière de dire quelle alternative est meilleure *a priori*. Cependant si on munit chaque critère X_i d'un poids w_i , on peut définir la relation d'ordre $\leq_{1,2}$ telle que :

$$x \leq_{1,2} y \Leftrightarrow \sum_{x_i \leq_i y_i} w_i \leq \sum_{y_i \leq_i x_i} w_i \quad (1)$$

Pourvu que les poids soient différents, la bonne méthode d'agrégation permet ici de trancher dans des situations où ça ne semble de prime abord pas possible. Cette méthode, appelée **méthode de Condorcet pondérée** peut être étendue à n critères. Habituellement, on souhaite caractériser le comportement de la méthode pour ensuite trouver son fonctionnement, et en effet, le théorème suivant est vérifié [5] :

Théorème 1 *Seulement la méthode de Condorcet pondérée vérifie tous ces axiomes :*

- **Anonymité pondérée** : Les critères ont le même rôle mais leur poids a une importance.
- **Convexité** : Soient deux profils de pondération w_1, w_2 . Si une alternative a est supérieure à b pour w_1 et w_2 , elle l'est aussi pour $\frac{w_1 + w_2}{2}$.
- **Archimédien** : La grande pondération d'un critère revient à l'ordre partiel sur ce critère.
- **Indépendance de critères** : Un critère n'a pas d'importance si son poids est nul.
- **Neutralité** : L'agrégation ne dépend pas du nom des alternatives.
- **Réponse positive** : Si la valeur dans un critère d'une alternative augmente et que les autres restent fixes, sa classification s'améliorera.
- **Fidélité** : Pour un seul critère, le meilleur élément a la plus haute valeur dans ce critère.
- **Indépendance des alternatives** : L'agrégation des alternatives dépend seulement des valeurs dans les critères.

3 Le modèle NCS

Nous venons de voir comment comparer des alternatives en construisant une méthode d'agrégation. Cependant il peut être utile de perdre l'information de comparaison sur certaines alternatives en les regroupant en échange dans des catégories. Voyons alors le modèle de classification NCS, qui reprend l'idée d'accorder différents degrés d'importances aux critères.

3.1 Définition

Définition 2 Soit $C^1 \prec \dots \prec C^p$ un ensemble de p catégories ordonnées.

Soit $\mathcal{E} = \prod_{i=1}^n X_i$, où chaque X_i est muni d'un préordre total \lesssim_i .

Soit $(b_i^h)_{i,h} \in \mathcal{M}_{p-1}(\mathcal{E})$ telle que $b_i^h \lesssim_i b_i^{h-1}$ pour tout i et tout $h \in \llbracket 2, p-1 \rrbracket$. Pour tout h , la famille (b_i^h) est un profil limite de la catégorie $h-1$.

Soient des sous-ensembles de $\mathcal{P}(\llbracket 1, n \rrbracket)$ ordonnés par l'inclusion $\mathcal{F}^1 \subseteq \dots \subseteq \mathcal{F}^{p-1}$. Ces ensembles constituent les coalitions (ensembles) de critères suffisants pour chaque catégorie.

On peut maintenant définir la règle de classification du modèle **NCS**.

Une alternative $x = (x_1, \dots, x_n)$ appartient à la classe $h = \mathbf{NCS}_{(b_i^h, \mathcal{F}^k)}(x)$ si et seulement si $\{i \in \llbracket 1, n \rrbracket, b_i^h \lesssim_i x_i\} \in \mathcal{F}^h$ et que $\{i \in \llbracket 1, n \rrbracket, b_i^{h+1} \lesssim_i x_i\} \notin \mathcal{F}^{h+1}$

Dans ce modèle, les préférences utilisateurs \mathcal{H} sont modélisées par les profils limites ainsi que les coalitions. Il n'y a pas cependant pas de notion d'incertitude.

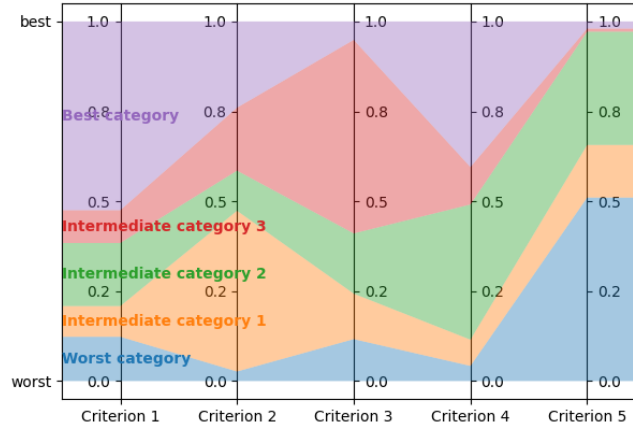


FIGURE 1 – Représentation des profils limites d'un modèle NCS

3.2 Reformulations et simplifications

Ce modèle est vaste et il peut être intéressant d'en étudier des cas plus restreint ou reformulés. J'ai pu par exemple aussi étudier la reformulation en problème de satisfaisabilité (SAT) de NCS/invNCS et aussi de maximum de satisfaisabilité (MaxSAT), qui offrent de meilleurs résultats [1] mais ils seraient trop longs à inclure ici.

Définition 3 le modèle $\mathbf{U}^b\mathbf{NCS}$ est le modèle **NCS** pour lequel les profils limites sont les mêmes pour chaque catégorie et les ensembles de critères suffisants varient.

Définition 4 le modèle $\mathbf{U}^c\mathbf{NCS}$ est le modèle **NCS** pour lequel les ensembles de critères suffisants sont les mêmes pour chaque catégorie et les profils limites varient.

On peut remarquer que l'on peut définir un modèle $\mathbf{U}^c\mathbf{NCS}$ à partir du modèle MR-Sort. Ces deux modèles sont même équivalents si le nombre de catégories est 2.

4 Apprentissage des préférences

Maintenant, le modèle NCS étant défini, on peut présenter le problème inverse **invNCS**. L'objectif est d'associer à une classification donnée un modèle NCS.

Définition 5 Soit $\alpha : \mathbb{X} \rightarrow \{C^1, \dots, C^p\}$. On dit que α peut être représentée par un modèle NCS si il existe des profils limites $(b_i^h)_{i,h}$ et des ensembles de coalitions de critères $(\mathcal{F}^k)_k$ tels que $\mathbf{NCS}_{(b_i^h, \mathcal{F}^k)} = \alpha$

Cependant ce problème a été montré comme étant NP difficile [4]. Les méthodes présentée ici ne vise donc pas à trouver si une telle représentation existe pour un α donné, mais plutôt trouver les paramètres du modèle NCS qui la représenterait le mieux. Des méthodes heuristiques existent déjà pour apprendre ces paramètres [6] [2].

4.1 Réseau de neurones

Pour trouver ces paramètres optimaux, on utilise ici un réseau de neurones. Il s'agit d'un problème classique de classification. On utilise l'algorithme suivant, en se restreignant d'abord à U^cNCS :

Algorithm 1 Algorithme d'apprentissage des paramètres

```

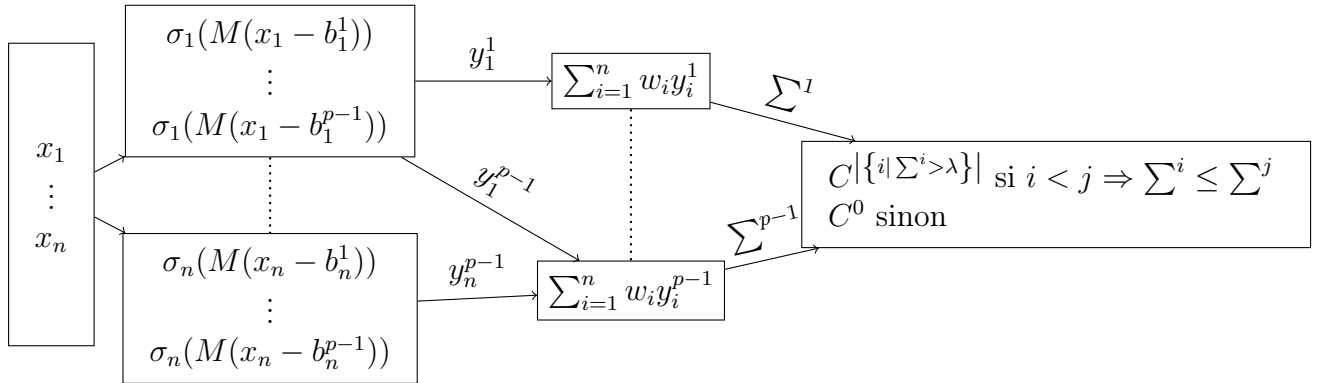
 $n, p \in \mathbb{N}$ 
Générer  $\mathcal{M}$  un modèle  $U^cNCS$  avec  $n$  critères et  $p$  catégories
Générer un dataset regroupant alternatives et catégories
Générer un réseau  $\mathcal{R}$  de neurones ANN-MRSort
while Apprentissage do
    Evaluer sur un nouveau batch  $b$  de données  $\mathcal{R}(b)$ 
    Calculer l'erreur avec  $\mathcal{M}(b)$  et la descente de gradient
    Modifier les paramètres de  $\mathcal{R}$ 
end while

```

Le réseau de neurones [3] utilisé (ANN-MR-Sort) consiste à approximer le modèle NCS en un modèle MR-Sort. Au lieu de modéliser les coalitions comme des ensembles, on associe à chaque critère de chaque catégorie un poids, et la somme de ces poids doit être supérieure au seuil pour être au moins dans cette catégorie.

Voici une représentation schématique du réseau.

- x_i est la composante selon le critère i de l'alternative x
- b_j^i est le **biais** associé au critère j pour la catégorie j
- M est la **dureté** de la sigmoïde. Idéalement, elle est prise faible au début, puis augmente au fur et à mesure de l'apprentissage.
- σ est la fonction sigmoïde
- w_i sont les **poids fictifs** des catégories
- $y_j^i = \sigma(M(x_j - b_j^i))$



Ci dessous, le graphe du taux de classifications correctes au cours de l'apprentissage du réseau pour un modèle NCS à 5 catégories et 20 critères :

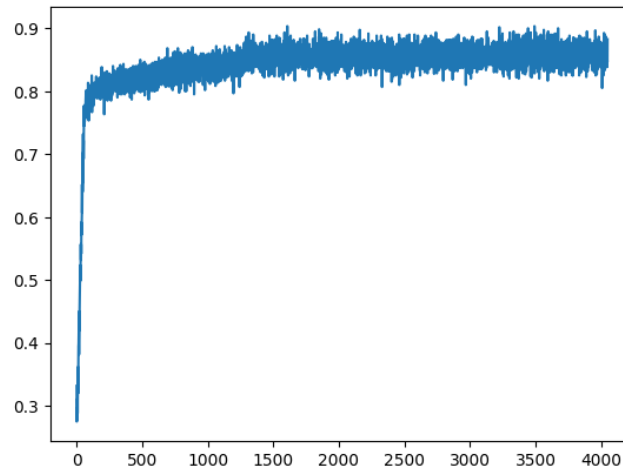


FIGURE 2 – Graphe du taux de classifications correctes sur chaque batch en fonction du nombre de batches déjà traités

5 Conclusion

J'ai pu au cours de cette année faire un pas vers l'étude des méthodes d'apprentissage de préférences, en traversant plusieurs aspects : L'étude des méthodes d'agrégation, des différentes formulations du problème NCS, certaines méthodes de résolution de invNCS. J'ai aussi pu effectuer l'implémentation d'un algorithme d'apprentissage, mais elle devrait être poussée plus loin pour arriver à des résultats plus concrets.

Références

- [1] Oumaima Khaled Vincent Mousseau Wassila Ouerdane Ali Tlili, Khaled Belahcène. Learning non-compensatory sorting models using efficient sat/maxsat formulations. *European Journal of Operational Research*, 298 :979–1006, 2021.
- [2] Khaled Belahcene. Towards accountable decision aiding : explanations for the aggregation of preferences. *Université Paris Saclay (COMUE)*, 2018.
- [3] V Team Automatants Golasowski. Formations pytorch. *Retrieved from Automatants website : <https://automatants.cs-campus.fr/formations>*, 2023.
- [4] N. Maudet V. Mousseau K. Belahcène, C. Labreuche and W. Ouerdane. An efficient sat formulation for learning multiple criteria non-compensatory sorting rules from examples. *Computers Operations Research*, 97 :58–71, 2018.
- [5] Patrick Meyer Vincent Mousseau Marc Pirlot Raymond Bisdorff, Luis C. Dias. Evaluation and decision models with multiple criteria. *International Handbooks on Information Systems*, 2015.
- [6] Olivier Sobrie. Learning preferences with multiple-criteria models. *Université Paris Saclay (COMUE), Université de Mons*, 2016.