Démocratie à géométrie variable (à l'usage des algorithmes)

Théo Delemazure, François Durand, Fabien Mathieu

Démocratie à géométrie variable (à l'usage des algorithmes)[†]

Théo Delemazure[1,2] et François Durand[1] et Fabien Mathieu[1]

[1] *Nokia Bell-Labs France*
[2] *École Normale Supérieure, Paris*

De nombreux problèmes, par exemple en décision, ne peuvent se résoudre de manière exacte et font appel à des heuristiques qui attribuent des scores aux différents choix possibles. Ces heuristiques peuvent être nombreuses et plus ou moins corrélées : par exemple, en variant les hyperparamètres d'un algorithme, on peut générer toute une famille d'heuristiques fortement corrélées mais de qualité variable. Plutôt que de chercher la *meilleure* heuristique, cet article propose d'agréger les scores de manière intelligente. Notre solution, qui utilise les corrélations entre heuristiques, est plus efficace que l'agrégation triviale et plus robuste qu'une approche basée sur le maximum de vraisemblance.

Mots-clefs : Agrégation d'algorithmes, Théorie du vote, Algorithmes de décision

1 Introduction

Certains problèmes de décision ne permettent pas de trouver une solution exacte en un temps raisonnable. On a en général recours à divers algorithmes heuristiques pour résoudre ces problèmes, chacun avec ses forces et ses faiblesses.

Par exemple, dans le cas des algorithmes de régression, il n'est pas rare d'agréger les résultats de différents algorithmes afin d'améliorer la qualité de la prédiction, à la condition que les algorithmes soient diversifiés. Les méthodes de *boosting* et de *bagging* permettent notamment une certaine forme d'agrégation [GVC05].

Nous proposons d'étudier le problème de l'agrégation à travers le problème général de décision suivant : étant donné un ensemble d'alternatives et des algorithmes attribuant des scores aux différentes alternatives, comment sélectionner la « meilleure » alternative. Ce problème est proche du problème de *Multi-Attribute Decision Making* (MADM), pour lequel différentes méthodes existent [FM20, HKL11]. Dans MADM, les scores correspondent à des métriques distinctes, et il faut trouver la meilleure option pour une pondération donnée. Ici, il n'y a pas de poids et les scores correspondent à différentes estimations d'une même métrique.

Lorsque l'on a plusieurs algorithmes pour un même problème, il est fréquent que certains d'entre eux soient corrélés, par exemple lorsqu'une même technique est utilisée avec différents hyper-paramètres ou différents jeux de données lors de l'entraînement. En principe lorsque l'on fait de l'agrégation, ces corrélations sont néfastes, et les principales méthodes requièrent que le panel d'algorithmes soit *diversifié* [GVC05]. Il s'agit notamment d'éviter de biaiser les résultats dans la direction d'un potentiel *« groupe majoritaire »*. Nous proposons ici des méthodes qui s'affranchissent de cette contrainte de diversification, et utilisent ces corrélations entre algorithmes pour améliorer leurs performances. [‡]

Dans la Section 2, nous introduisons notre modèle d'algorithmes-électeurs, ainsi que quelques méthodes d'agrégation naturelles. Notre méthode d'agrégation est présentée en Section 3. Enfin, les résultats expérimentaux sont présentés dans la Section 4.

[†] Ce travail a été effectué au LINCS (`https://www.lincs.fr/`).

‡. Documentation de notre package Python «*Embedded Voting*» : `https://embedded-voting.readthedocs.io/`

2 Des algorithmes qui votent

Les notations que nous utilisons sont empruntées à la théorie du vote en choix social [BCE+16]. On considère un problème de décision tel qu'il existe n algorithmes $\mathcal{A} = \{a_1, \cdots, a_n\}$ pour résoudre ce problème de manière approchée. On propose aux algorithmes un ensemble $\mathcal{C}$ de m alternatives $c_1, \cdots, c_m$ (les *candidats*). Les algorithmes vont alors classer les différentes alternatives et choisir celle qu'ils estiment être la meilleure.

Nous supposons que chaque algorithme fait une estimation imparfaite du score de chaque alternative. Plus précisément, pour toute alternative $c_i \in \mathcal{C}$, nous faisons l'hypothèse qu'il existe un score exact $s(c_i)$, et que les différents algorithmes fournissent une version bruitée de ce score $s_j(c_i) = s(c_i) + \epsilon_j(c_i)$. Pour une alternative donnée, il peut exister des corrélations entre les bruits de plusieurs algorithmes. Par exemple, si un sous-ensemble d'algorithme ne diffère que d'un hyper-paramètre, les bruits qui leur sont associés seront probablement très similaires.

Le problème considéré ici est celui de trouver l'alternative qui maximise le score, $c^* = \text{argmax}_{c_i \in \mathcal{C}} s(c_i)$, à partir des estimations bruitées des différents algorithmes. Une solution triviale consiste à se baser sur la moyenne des estimations données par les algorithmes. En théorie du vote, cela correspond à utiliser la règle appelée *range voting*. D'autres règles simples sont la maximisation de la moyenne géométrique (produit des scores) ou du minimum des scores.

L'inconvénient de ces règles est qu'elles ne tiennent pas compte des corrélations entre algorithmes. Imaginons par exemple que parmi $n = 20$ algorithmes, 15 donnent des résultats très similaires. Lorsque ce groupe de 15 algorithmes se trompe totalement (en bloc), la moyenne des estimations est fortement impactée, même si les 5 autres proposent de bonnes estimations : un groupe d'algorithmes en surnombre a trop de pouvoir dans cette « élection ».

Dans le cas où on connaît une partition de l'ensemble des algorithmes $\mathcal{A}$ en groupes bien identifiés, on peut envisager de prendre la moyenne des scores moyens de chaque groupe. Cependant, il n'est pas rare que des groupes ne puissent pas être précisément délimités, soit car les corrélations à l'intérieur des groupes sont faibles, soit parce que certains algorithmes sont à cheval sur plusieurs groupes.

Lorsque l'on possède une information solide sur la distribution des scores ou sur la forme de cette distribution, il est raisonnable d'utiliser une règle bayésienne ou de *maximum de vraisemblance* (MV) [Eli93]. Cependant, ces méthodes sont difficilement généralisables si la distribution des scores s'éloigne d'une distribution de probabilité classique. C'est pourquoi nous proposons une méthode capable d'exploiter les corrélations entre algorithmes sans nécessiter d'a priori sur les données.

3 Un mode de scrutin géométrique

Contrairement à la plupart des règles de votes classiques, qui traitent les électeurs de façon symétrique [§], on veut ici se servir des informations supplémentaires que nous possédons sur les algorithmes-électeurs. Nous proposons d'utiliser l'analyse spectrale pour extraire et utiliser ces informations.

Afin de donner l'intuition, considérons le cas où une partition nette est connue : l'ensemble $\mathcal{A}$ des algorithmes est partitionné en $p \leq n$ groupes $G_1, \ldots, G_p$ tels que les estimations sont très similaires à l'intérieur des groupes, mais totalement indépendantes entre différents groupes. On peut donc associer à chaque algorithme $a_j \in G_k$ un vecteur d'embedding $\vec{v_j} = (\delta_l^k)_{1 \leq l \leq p}$ (des 0 partout, sauf à la position k).

On définit ensuite la matrice de score d'un candidat c_i comme

$$M_i = \begin{pmatrix} \sqrt{s_1(c_i)}\vec{v_1} \\ \cdots \\ \sqrt{s_n(c_i)}\vec{v_n} \end{pmatrix} \in \mathcal{M}_{n,p} \tag{1}$$

On note $(\lambda_1, \ldots, \lambda_p)$ les valeurs singulières de la matrice M_i, obtenues par *décomposition en valeurs singulières* (SVD) [WRR03]. Dans ce premier scénario avec p groupes parfaitement distincts, nous avons la relation $\lambda_k = \sum_{a_j \in G_k} s_j(c_i)$ (la racine carrée dans (1) permet d'avoir cette formule simple). Ainsi, chaque λ_k représente le score attribué par le groupe G_k.

§. Cette propriété est appelée *anonymat* dans la littérature du choix social.

Pour obtenir un score global pour c_i, il reste à combiner les valeurs singulières. Nous proposons d'appliquer une des fonctions du tableau 1.

Règle	SVD-Sum	SVD-Nash
Fonction d'agrégation	$\sum_k \lambda_k$	$\prod_k \lambda_k$

TABLE 1: Différentes règles basées sur les valeurs singulières.

Dans ce scénario idéal, SVD-Sum est équivalent à la somme des scores, ou *range voting*. Le produit des groupes, SVD-Nash, fait référence au concept de *Nash product* [Nas50, KN79] utilisé en théorie des jeux. Une qualité est qu'il est insensible à la taille des groupes : si l'on clone tous les algorithmes d'un groupe, les scores de toutes les alternatives sont simplement doublées et leur classement est donc inchangé.

L'emploi de la SVD peut sembler excessif dans le cas idéal où une partition parfaite est connue, mais il prend tout son sens dans le cas général où les relations entre algorithmes sont inaccessibles. Dans ce cas, nous proposons tout d'abord de calculer la matrice de covariance des algorithmes, obtenue en les testant sur un ensemble d'alternatives servant d'entraînement (il peut s'agir directement de l'ensemble des alternatives envisagées, ou d'un ensemble plus grand). Par une *analyse en composantes principales* (PCA) [WRR03] de cette matrice, on estime le nombre de dimensions sous-jacentes et on calcule les embeddings $\vec{v_j}$ des algorithmes. Enfin, on utilise ces embeddings pour appliquer la méthode décrite précédemment basée sur la décomposition en valeur singulière.

4 Résultats expérimentaux

Nous présentons ici un scénario où les algorithmes obéissent au modèle suivant, dit de *précision par groupe*, qui combine erreur de groupe et erreur individuelle. Chaque alternative c_i possède un score véritable $s(c_i)$ pris uniformément entre 10 et 20. Pour chaque groupe G_k, on tire un bruit de groupe ϵ_k selon la loi $\mathcal{N}(0, {\sigma_g}^2)$. Le bruit d'un algorithme appartenant au groupe G_k est ensuite tiré selon $\mathcal{N}(\epsilon_k, {\sigma_a}^2)$. Pour rendre l'appartenance à un groupe moins évidente, nous supposons qu'un algorithme peut également être défini comme combinaison linéaire de groupes : si un algorithme est à 20% dans le groupe G_1 et 80% dans le groupe G_2, son bruit sera tiré selon la loi $\mathcal{N}(0{,}2\,\epsilon_1 + 0{,}8\,\epsilon_2, {\sigma_a}^2)$. Par simplicité, nous supposons que les paramètres σ_g et σ_a sont communs à tous les groupes et algorithmes.

Nous comparons les différentes méthodes en calculant le *welfare* moyen obtenu. La formule du welfare est la suivante : si une méthode propose le candidat c_i, le welfare sera $w = (s(c_i) - s_{\min}) \,/\, (s_{\max} - s_{\min})$, où $s_{\max}$ (resp. $s_{\min}$) est le meilleur (resp. le pire) score parmi ceux des alternatives.

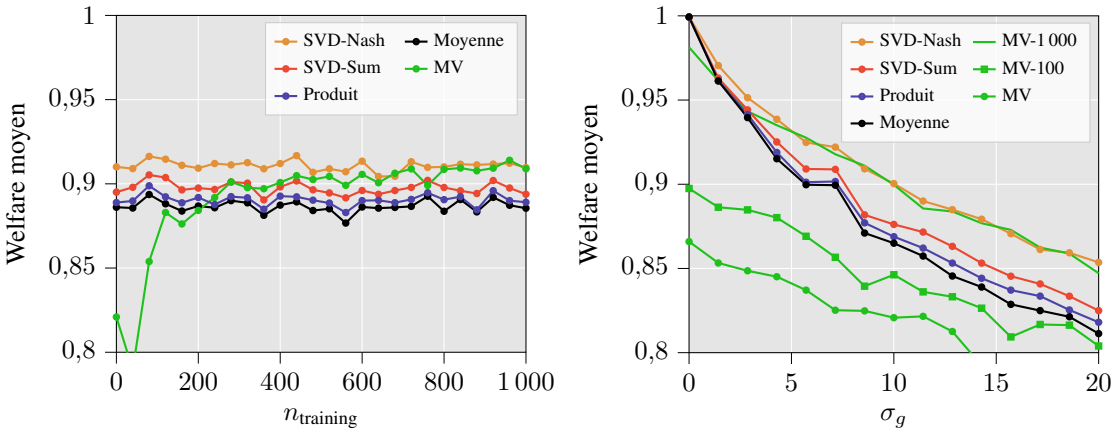
En plus des règles classiques (somme et produit), nous considérons les deux approches SVD précédemment définies et un maximum de vraisemblance (MV) basé sur l'hypothèse d'un bruit gaussien multivarié. Aussi bien les règles SVD que MV peuvent éventuellement bénéficier de données d'entraînement pour affiner leurs estimations.

La figure 1(a) montre le welfare en fonction du nombre de données d'entraînement aditionnelles utilisées. On observe que les méthodes SVD sont performantes et fonctionnent *à froid*, c'est-à-dire lorsque les seules données exploitables sont les prédictions des algorithmes pour les alternatives de « l'élection » en cours. À l'inverse, MV a un comportement erratique à froid, et ne devient intéressant qu'à partir de plusieurs centaines de données d'entraînement.

La figure 1(b) présente l'évolution du welfare moyen selon le bruit de groupe σ_g. On constate sans surprise que celui-ci diminue lorsque le bruit augmente, et que l'écart se creuse entre les méthodes classiques, et les méthodes à base de valeurs singulières, en particulier SVD-Nash, que seul un MV suffisamment entraîné arrive à égaler.

Globalement, notre méthode est 1 à 5 $\%$ meilleure en welfare moyen que les méthodes classiques. Elle est robuste lorsque les tailles des différents groupes sont totalement disproportionnées, ou en présence d'un bruit important, ce qui sont les points faibles des méthodes classiques d'agrégations, où il est généralement exigé de la diversification et de la précision [GVC05].

Nous avons testé nos méthodes sur d'autres modèles probabilistes théoriques, non montrés par souci de concision, par exemple en considérant que les éléments d'un même groupe avaient la même variance plutôt que la même moyenne, ou alors que pour chaque alternative, seulement certains groupes visent juste



(a) Impact de l'entraînement additionnel ($\sigma_g = 8$).

(b) Impact du bruit de groupe. n_{training} vaut 0, 100 ou 1 000 pour le maximum de vraisemblance, 0 pour les méthodes SVD.

FIGURE 1: Efficacité (welfare moyen) des différentes méthodes en fonction des paramètres. Chaque point est le résultat moyen de 1 000 expériences. Les algorithmes sont répartis de la manière suivante : 30 appartiennent à un groupe G_1, 2 à un groupe G_2, et 5 viennent d'une combinaison linéaire $0{,}3\,G_1 + 0{,}7\,G_2$. $m = 20$ et $\sigma_a = 0{,}5$. Le maximum de vraisemblance (MV) fait l'hypothèse d'un bruit gaussien multivarié.

tandis que d'autres attribuent un score au hasard. Dans tous les modèles, SVD-Sum et SVD-Nash font au moins aussi bien que les méthodes triviales (somme ou produit). La maximisation de la vraisemblance, très performante lorsque l'hypothèse sur la distribution du bruit est correcte, se dégrade lorsque cela n'est pas le cas.

5 Conclusion

Nous avons proposé une nouvelle méthode pour l'agrégation d'algorithmes dans les problèmes de décision. Cette méthode exploite les corrélations entre algorithmes afin d'améliorer la prise de décision. Elle améliore les techniques traditionnelles telles que la moyenne ou le produit, et ne nécessite pas de connaissance préalable sur les données, contrairement aux méthodes bayésienne ou de maximisation de la vraisemblance. Par la suite, nous pensons qu'il pourrait être intéressant d'utiliser la notion d'information mutuelle pour améliorer la prise en compte des relations entre algorithmes, ainsi que leurs degrés de performance individuels.

Références

[BCE+16] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. *Handbook of computational social choice*. Cambridge University Press, 2016.

[Eli93] Scott R Eliason. *Maximum likelihood estimation logic and practice*. Quantitative applications in the social sciences ; 96. SAGE, Newbury Park, [Calif.] ; London, 1993.

[FM20] Brandon Foubert and Nathalie Mitton. Sélection d'interface de communication dans les réseaux de capteurs multi-technologies. In *CoRes*, Lyon, France, September 2020.

[GVC05] P.M. Granitto, P.F. Verdes, and H.A. Ceccatto. Neural network ensembles : evaluation of aggregation algorithms. *Artificial Intelligence*, 163(2) :139–162, 2005.

[HKL11] I.B. Huang, J. Keisler, and I. Linkov. Multi-criteria decision analysis in environmental sciences : Ten years of applications and trends. *Science of The Total Environment*, 409(19), 2011.

[KN79] Mamoru Kaneko and Kenjiro Nakamura. The Nash social welfare function. *Econometrica : Journal of the Econometric Society*, pages 423–435, 1979.

[Nas50] John F. Nash. The bargaining problem. *Econometrica*, 18(2) :155–162, 1950.

[WRR03] Michael E. Wall, Andreas Rechtsteiner, and Luis M. Rocha. *Singular Value Decomposition and Principal Component Analysis*, pages 91–109. Springer US, Boston, MA, 2003.