Journées SMAI MODE 2018

Apprentissage dans le cas bandit sur des jeux de potentiel

Amélie Héliou
Université Grenoble Alpes

Johanne Cohen
Université Paris Saclay

Panayotis Mertikopoulos
Université Grenoble Alpes

Résumé. Motivés par des défis actuels (réseaux, biologie, ...) nous avons étudié des algorithmes qui peuvent être appliqués à un grand nombre de joueurs qui ont une connaissance limitée du jeu. Dans de tels jeux, les algorithmes de non regret sont largement utilisés. Ces algorithmes garantissent que la différence entre les sommes dans le temps des revenus d'un joueur et des revenus de sa meilleure stratégie est sous-linéaire.
Les équilibres de Nash sont des états où aucun joueur ne bénéficierait de changer seul de stratégie. Des études récentes ont montré que la limite de la séquence de stratégies de certains types d'algorithmes de non regret peut être arbitrairement proche d'un équilibre de Nash avec une probabilité proche de 1.
Les équilibres de Nash peuvent-ils être la limite presque surement d'un algorithme de non regret ?
Nous répondons positivement à cette question en analysant l'algorithme Hedge qui a la propriété de non regret. Nous nous plaçons dans le cadre d'une information imparfaite (bandit), où les joueurs ont uniquement accès à une estimation du revenu rapporté par la stratégie pure qu'ils ont joué. Nous montrons que lorsque Hedge est appliqué à des jeux de potentiel génériques, la séquence de stratégies obtenues converge vers un équilibre de Nash.

Mots-clefs : Théorie des jeux, optimisation stochastique.

Introduction De nombreuses situations de la vie quotidienne peuvent être représentées sous la forme d'un jeu multijoueur : le choix de son itinéraire, le choix d'une fréquence pour un réseau de téléphonie ou bien encore l'orientation d'un nucléotide lors du repliement d'une molécule. Cependant dans ces jeux, les joueurs ont très peu d'information sur le jeu, ils ignorent le nombre de joueurs ainsi que les actions que peuvent choisir les autres joueurs. Les équilibres de Nash sont des équilibres de jeux multijoueur où chaque joueur ne peut pas augmenter ses gains en changeant seul sa stratégie. Les équilibres de Nash sont difficiles à calculer, il est alors intéressant de se demander s'ils peuvent être obtenus par des algorithmes répétés où chaque joueur adapte son action en fonction de ce qu'il observe.
Nous allons montrer que oui, en étudiant l'algorithme Hedge [2], qui est une variante des algorithmes à poids exponentiels, appliqué aux jeux de potentiel atomiques génériques [5].

Hedge est un algorithme de non regret, il garantit que la différence entre les sommes dans le temps des gains d'un joueur et des gains de sa meilleure stratégie est sous-linéaire par rapport au temps. Les algorithmes de non regret convergent vers un ensemble d'équilibres [3] qui est plus large que l'ensemble des équilibres de Nash. Kleinberg et al. [4] ont montré que la limite de la séquences des stratégies de certains types d'algorithmes de non regret peut être arbitrairement proche d'un équilibre de Nash avec une probabilité proche de 1. Cependant ces études considèrent que les joueurs connaissent le gain de chacune de leurs stratégies, au lieu de uniquement celui de la stratégie choisie.

Algorithme Nous étudions des jeux finis : un ensemble fini de joueurs $\mathcal{N} = \{1, ..., N\}$ qui ont chacun un ensemble fini de stratégies pures, $\mathcal{S}_i$, où l'indice i correspond au joueur. Chaque joueur a une fonction d'utilité, $u_i \equiv \prod_i \mathcal{S}_i \to \mathbb{R}$, qui détermine ses gains en fonction des stratégies de tous les joueurs. Chaque joueur peut également jouer une stratégie mixte, $x_i \in \Delta\mathcal{S}_i$, qui correspond à une probabilité de jouer chacune de ses stratégies pures. La fonction d'utilité de chaque joueur est étendue aux stratégies mixtes par multi-linéarité : $u_i(x) = \sum_{s_1 \in \mathcal{S}_1} \cdots \sum_{s_N \in \mathcal{S}_N} x_{1s_1} \cdots x_{Ns_N} u_i(s_1, \ldots, s_N)$.
On définit aussi le vecteur d'utilité qui correspond à l'utilité de chaque stratégie pure d'un joueur lorsque les autres joueurs jouent leur stratégie mixte, $v_i(x) = (u_i(x_1, \ldots, s_i, \ldots, x_N))_{s_i \in \mathcal{S}_i}$.
Les fonctions d'utilités sont telles que le jeu est un jeu de potentiel, c'est-à-dire qu'il existe une fonction de potentiel, $f \equiv \prod_i \mathcal{X}_i \to \mathbb{R}$, telle que : $u_i(x_i; x_{-i}) - u_i(x'_i; x_{-i}) = f_i(x_i; x_{-i}) - f_i(x'_i; x_{-i})$, où x_i et x'_i sont des stratégies mixtes du joueur i et x_{-i} représente l'ensemble des stratégies mixtes de tous les joueurs sauf i.
L'algorithme Hedge, décrit ci-dessous, consiste à mettre à jour deux vecteurs par joueur, sa stratégie mixte qui est aussi un vecteur de probabilités sur les stratégies pures et son vecteur de scores. À l'initialisation chaque joueur attribue un poids à chacune de ses stratégies de façon aléatoire. À chaque étape, chaque joueur calcule sa stratégie mixte par la formule des poids exponentiels $\Lambda(Y)$ mélangée à la distribution uniforme avec le facteur d'exploration ϵ :

$$X_i = \frac{\epsilon}{|\mathcal{S}_i|} + (1 - \epsilon)\frac{(\exp(Y_{is_i}))_{s_i \in \mathcal{S}_i}}{\sum_{s_i \in \mathcal{S}_i} \exp(Y_{is_i})}.$$

Ensuite il tire une action (stratégie pure) en fonction de la loi de probabilité donnée par sa stratégie mixte. Chaque joueur peut alors estimer son gain, selon l'estimateur bandit : $\hat{v}_i(n) = \left(\frac{\mathbb{1}_{s_i(n)=s_i}}{X_{is_i}} u_i(s(n))\right)_{s_i \in \mathcal{S}_i}$, où chaque stratégie reçoit 0 à part celle qui a été tirée par le joueur. Finalement, les joueurs mettent à jour leur vecteur de score $Y_i(n+1) \leftarrow Y_i(n) + \gamma_n \hat{v}_i(n)$.

Algorithm 1 ϵ-HEDGE en bandit

Require: séquences $\gamma_n > 0$ et $\epsilon_n \in [0, 1]$, les vecteurs initiaux $Y_i \in \mathbb{R}^{S_i}$, $i \in \mathcal{N}$.
1: **for** $n = 1, 2, ...$ **do**
2: **for** chaque joueur $i \in \mathcal{N}$ **do**
3: met à jour sa stratégie mixte : $X_i \leftarrow \epsilon_n/|\mathcal{S}_i| + (1 - \epsilon_n)\Lambda_i(Y_i)$;
4: choisit une action $s_i \sim X_i$;
5: calcule son estimateur bandit $\hat{v}_i(n)$;
6: met à jour ses scores : $Y_i \leftarrow Y_i + \gamma_n \hat{v}_i$;
7: **end for**
8: **end for**

Méthode Notre méthode d'analyse repose sur la méthode des équations différentielles ordinaires de Benaïm [1] et peut être divisée en quatre étapes principales. Tout d'abord, nous montrons que la suite des stratégies mixtes $(X(n))_{n\in\mathbb{N}}$ obtenues par l'algorithme 1 est une trajectoire pseudo asymptotique de la dynamique de réplicateur. Ensuite, nous montrons que la fonction de potentiel du jeu est une fonction de Lyapounov stricte de la dynamique. Puis, nous montrons que X converge vers les points stationnaires de la dynamique. Et enfin nous montrons que X converge vers les équilibres de Nash.

Resultats Le facteur d'exploration est essentiel pour utiliser l'estimateur bandit. En particulier, il ne peut pas être nul. En effet, nous avons besoin d'un borne à chaque étape sur $\hat{v}_i(n)$, et donc d'une valeur minimale non nulle pour X_{is_i}. Si ϵ est fixe nous montrons la convergence vers un équilibre de Nash ϵ-approximé.

Theorem 1 *Soit Γ un jeu de potentiel générique et supposons que l'algorithme 1 est joué avec l'estimateur bandit, un facteur d'exploration ϵ constant et strictement supérieur à 0 et une séquence de pas de la forme $\gamma_n \propto 1/n^\beta$ avec $\beta \in]0,1]$.*

1. *$X(n)$ converge presque surement vers un δ-équilibre de Γ avec $\delta \to_{\epsilon\to 0} 0$.*
2. *Si $X(n)$ converge vers un état ϵ-pur de la forme $x_i^* = \frac{\epsilon}{|\mathcal{S}_i|} + (1-\epsilon)e_{\hat{s}_i}$ alors $\hat{s}$ est un équilibre strict de Γ et la convergence est quasi exponentielle :*

$$X_{i\hat{s}_i}(n) \geq 1 - \epsilon - b\exp(-c\sum_{k=1}^{n}\gamma_k) \text{ avec } b,c>0.$$

Si ϵ décroit vers 0 nous montrons la convergence vers un équilibre de Nash du jeu.

Theorem 2 *Soit Γ un jeu de potentiel générique, supposons que l'algorithme 1 est joué avec l'estimateur bandit, une séquence de pas de la forme $\gamma_n \propto 1/n^\beta, \beta \in]1/2,1]$ et de facteurs d'exploration ϵ_n, telles que : $\lim_{n\to\infty}\epsilon_n = 0, \lim_{n\to\infty}\frac{\gamma_n}{\epsilon_n^2} = 0, \sum_{n=1}^{\infty}\frac{\gamma^2}{\epsilon_n} < \infty,$ et $\lim_{n\to\infty}\frac{\epsilon_n-\epsilon_{n+1}}{\gamma_n^2} = 0$. Alors $X(n)$ converge presque surement vers un équilibre de Nash de Γ.*

Références

[1] Michel Benaïm. Dynamics of stochastic approximation algorithms. *Séminaire de probabilités de Strasbourg*, 33, 1999.

[2] Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1):79–103, 1999.

[3] James Hannan. Approximation to Bayes risk in repeated play. In Melvin Dresher, Albert William Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games, Volume III*, volume 39 of *Annals of Mathematics Studies*, pages 97–139. Princeton University Press, Princeton, NJ, 1957.

[4] Robert Kleinberg, Georgios Piliouras, and Eva Tardos. Multiplicative updates outperform generic no-regret learning in congestion games. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 533–542. ACM, 2009.

[5] Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124 – 143, 1996.