

Dominance based monte carlo : une méthode d'élicitation des préférence pour le problème du tri multi-critère

Tom Denat¹, Meltem Öztürk¹

Université Paris Dauphine, PSL Research University, CNRS, LAMSADE, 75016 Paris, France
{tom.denat,meltem.ozturk}@lamsade.dauphine.fr

Introduction Nous étudions ici un algorithme d'élicitation des préférences pour le problème du Tri multi-critère qui consiste à affecter des objets parmi un ensemble pré-défini de catégories en se basant sur les évaluation de ces objets sur un ensemble des critères prédéfinis. Pour ce faire notre méthode se base sur un échantillon d'objets triés par le décideur (le learning set). Notre méthode réuni deux propriétés plutôt rares en aide multi-critère à la décision : l'absence de modèle et le fonctionnement aléatoire. Ici l'"absence de modèle" signifie qu'il n'est pas supposé que le raisonnement du décideur est fondé sur un ensemble de règle connues de l'analyste. Comme tout algorithme de monte carlo son fonctionnement est non déterministe. La monotonie (améliorer un objet sur un critère ne peut pas le rendre globalement moins bon) et le fait de retourner systématiquement le learning set dans la catégorie à laquelle le décideur l'a affecté, considérés dans les autres méthodes comme de bonnes propriétés, constituent les uniques contraintes qui encadrent celles ci. Notre méthode que nous appelons Dominance Based Monte Carlo (DBMC) prend en paramètres : un ensemble de critères N exprimés sur des échelles finies et discrètes v_i , un ensemble d'objets A (ici chaque combinaison de valeurs sur les critères forme un objet) et un ensemble de catégories C . Il fonctionne comme suit et décrit par l'algorithme 1. Le décideur fournit un learning set L (il affecte certains objets à des catégories). Ensuite un objet est choisi au hasard puis on l'affecte à une catégorie au hasard parmi les catégories auxquelles cet objet peut appartenir sans violer la monotonie. Cette affectation génère de nouvelles contraintes sur les classifications possible des autres objets. Un autre objet est ensuite choisi au hasard et affecté à une catégorie et ce jusqu'à ce que chaque objet soit affecté à une catégorie. Cette classification que nous appelons un lancer est hautement aléatoire. Afin de corriger cette propriété, on effectue T lancers. A la fin du processus chaque objet est affecté dans une classification globale à la catégorie médiane des catégories auxquelles il a été affecté au cours des T lancers.

Algorithm 1: DBMC algorithm

Input: Problème de Tri $S = \langle N, V, A, C, P \rangle$

Output: Classification $f_1 : A \rightarrow C$ monotone, stable et compatible avec L

- 1 **for** j de 1 à T **do**
 - 2 $S' \leftarrow S$
 - 3 Compléter aléatoirement S' par l'algorithme 2
 - 4 Affecter chaque objet à la médiane des catégorie médiane de ses affectations au cours des T lancers
-

Distribution de probabilités des lancers :

Comme nous avons pu le voir le résultat d'un lancer est aléatoire. Si nous pouvions connaître la distribution de probabilité de l'affectation de chaque objet, alors l'affectation médiane pourrait être calculée sans qu'il ne soit nécessaire d'appliquer l'algorithme. Malheureusement, cette distribution semble très difficile à calculer. Nous avons néanmoins prouvé que chaque classification monotone compatible avec le learning set peut être obtenu avec une probabilité strictement positive bien que cette distribution ne soit pas uniforme parmi toutes les classifications monotones compatibles avec le learning set.

Algorithm 2: Random completion - trial

Data: Problème de Tri $S = \langle N, V, A, C, L \rangle$

Result: Classification $f_1 : A \rightarrow C$ monotone et compatible avec L

- 1 **while** *Il existe des objets non affectés* **do**
 - 2 Choisir un objet aléatoirement χ en suivant une distribution uniforme parmi A ;
 - 3 Choisir aléatoirement un catégorie Δ with a uniform discrete distribution between $\gamma_{min}(\chi)$ and $\gamma_{max}(\chi)$;
 - 4 Add the information $\langle \chi, \Delta \rangle$ to the learning set;
-

Propriété théoriques :

Nous avons étudié les propriétés théoriques de notre algorithme. Chaque lancer respecte la monotonie et le learning set. De ce fait, le résultat de l'algorithme respect également ses deux propriétés. Le résultat de l'algorithme est théoriquement sujet à l'aléatoire mais nous avons prouvé qu'il converge presque sûrement lorsque le nombre de lancers tend vers $+\infty$. De plus nos tests appliqués ont montré qu'à partir de 100 lancers les résultats sont relativement stables. La complexité algorithmique de l'algorithme DBMC est de l'ordre de $O(m^2 \times T)$ et il tourne en une heure avec 100.000 objets et 100 lancers.

Validations pratiques :

Afin de tester les performance de prédiction de l'algorithme DBMC nous avons appliqué un teste nommé 2-fold validation. Il consiste à utiliser un learning set réel que l'on divise aléatoirement en deux jeu de données de taille identique. Ensuite l'algorithme est appliqué pour apprendre sur une moitié des données, tenter de prédire l'autre moitié et comparer cette prédiction avec l'affectation réelle. Les trois learning sets utilisés, car evaluation (CEV), lecture evaluation (LEV) et breast cancer (BCC) ont également été testé par [Sobrie et al., 2015] sur deux autres algorithmes d'élicitation pour le tri : non-compensatory sorting et MR-Sort.

	NCS	MR-Sort	DBMC
CEV	$8.82 \pm 2.63\%$	$9.36 \pm 1.19\%$	$3.69 \pm 0.27\%$
LEV	$14.92 \pm 1.88\%$	$15.92 \pm 1.22\%$	$18.66 \pm 1.1\%$
BCC	$26.72 \pm 3.45\%$	$27.5 \pm 3.79\%$	$25.96 \pm 0.61\%$

TAB. 1 – Résultat du 2-fold validation. Les chiffres présentés dans le tableau 1 représentent le pourcentage d'erreur dans la prédiction.

Nous proposons une méthode d'élicitation qui présente de bonnes propriétés théoriques et des résultats pratiques comparables aux autres algorithmes d'élicitation pour le Tri. L'absence de modèle peut être vue comme une bonne ou une mauvaise propriété selon le contexte. Dans le cadre d'une décision publique un modèle peut aider à justifier la décision. Dans le cas de l'étude d'un jeu de données représentant des préférences humaines il se peut qu'aucun modèle ne puisse a priori être choisi pour les représenter. Notre algorithme doit être appliqué lorsque les échelles utilisées sont finies et discrètes. Dans le cas inverse une discrétisation peut être envisagées sous certaines précautions.

Références

[Sobrie et al., 2015] Sobrie, O., Mousseau, V., and Pirlot, M. (2015). *Learning the Parameters of a Non Compensatory Sorting Model*.