
Projet Science des Données: Online learning with abstention

Yanis Amirou¹ Omar Khatib¹ Killian Mancheron¹ Guillaume Mendlevitch¹

Abstract

L'article présente une étude réalisée sur des algorithmes d'apprentissage en ligne avec abstention, une méthode pour laquelle les prédicteurs peuvent faire le choix de s'abstenir durant leur apprentissage. Tandis que les algorithmes classiques utilisent un ensemble d'entraînement entier ou des batchs, les méthodes de la famille de l'apprentissage en ligne ont pour but d'apprendre et de mettre à jour le meilleur prédicteur pas à pas à l'aide de données distribuées à la suite. C'est sur cette famille d'algorithmes que l'on applique la notion d'abstention. Celle-ci permet de considérer que les algorithmes possèdent le choix de s'abstenir plutôt que de prédire un label. On spécifie alors un coût généralement inférieur à une prédiction aléatoire et le label n'est pas révélé. Cela permet de minimiser les erreurs de prédictions.

1. Description et explication des objectifs

L'article s'intéresse à un cas de d'apprentissage en ligne avec abstention faisant intervenir une famille d'experts ξ de taille finie $[K]$ ou infinie. Selon le principe de l'abstention, un expert est défini comme étant composé de deux fonctions $\xi_j = (h_j, r_j)$, $j \in [K]$, où $h_j : X \rightarrow \mathbb{R}$ et $r_j : X \rightarrow \mathbb{R}$ sont respectivement des fonctions de prédiction et d'abstention. Cette dernière est généralement définie par $r_j = |h_j| - \theta$, et l'on considère que l'expert ξ_j s'abstient lorsque $r_j \leq 0$. L'objectif ici est de réaliser une classification binaire d'un ensemble de données $(x, y) \in X \times \{\pm 1\}$ ($X \in \mathbb{R}^d$), où à chaque itération $t \in [T]$, un point $x_t \in X$ est tiré aléatoirement. Ensuite, un expert est choisi (aléatoirement ou non) et il peut alors soit faire le choix de prédire la classe y_t de x_t ou bien s'abstenir. Dans le cas où l'expert s'abstient la classe y_t n'est pas dévoilée et la fonction de perte d'abstention prend systématiquement une valeur constante $c \in [0, 1]$. Aussi, la performance des algorithmes est déterminée par la valeur du regret, défini comme l'espérance de la différence de la somme des valeurs de la fonction de perte pour les experts choisis par l'algorithme et la somme des valeurs des fonctions de perte pour le meilleur expert possible, celui qui minimisait le mieux la loss. En-

fin, le principe d'apprentissage par abstention se base sur l'utilisation d'un graphe de retour appelé G_t^{ABS} (figure 1). Ce dernier est un graphe orienté, composé de l'ensemble des experts. Lorsqu'un expert est choisi et que celui-ci délivre une prédiction, la fonction de perte de tous les autres experts sera alors observée - on dit alors que tous les experts seront dans le voisinage $N_t(I_t)$ de l'expert ξ_{I_t} choisi. A l'inverse, si l'expert s'abstient, seules les fonctions de perte des autres experts s'abstenant seront révélées, la véritable classe du point x n'étant pas révélée. Ainsi l'algorithme ne pourra pas avoir une vision sur l'ensemble des prédicteurs et ne mettra à jour sa connaissance uniquement sur un sous-ensemble de prédicteurs.

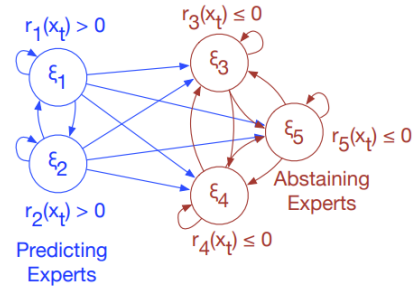


Figure 1. Graphe de retour G_t^{ABS} dans le cas de l'apprentissage en ligne avec abstention ($K=5$)

2. Le problème du biais

La première contribution que met en avant l'article est de majorer les regrets des algorithmes prédicteurs lorsqu'une politique non-stochastique est utilisée et dans les cas où la famille des experts est finie comme infinie. Dans une seconde partie, on s'intéresse à la stochastique de notre problème : à chaque pas de temps le graphe de retour est modifié puisqu'il varie selon les choix d'abstention des prédicteurs. Dans le cas de paramètres stochastiques, les auteurs proposent en premier lieu d'utiliser une politique UCB-NT appliquée généralement à un problème des k-bandits manchots. Il s'agit d'une extension de la méthode UCB, qui utilise le principe du graphe de retour variable dans le temps introduit section 1. Néanmoins, cette approche peut provoquer un biais selon l'ordre des données et les abstentions des prédicteurs. L'article introduit la notion du

biais qui pourrait se créer au cours des prédictions des experts en mettant en évidence un déséquilibre de fréquence de prédiction entre deux experts 1 et 2 au cas où on disposerait d'un ensemble de données uniformément distribué sur l'intervalle $[0,1]$ avec une densité légèrement prévalente dans la première moitié de l'intervalle. Le premier des experts est créé de façon à toujours s'abstenir sur l'intervalle $[\frac{1}{2}, 1]$ et prédire sur $[0, \frac{1}{2}]$ tandis que le second prédit quelque soit les données. Comme il y a une plus forte probabilité d'obtenir une donnée $x_i \leq \frac{1}{2}$, l'algorithme aura privilégié initialement le premier prédicteur comme l'illustre la figure 5 car sa perte estimée sera plus faible. De plus l'algorithme ne mettra alors pas la perte du deuxième prédicteur à jour si les données sont dans l'intervalle $[\frac{1}{2}, 1]$ puisque le premier prédicteur s'abstiendra. Ainsi, bien que la loss moyenne du deuxième prédicteur soit meilleure sur l'ensemble des données, l'algorithme UCB-NT choisira le mauvais algorithme.

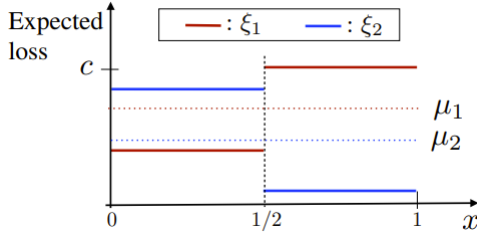


Figure 2. Illustration du problème du biais avec deux experts ξ_1 et ξ_2

3. Graphe à sous-ensembles

Néanmoins, l'article démontre que le phénomène de biais est annihilé et donc le regret de l'algorithme UCB-NT est borné dans le cas où le graphe de retour dépend de temps précédents et par implication s'il est fixe dans le temps et qu'il respecte une certaine logique. Les auteurs proposent donc, dans un premier temps, d'utiliser un graphe à sous-ensembles G^{SUB} : connaissant les fonctions d'abstention de chaque noeud, une arrête orientée de ξ_i vers ξ_j est définie uniquement si la région d'abstention d'un expert i est une sous-région de celle d'un expert j ($\{x \in X : r_i(x) \leq 0\} \subseteq \{x \in X : r_j(x) \leq 0\}$). Alors, si l'expert i est sélectionné à un instant t et s'abstient, l'expert j s'abstiendra lui aussi. Ce graphe fixe permet alors de parer au problème du biais mais ne bénéficie pas des avantages de l'apprentissage en ligne puisqu'une minorité des fonctions de perte sont mises à jour. Cependant, une autre démonstration contenue dans l'article, prouve que le regret est borné lorsque l'algorithme utilise uniquement les valeurs des itérations de 1 à $t-1$ - par conséquent, le graphe ne dépend plus de x_t . Une deuxième approche proposée par les auteurs est donc de compléter

le graphe à sous-ensembles en utilisant les observations jusqu'à $t-1$ pour obtenir un graphe plus informatif tout en limitant le regret: c'est le principe de l'algorithme UCB-GT.

4. Algorithme UCB-GT

On considère toujours l'inclusion suivante ($\{x \in X : r_i(x) \leq 0\} \subseteq \{x \in X : r_j(x) \leq 0\}$) mais l'inverse n'est pas réciproque, à savoir que les points d'abstention de ξ_j ne constituent pas un sous-ensemble des points d'abstention de ξ_i . La méthode explicitée dans l'article se base sur l'hypothèse que les régions d'abstention de ξ_i et ξ_j sont assez similaires pour considérer que si ξ_j s'abstient alors ξ_i s'abstiendra sûrement. Les auteurs introduisent alors $p_{j,i} = P[r_j(x) \leq 0 \cap r_i(x) > 0]$, la probabilité que l'expert j s'abstienne mais que l'expert i prédise. L'idée est qu'à chaque itération la probabilité P soit calculée ainsi qu'un seuil γ tels que si $p_{j,i}^{t-1}(x) \leq \gamma_i^{t-1}$, ξ_i sera considéré dans le voisinage de ξ_j . On note que la probabilité P et γ ne dépendent au temps t que des observations jusqu'au temps $t-1$. Cet algorithme permet donc d'avoir plus de liens entre experts par rapport au graphe G^{SUB} ce qui conduit à une mise à jour de la fonction de perte pour plus d'apprenants et donc à une meilleure valeur de regret pour UCB-GT par rapport à un algorithme utilisant un graphe statique.

5. Résultats expérimentaux

Afin de mettre en pratique ces méthodes, l'article réalise plusieurs comparaisons entre les différentes politiques UCB-GT, UCB-NT utilisant le graphe G^{SUB} , UCB et FS pour différentes valeurs de coût d'abstention. Ici FS est simplement un modèle optimiste irréaliste qui vient majorer la meilleure performance en renvoyant un regret minimum. Les expériences sont réalisées sur dix jeux de données pour chacun desquels 2 100 experts sont créés et chaque algorithme est lancé sur 10 000 épisodes. Les résultats sont obtenus pour trois coûts d'abstention différents, $c = 0.1, 0.2, 0.3$. Il s'avère alors que pour les dix jeux de données et peu importe la valeur du coût, UCB-GT surpasse UCB & UCB-NT et offre des performances similaires à FS, c'est-à-dire que le regret moyen de l'algorithme UCB-GT se rapproche de la meilleure valeur possible. Cette supériorité peut notamment s'expliquer par la présence d'environ 1 à 3 million(s) de liens entre experts pour UCB-GT soit au minimum cinq fois plus que pour UCB-NT qui en compte un nombre fixe de 200 000. Les résultats expérimentaux corroborent bien les démonstrations théoriques puisqu'augmenter le nombre de liens entre experts semble minimiser le regret.

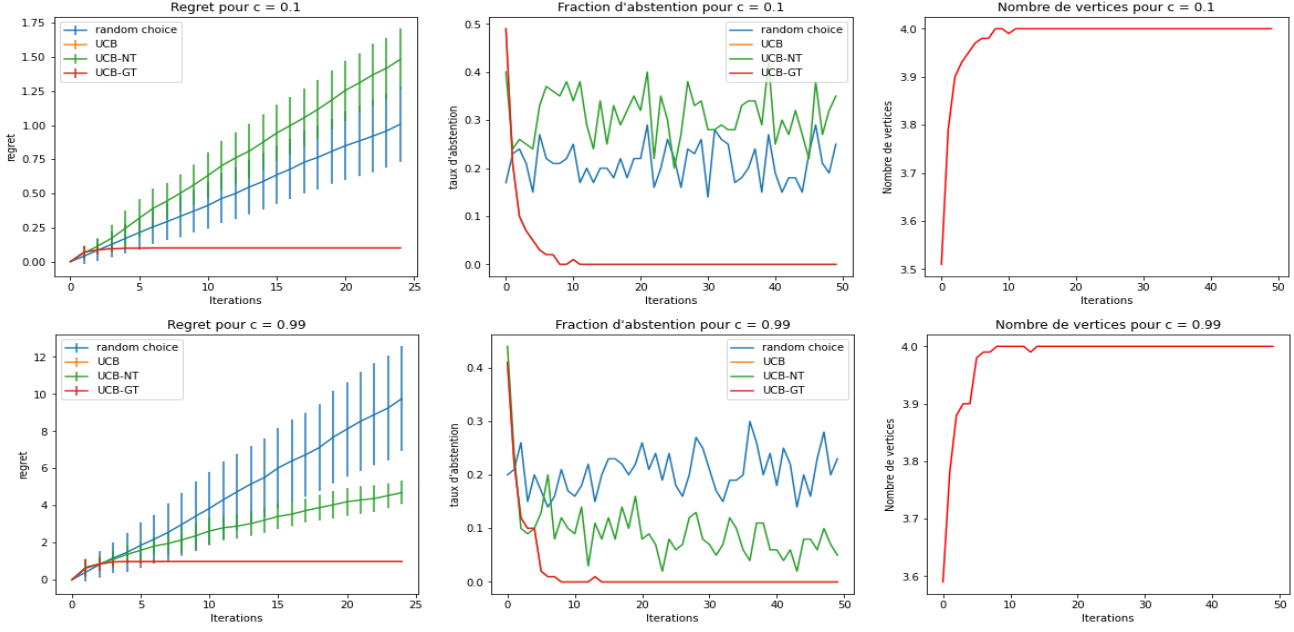


Figure 3. Les deux lignes représentent la simulation du biais présenté dans la première partie. Quatre algorithmes ont été comparés pour voir leur réaction face à un cas de biais : Uniforme, UCB, UCB-NT, UCB-GT (l'algorithme UCB classique n'est pas visible car égal à celui de UCB-GT). Afin d'éviter les résultats aléatoires, nous avons réalisés 100 expériences.

Partie II: Analyses de nos résultats expérimentaux

Afin de vérifier expérimentalement les résultats de l'article, nous étudions le comportement des algorithmes qui nous intéressent à savoir UCB, UCB-NT, UCB-GT ainsi qu'une politique de sélection de prédicteur aléatoire. Une première partie visant à montrer le phénomène de biais est d'abord abordée puis dans un second temps une comparaison similaire à ce que les auteurs ont réalisée entre les différents algorithmes est explicitée.

6. Mise en avant du phénomène du biais

Afin d'expérimenter sur le cas du biais énoncé dans le résumé, nous générons des données synthétiques uniformes sur $[0, 1]$ avec une légère concentration supérieure sur l'intervalle $[0, \frac{1}{2}]$. Le label à prédire est distribué selon la règle suivante : si $x > \frac{1}{2}$ on leur attribue le label 1, 0 sinon. Pour le premier prédicteur, celui qui s'abstient sur $[\frac{1}{2}, 1]$, il prédit si $x < \frac{3}{4}$ et s'abstient si $|x - \frac{3}{4}| < \frac{1}{4}$ permettant ainsi de garantir qu'il s'abstiendra toujours sur $[\frac{1}{2}, 1]$ comme l'indique l'énoncé et prédit avec une faible perte sur $[0, \frac{1}{2}]$. Pour le second prédicteur, celui qui prédit toujours, il prédit le label 0 si $x < \frac{1}{2}$, 1 sinon et ne s'abstient jamais permettant ainsi de garantir qu'il sera le prédicteur minimisant la perte moyenne. Nous utilisons UCB-NT et UCB-GT ainsi que UCB classique et le choix aléatoire de

prédicteur afin de comparer au mieux les résultats et de déterminer si un algorithme possède un regret sous-linéaire. Pour obtenir des résultats fiables, on réalise 100 expériences permettant ainsi de visualiser la moyenne et l'écart-type des résultats. La figure 3 permet de constater que UCB-NT est pire qu'un choix aléatoire de prédicteur; ceci est attendu puisqu'en effet le choix aléatoire alterne entre les deux prédicteurs possibles tandis que UCB-NT avec un graphe de feedback, qui varie à chaque pas de temps, favorise plus souvent le pire prédicteur. Avec des valeurs plus élevée de coût d'abstention (très proche de 1), on observe que l'algorithme UCB-NT reste inférieur à UCB-GT mais n'est cependant plus linéaire et on peut espérer qu'au bout d'un nombre élevé d'itération il finisse par choisir le meilleur prédicteur. Ce phénomène est encore plus visible sur le graphique représentant le taux d'abstention puisqu'on observe le caractère aléatoire du choix du premier prédicteur. UCB-GT et UCB classique choisissent très rapidement le deuxième prédicteur et possèdent le même comportement. Par définition des estimateurs, le graphe de retour possède trois ou quatre liaisons c'est d'ailleurs pour cela que l'on remarque que le nombre de liens entre les deux sommets du graphe de UCB-GT converge rapidement vers quatre. L'expérience a permis de mettre en avant le problème du biais et la résistance de UCB-GT à ce cas de figure ce dont n'avait pas réalisé l'article.

7. Comparaison des performances

Pour comparer nos quatre algorithmes, nous réalisons nos expériences sur deux classes du jeu de données CIFAR10 normalisé, sur lequel nous appliquons une Analyse en Composantes Principales pour ne garder que 26 composantes. Nous utilisons pour chaque algorithme 50 prédicteurs et réalisons une moyenne sur 5 épisodes de chacun 100 itérations. Nous définissons le coût d’abstention $c = 0.4$.

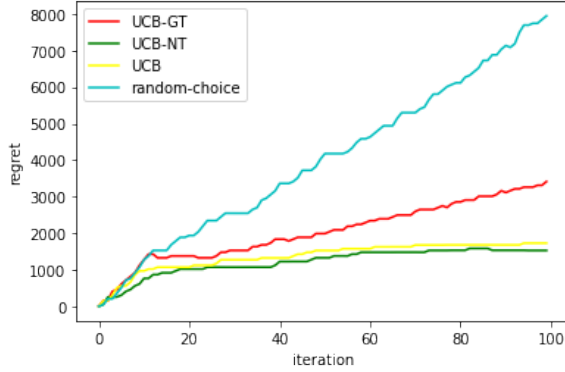


Figure 4. Comparaison des performances des algorithmes sur les classes 0 et 1 de CIFAR10 après sélection de 26 premières composantes principales (K=50, T=100, c=0.4)

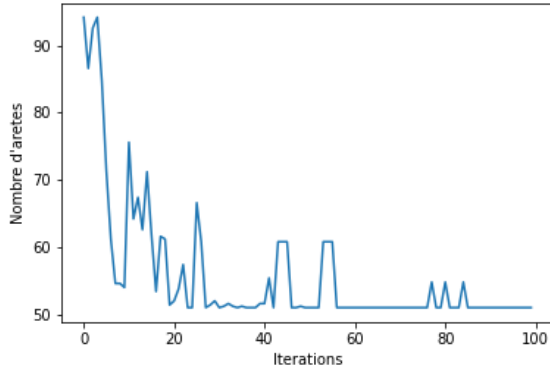


Figure 5. Nombre d’arêtes du graphe de UCB-GT (K=50, T=100, c=0.4)

En premier lieu, nous observons que toutes les méthodes UCB possèdent un regret sous-linéaire puisque leur courbe se situent toutes en dessous de la valeur moyenne du regret pour l’algorithme de choix aléatoire de prédicteur. En revanche, nous constatons qu’à l’inverse de l’article, notre algorithme UCB-GT ne semble pas être le plus performant. En effet, les méthodes UCB et UCB-NT semblent converger plus rapidement et tendent vers une valeur de regret inférieure. Cependant, ce résultat est à mettre en

perspective pour diverses raisons. Déjà, il est important de noter que nous n’utilisons pas un graphe fixe dans le temps comme le fait l’article mais bien le graphe de retour variable G_t^{ABS} pour l’algorithme UCB-NT. Ce dernier, bien qu’il puisse induire un biais (voir section 6), utilise un graphe variable dans le temps et ainsi, offre théoriquement de meilleures garanties pour le regret que la méthode utilisant G_t^{SUB} . De plus, les auteurs poussent plus loin les limites de l’expérience puisqu’ils utilisent environ 40 fois plus de prédicteurs que dans notre étude, et réalise 100 fois plus d’itérations.

8. Conclusion

L’article transpose intelligemment le problème des bandits manchots au concept d’apprentissage en ligne avec abstention. Néanmoins, bien que la partie centrée sur l’*adversarial setting* soit bien explicite, nous constatons l’absence d’expériences algorithmiques sur cette section. De même, nous constatons qu’aucune expérience sur la méthode UCB-NT utilisant le graphe de retour G_t^{ABS} n’a été menée dans les expériences générales de l’article ainsi que dans le cadre du phénomène du biais. Aussi, notre algorithme UCB-GT ne renvoie pas des performances similaires à ce qu’illustre l’article. Par ailleurs, nous manquons d’informations concernant l’initialisation ainsi que les problèmes liés aux indices. Etant donné que l’article propose des expérimentations couteuses, nous ne sommes pas en mesure de les répliquer correctement ce qui nous empêche de comparer nos résultats.

References

Cortes, C., DeSalvo, G., Gentile, C., Mohri, M., and Yang, S. Online learning with abstention. 80:1059–1067, 10–15 Jul 2018. URL <http://proceedings.mlr.press/v80/cortes18a.html>.