Introduction au domaine de recherche Bandits et problèmes de décision séquentiels

Ludovic Schwartz

Octobre 2021

1 Introduction

Le modèle de décision séquentiel s'intéresse à un agent qui interagit avec un environnement. L'agent observe l'état d'un système, choisit une action, pour laquelle il reçoit une récompense(ou subit un coût), et l'état du système change et ceci de façon non nécessairement déterministe. En particulier l'apprentissage par renforcement s'intéresse à la création et à l'étude d'algorithmes qui maximise la récompense à long terme dans ce contexte séquentiel. On peut se référer à Puterman (2014) pour une introduction plus complète du sujet. Les bases modernes de ce domaine ont été posées dans Bellman (1954). On peut se référer à Sutton and Barto (2018) et à Bertsekas and Tsitsiklis (1996) pour comprendre les fondamentaux de l'apprentissage par renforcement et à Szepesvári (2010) pour un résumé des idées basiques et des algorithmes. L'algorithme UCRL(Upper Confidence Reinforcement Learning) et l'analyse de ses performances et de son regret vient de Jaksch, Ortner, and Auer (2010) et de Auer, Jaksch, and Ortner (2009).

On peut voir ces problématiques comme une extension d'un d'un domaine plus ancien, l'étude des problèmes de bandits. Introduit dans une version simplifiée dans Thompson (1933), et popularisé par plusieurs pionniers dont Herbert Robbins et Herman Chernoff(voir Robbins (1952) et Bather and Chernoff (1967)), on peut trouver de nombreux livres plus complets sur le sujet comme Bubeck and Cesa-Bianchi (2012) ou Lattimore and Szepesvári (2020). Ces domaines sont en plein essor aujourd'hui aussi bien dans la théorie que dans les applications pratiques. Par exemple alphago dans Silver et al. (2016) est une des applications qui utilise ces technologies.

On trouve des applications en informatique, en biologie, en médecine, en économie des idées que l'on va développer dans les paragraphes suivants. Les bandits et les problèmes de décision séquentiels font appel à plusieurs domaines des mathématiques, on y trouve des probabilités, de la théorie des martingales, des bornes de concentration, de l'optimisation, de la théorie des jeux, de la théorie de l'information. On va d'abord s'intéresser en détails aux bandits et aux concepts clés qui y apparaissent. Ces concepts apparaissent aussi dans l'apprentissage par renforcement. On donnera ici seulement les idées de preuves.

On verra ensuite comment certains résultats évoluent dans des contextes plus généraux.

2 Bandits Stochastiques

Cette section est basée sur la section équivalente de "Bandits algorithms" de Lattimore and Szepesvári (2020). On va s'intéresser aux définitions et aux principales bornes de regret.

2.1 Définition générale

Un bandit stochastique est la donnée d'un ensemble d'actions $\mathcal{A}$ et d'une collection de lois de probabilités $\{\nu_a : a \in \mathcal{A}\}$. On appellera indifféremment action ou bras un élément de $\mathcal{A}$. Un agent va intéragir de façon séquentielle avec un environnement, à chaque instant t, l'agent va choisir une action $a_t \in \mathcal{A}$ et recevoir une récompense $X_t \in \mathbb{R} \sim \nu_{a_t}$. Cela induit une mesure de probabilité sur la séquence $A_1, X_1, \ldots, A_n, X_n$. On appelle n l'horizon.

Les conditions suivantes doivent être satisfaites :

- $X_t | H_t \cup A_t \sim \nu_{A_t}$. Cela signifie que la récompense est tirée selon la loi associée au bras sélectionné par l'agent.
- La loi conditionelle $A_t | H_t$ est $\pi_t(\cdot | H_t)$ où $\pi_1, \pi_2, \ldots$ est une séquence de noyau de probabilités qui caractérise l'agent. On appelle π la stratégie de l'agent. En particulier, la stratégie de l'agent à l'instant t ne dépends que du passé et l'agent ne peux pas utiliser les observations futures pour ses choix présents.

Remarque. *En général, on ne mentionne pas les espaces de probabilités sur lesquels on travaille, mais il existe un modèle canonique des bandits stochastiques que l'on peut choisir lorsqu'il est utile de fixer l'espace de probabilité.*

Remarque. *L'agent ne reçoit que la récompense du bras qu'il a tiré. En particulier, il peut lui être nécessaire d'effectuer de "mauvaises" actions pour obtenir de l'information sur son environnement.*

2.2 Optimisation

On définit la récompense totale de l'agent

$$S_n \stackrel{\text{def}}{=} \sum_{t=1}^{n} X_t \tag{1}$$

L'objectif de l'agent est de maximiser cette quantité aléatoire qui dépends de l'environnement. Plusieurs questions se posent alors ici :

- Quelle est la valeur de n pour laquelle on cherche à maximiser cette quantité ?

- La récompense totale est stochastique, comment souhaite-t-on contrôler cette quantité ? En espérance, avec haute probabilité ?
- Les distributions des différentes actions sont inconnues de l'agent.

La résultats obtenus pour un horizon fini connu sont souvents adaptables à un horizon connus moyennant quelques facteurs dans les bornes obtenues.

Remarque. *On peut aussi se placer dans un cadre où l'agent subit une perte pour chaque action qu'il entreprends et l'on cherche alors à minimiser la perte totale subie par l'agent.*

Par défaut, on va s'intéresser aux bornes en espérance mais il est important de noter qu'on peut aussi s'intéresser à la variance et au comportement plus précis de la distribution de la récompense.

2.3 Environnement

L'agent n'a pas accès à $\nu \stackrel{\text{def}}{=} (\nu_a : a \in \mathcal{A})$, en particulier une stratégie très efficace sur une première instance de ν peut très mal se comporter sur une autre. En général, l'agent dispose d'une information partielle sur ν que l'on représente en définissant un ensemble $\mathcal{E}$ pour lequel $\nu \in \mathcal{E}$. On appelle cet ensemble la classe de l'environnement. On peut distinguer deux cas principaux :

2.3.1 Bandits non structurés

Si $\mathcal{A}$ est fini et qu'il existe un une famille $(\mathcal{E}_a)_{a\in\mathcal{A}}$ telle que

$$\mathcal{E} = \{\nu = (\nu_a)_{a\in\mathcal{A}} : \nu_a \in \mathcal{E}_a, \forall a \in \mathcal{A}\}$$

on dit que la classe de l'environnement est non structurée. Cela revient à dire que

$$\mathcal{E} = \bigotimes_{a\in\mathcal{A}} \mathcal{E}_a$$

En particulier, en jouant l'action a, l'agent n'obtient aucune information sur $\{\nu_b, b \neq a\}$

Remarque. *La classe de l'environnement peut être paramétrique(par exemple les bandits Gaussiens à k bras de variance 1 et de moyenne $\mu \in [0,1]^k$) ou non.*

Remarque. *Si $\mathcal{A}$ était infini, on ne pourrait jamais avoir de l'information sur toutes actions possibles et il serait inutile de chercher à contrôler la récompense reçue par l'agent.*

En général, on cherche à contrôler la performance d'un algorithme sur une classe d'environnement, plus cette classe est large, plus cette tâche est difficile.

2.3.2 Bandits structurés

Dans le cas contraire, la classe de l'environnement est dite structurée. L'un des exemples les plus importants est celui des bandits linéaires stochastiques. Dans le cas gaussien de variance 1, On se donne un ensemble $\mathcal{A} \subset \mathbb{R}^d$, on définit

$$\forall \theta \in \mathbb{R}^d, \quad \nu_\theta \stackrel{\text{def}}{=} (\mathcal{N}(\langle a, \theta \rangle, 1) : a \in \mathcal{A})$$

et

$$\mathcal{E} = \{\nu_\theta : \theta \in \mathbb{R}^d\}$$

Il existe alors des algorithmes qui contrôlent la récompense reçue par l'agent alors que l'espace des actions est infini. Voir Dani, Hayes, and Kakade (2008) pour une analyse plus complète.

Remarque. *On parle d'un problème de bandits $\alpha - sous - gaussien$ si la classe de l'environnement est restreinte à des variables $\alpha - sous - gaussienne$*

2.4 Regret

Maintenant que l'on dispose d'une classe d'environnement, on cherche à contrôler la performance de notre agent et de sa stratégie sur cette classe. On peut alors définir la meilleur stratégie pour chaque instance de ν et comparer notre agent à cette stratégie (qui est inconnue de l'agent). On définit $\mu_a \stackrel{\text{def}}{=} \mathbb{E}[\nu_a]$,

$$\mu^* \stackrel{\text{def}}{=} \max_{a \in \mathcal{A}} \mu_a \tag{2}$$

et

$$a^* \stackrel{\text{def}}{=} \arg\max_{a \in \mathcal{A}} \mu_a \tag{3}$$

(On se restreint aux cas où ces quantités sont bien définies. a^* est alors l'une des meilleurs actions et la récompense moyenne que reçoit le joueur en l'actionnant est μ^*). On définit alors le regret

$$R_n(\pi, \nu) = n\mu^*(\nu) - \mathbb{E}\left[\sum_{t=1}^{n} X_t\right] \tag{4}$$

Le regret compare notre agent à la meilleure stratégie sur l'instance ν du problème de bandits. En particulier minimiser le regret est équivalent à maximiser la récompense totale. Les propriétés suivantes sont vérifiées :

- $R_n(\pi, \nu) \geq 0, \quad \forall \pi \in \Pi$
- Si la stratégie π choisit toujours $A_t \in \arg\max_{a \in \mathcal{A}} \mu_a$, alors $R_n(\pi, \nu) = 0$
- Si $R_n(\pi, \mu) = 0$, alors $\mathbb{P}[\mu_{A_t} = \mu^*] = 1$

En général, on cherche à ce que le regret soit sous-linéaire:

$$\forall \nu \in \mathcal{E}, \quad \lim_{n\to\infty} \frac{R_n(\pi, \nu)}{n} = 0$$

Sous certaines hypothèses, il est possible d'avoir des bornes beaucoup plus précises du regret. On définit $\Delta_a(\nu) \overset{\text{def}}{=} \mu^*(\nu) - \mu_a(\nu)$ le regret instantané du bras a et

$$N_a(t) \overset{\text{def}}{=} \sum_{s=1}^{t} \mathbb{1}_{\{A_t = a\}}$$

le nombre de fois où l'action a a été sélectionnée jusqu'au temps t.

On peut alors décomposer le regret selon ces nouvelles variables et l'équation suivante est vérifiée :

$$R_n = \sum_{a\in\mathcal{A}} \Delta_a \mathbb{E}[N_a(n)] \tag{5}$$

3 Algorithmes et Bornes

On se place dans le cadre des bandits stochastiques où la classe d'environnement est non structurée, en particulier $\mathcal{A}$ est fini et sans perte de généralité $\mathcal{A} = [k]$ pour un entier naturel $k > 1$. Considérons l'algorithme naïf suivant :

- Pendant les rounds $t = 1, \ldots, t = k$, on tire l'action $a_t = t$ et on obtient une récompense $X_t \sim \nu_t$ et une estimation de la moyenne empirique de la récompense pour chaque action $\hat{\mu}_a = X_a$.
- A chaque round $t > k$, on sélectionne $a_t \in \arg\max_{a\in\mathcal{A}} \hat{\mu}_a$, on obtient une nouvelle récompense $X_t \sim \nu_{a_t}$ et on mets à jour $\hat{\mu}_{a_t}$

Si l'on considère un problème de bandits stochastiques avec uniquement deux bras dont les récompenses suivent des lois de Bernoulli de paramètre $p_1 < p_2$, alors avec une probabilité $p_1(1-p_2)$, on obtient 1 et 0 aux deux premiers lancers, la première moyenne empirique du premier bras est 1, et celle du deuxième bras est 0, on ne tire alors plus que le premier bras qui est sous optimal. En effet, la moyenne empirique du premier bras restera toujours positive et celle du deuxième bras nulle. En particulier, le regret est linéaire, ce que l'on souhaite éviter.

3.1 ETC

On s'intéresse ici à l'algorithme ETC("explore then commit", explorer puis s'engager) qui va d'abord obtenir une première estimation de la valeur de chaque bras et ensuite sélectionner uniquement la meilleure action selon cette estimation. C'est une version plus générale de l'algorithme précédemment décrit. On définit à chaque instant t l'estimateur empirique de la moyenne de la récompense de l'action a :

$$\hat{\mu}_a(t) \overset{\text{def}}{=} \frac{1}{N_a(t)} \sum_{s=1}^{t} \mathbb{1}_{\{A_s = a\}} X_s$$

Cette quantité est bien définie dès qu'on a tiré l'action a au moins une fois. L'algorithme ETC est le suivant :

- On choisit un entier m
- Pendant les rounds $t = 1, \ldots, t = mk$, on choisit l'action $a_t = t \mod k + 1$
- Pendant les rounds $t > mk$, on sélectionne $a_t \in \arg\max_{a \in \mathcal{A}} \hat{\mu}_a$

On obtient alors trivialement la décomposition suivante du regret :

$$\begin{aligned} R_n = & \sum_{t=1}^{mk} \Delta_{a_t} + \sum_{t=mk+1}^{n} \Delta_{a_t} \\ = & \; m \cdot \sum_{a \in \mathcal{A}} \Delta_a + \sum_{t=mk+1}^{n} \Delta_{a_t} \end{aligned}$$

En particulier le premier terme est d'autant plus grand que m est grand. Si on tire beaucoup une action sous-optimale après la phase d'exploration, ce qui est d'autant plus probable qu'on a peu exploré et que m est petit, alors le deuxième terme devient grand. On a donc un compromis à trouver entre exploration et exploitation. On ne doit pas trop explorer car cela engendre du regret, mais suffisament pour limiter la probabilité de se tromper sur l'action optimale. Plus précisément, on peut obtenir la borne suivante :

Theorème 1. *Si ETC intéragit avec des bandits 1-sous-gaussien, et que* $1 \leq m \leq n/k$*, alors*

$$m \cdot \sum_{a \in \mathcal{A}} \Delta_a + \sum_{a \in \mathcal{A}} \Delta_a \exp\left(-\frac{m\Delta_a^2}{4}\right) \tag{6}$$

3.2 UCB

On va s'intéresser avec plus de détails à l'algorithme UCB (Upper Confidence Bounds) qui fonctionne selon le principe d'optimisme face à l'incertitude. L'utilisation des intervalles de confiance et l'idée de l'optimisme sont introduits dans Lai and Robbins (1985), et la première version d'UCB est décrite dans Lai (1987). Il existe différentes versions d'UCB, selon les hypothèses faites sur le problème de bandit sous-jacent.

On définit pour chaque action et pour tout réel $\delta > 0$ les quantités suivantes :

$$\hat{\mu}_a(t) \stackrel{\text{def}}{=} \frac{1}{N_a(t-1)} \sum_{s=1}^{t-1} \mathbb{1}_{\{a_s = a\}} X_s$$

$$UCB_a(t-1, \delta) = \hat{\mu}_a(t) + \sqrt{\frac{2\log(1/\delta)}{N_a(t-1)}}$$

L'algorithme est le suivant :

- Pour chaque round $t = 1, \ldots, k$ on choisit l'action $a_t = t$
- Pour chaque round $t = k+1, \ldots, n$:
- On choisit l'action $a_t \in \arg\max_{a \in \mathcal{A}} UCB_a(t-1, \delta)$
- On mets à jour $UCB_{a_t}(t, \delta)$

Essentiellement, on donne un "bonus" d'exploration à l'estimation empirique de la valeur de chacune des actions. Ce bonus est d'autant plus grand que les actions ont été peu explorées. C'est cela qu'on appelle le principe d'optimisme face à l'incertitude : quand la valeur d'une action est mal estimée, on va être optimiste et la tester dans l'espoir qu'elle soit bonne. Cela évite d'ignorer des actions sur lesquelles on a peu d'informations et pousse l'algorithme à explorer les actions peu utilisées, d'autant plus si elles sont prometteuses, c'est à dire que leur moyenne empirique est proche de la meilleure moyenne empirique trouvée jusque là. En particulier, on cherche à sélectionner une action si sa moyenne empirique est élevée ou si elle a peu été explorée, et la quantité UCB permet d'obtenir ce comportement.

En particulier, ce choix de bonus s'explique par la propriété suivante :

Proposition 1. *On considère une suite $(X_s)_{s=1}^t$ de variables aléatoires indépendantes de même loi et d'espérance μ 1-sous-gaussienne, et on définit*

$$\hat{\mu} \overset{def}{=} \frac{1}{t} \sum_{s=1}^{t} X_s$$

alors

$$\mathbb{P}\left(\mu \geq \hat{\mu} + \sqrt{\frac{2\log(1/\delta)}{n}}\right) \leq \delta \tag{7}$$

On dispose d'un intervalle de confiance supérieur pour toute suite de variable aléatoire iid et sous-gaussienne.

Remarque. *Dans l'algorithme d'UCB, le nombre de termes présents dans la moyenne empirique est stochastique.*

Les avantages d'UCB sont les suivants :

- Il est agnostique aux regrets instantannés des actions et peut être rendu agnostique à l'horizon.
- Il est efficace pour un grand nombre d'actions.

Remarque. *L'algorithme spécifié ici est utilisé pour résoudre des problèmes de bandits stochastiques non structurées, mais il existe des variations basées sur le même principe d'optimisme face à l'incertitude et sur l'utilisation d'intervalles de confiance pour s'attaquer à une grande quantité de problèmes comme les bandits stochastiques linéaires, les bandits contextuels, et l'apprentissage par renforcement.*

3.3 Borne de regret

3.3.1 Bornes supérieures

On va donner ici deux bornes de regret pour UCB, l'une avec un horizon fini, l'autre avec un horizon infini.

Theorème 2. *Considérons l'algorithme défini précédemment sur un problème de bandits stochastiques avec k actions et où les récompenses sont 1-subgaussiennes. Alors pour tout horizon n, si $\delta = 1/n^2$, on a*

$$R_n \leq 3\sum_{i=1} \Delta_i + \sum_{i:\Delta_i>0} \frac{16log(n)}{\Delta_i} \tag{8}$$

Theorème 3. *Considérons l'algorithme défini précédemment sur un problème de bandits stochastiques avec k actions et où les récompenses sont 1-subgaussiennes. Le choix de $\delta_t = \frac{1}{1+t\log^2(t)}$ vérifie la borne de regret suivante :*

$$R_n \leq C \sum_{i:\Delta_i>0} \left(\Delta_i + \frac{\log(n)}{n}\right) \tag{9}$$

où C est une constante universelle.

On peut se référer à Lattimore and Szepesvári (2020) pour une preuve complète de ces théorèmes. On va ici exposer l'étape clé de la preuve. Essentiellement, on a la décomposition suivante du regret par l'équation 5:

$$R_n = \sum_{a\in\mathcal{A}} \Delta_a \mathbb{E}[N_a(n)]$$

Il suffit donc de contrôler $\mathbb{E}[N_a(n)]$ pour contrôler le regret d'UCB. Sans perte de généralité, $a^* = 1$ On se place à un instant $t > k$, alors l'une de ces trois conditions est vérifiée :

1. $\mu_1 \geq \min_{t\in[n]} UCB_1(t,\delta)$
2. $\mu_1 \leq \hat{\mu}_{a,u_a} + \sqrt{\frac{2}{u_a}\log(\frac{1}{\delta})}$
3. $N_a(t) \leq u_a$

où u_a est un nombre entier choisi plus tard dans la preuve et $\hat{\mu}_{a,u_a}$ est l'estimation empirique de la valeur du bras a après qu'il est été tiré u_a fois. Appellons I_1, a, I_2, a, I_3, a ces évènements. On a alors :

$$\begin{aligned}
R_n &= \sum_{a\in\mathcal{A}} \Delta_a \mathbb{E}[N_a(n)] \\
&= \sum_{a\in\mathcal{A}} \Delta_a \mathbb{E}[N_a(n)|I_{1,a} \cup I_{2,a} \cup I_{3,a}] \\
&\leq \sum_{a\in\mathcal{A}} \Delta_a \mathbb{E}[N_a(n)|I_{1,a}] + \sum_{a\in\mathcal{A}} \Delta_a \mathbb{E}[N_a(n)|I_{2,a}] + \sum_{a\in\mathcal{A}} \Delta_a \mathbb{E}[N_a(n)|I_{3,a}]
\end{aligned}$$

Le premier cas corresponds à la quantité UCB qui est inférieure à la vraie valeur μ_1, le deuxième cas corresponds à l'estimation d'un autre bras qui est supérieure à la vraie valeur de μ_1. Ces deux évènements sont de faible probabilités d'après les estimations précédentes et on peut contrôler leur contribution au regret. Le terme général de la troisième somme est borné de façon déterministe par u_a. En particulier, le choix

$$u_a = \left\lceil \frac{8 \log(1/\delta)}{\Delta_i^2} \right\rceil$$

donne la borne de regret souhaité.

3.3.2 Borne inférieure

On définit pour une stratégie π le **regret dans le pire des cas** sur un ensemble $\mathcal{E}$ de bandits stochastiques

$$R_n(\pi, \mathcal{E}) \stackrel{\text{def}}{=} \sup_{\nu \in \mathcal{E}} R_n(\pi, \nu) \tag{10}$$

On définit alors le regret MinMax comme

$$R_n^*(\mathcal{E}) \stackrel{\text{def}}{=} \inf_{\pi \in \Pi} R_n(\pi, \mathcal{E}) = \inf_{\pi \in \Pi} \sup_{\mu \in \mathcal{E}} R_n(\pi, \mu) \tag{11}$$

Le regret de n'importe quel agorithme sera toujours supérieur au regret MinMax. En particulier, cela nous permet de savoir si on a le bon ordre de grandeur du regret.

Theorème 4. *Soit $\mathcal{E}^k$ l'ensemble des bandits à k bras Gaussien de variance 1 et de moyenne $\nu \in [0,1]^k$. Soit $n \geq k-1$ alors, pour toute stratégie π, il existe un vecteur ν tel que*

$$R_n(\pi, \mu_\nu) \geq \frac{1}{27} \sqrt{(k-1)n} \tag{12}$$

en particulier

$$R_n^*(\mathcal{E}^k) \geq \frac{1}{27} \sqrt{(k-1)n} \tag{13}$$

On peut trouver une preuve complète de cette borne dans le chapitre 15 de Lattimore and Szepesvári (2020). On explicite ici le début de la preuve. On se donne une stratégie π fixée et un réel $\Delta \in [0, 1/2]$. On considère un problème de bandits gaussien de variance unitaire et de vecteur moyen $\mu = (\Delta, 0, 0, \ldots, 0)$. Cet environnement et cette stratégie nous donne une mesure de probabilité P_μ sur les événements de ce problème de bandits(explicitement sur les séquences d'actions et de récompense jusqu'à l'horizon n). Soit alors

$$i \stackrel{\text{def}}{=} \arg\min_{j>1} \mathbb{E}_\mu[N_j(n)]$$

en particulier, $\mathbb{E}_\mu[N_j(n)] \leq n/(k-1)$ On considère maintenant un nouveau problème de bandit gaussien avec un vecteur moyen

$$\mu' = (\Delta, 0, 0, \ldots, 0, 2\Delta, 0, \ldots, 0)$$

Explicitement, $\mu'_i = 2\Delta$. En particulier

$$R_n(\pi, \nu_\mu) \geq \mathbb{P}_\mu \left[N_1(n) \leq n/2\right] \frac{n\Delta}{2}$$

et

$$R_n(\pi, \nu_{\mu'}) \geq \mathbb{P}_{\mu'} \left[N_1(n) > n/2\right] \frac{n\Delta}{2}$$

Essentiellement, notre stratégie va difficilement contrôler en même temps son regret dans ces deux problèmes de bandits. On choisit le bras le plus défavorisé par la stratégie dans le premier problème de bandit et on en fait le bras optimal du second problème. On obtient la borne voulue en faisant intervenir l'entropie relative des deux problèmes de bandits(voir de nouveau Lattimore and Szepesvári (2020) pour une preuve complète.

4 Extensions

On peut s'intéresser à des problèmes de décisions séquentiels plus généraux. Dans le cas des bandits stochastiques, le comportement des différents bras est le même à chaque instant t. On peut s'intéresser à une configuration qui évolue au fil du temps. On rajoute alors un "contexte" qui est accessible à notre agent qui va devoir prendre la meilleure décision en s'aidant aussi du contexte. Par exemple, une des applications des bandits est la publicité en ligne, le site internet doit choisir parmi une liste de publicités laquelle il veut montrer à un utilisateur et reçoit une récompense si l'utilisateur clique sur cette dernière. Les bandits stochastiques telles qu'on les a décrit jusqu'ici permettent d'évaluer la meilleure publicité selon l'hypothèse que les utilisateurs sont indépendants et identiquement distribués (dans leurs habitudes de clic). En pratique, on dispose de données sur les utilisateurs qui permettent de faire une décision mieux informée et adaptée à l'utilisateur.

4.1 Bandits contextuels

On considère le protocole suivant qui se passe sur n rounds. A chaque round t, l'environnement fournit un context $x_t \in \mathcal{X}$ et une fonction de perte $l_t : \mathcal{A} \to [0, 1]$ où $\mathcal{A}$ est l'ensemble des actions que peut prendre l'agent. L'agent choisit une action a_t et subit une perte $l_t(x_t)$. On suppose que chaque fonction de perte est tirée indépendamment selon une distribution fixée $\mathbb{P}_{l_t}(\cdot|x_t)$ où $\mathbb{P}_{l_1} \dots \mathbb{P}_{l_t}$ sont sélectionnés avant le début des rounds. On suppose que l'agent a accès à une classe de fonction $\mathcal{F} \subset (\mathcal{X} \times \mathcal{A}) \to [0, 1]$.

Hypothèse 1. *On suppose qu'il existe une fonction $f^* \in \mathcal{F}$ qui vérifie*

$$\forall t, a \quad f^*(x, a) = \mathbb{E}[l_t(a)|x_t = x] \tag{14}$$

On définit alors

$$\forall f \in \mathcal{F}, \pi_f(x) \stackrel{\text{def}}{=} \arg\min_{a \in \mathcal{A}} f(x, a)$$

On peut alors définir le regret comme

$$Reg_{CB}(n) \overset{\text{def}}{=} \sum_{t=1}^{n} l_t(a_t) - l_t(\pi^*(x_t)) \tag{15}$$

où $\pi^* \overset{\text{def}}{=} \pi_{f^*}$

Remarque. *Notre agent peut recevoir une récompense qu'il cherche à maximiser ou subir une perte qu'il cherche à minimiser. Ces deux visions sont équivalentes et on prends ici la notation utilisée par l'article.*

On présente ici l'algorithme SquareCB proposé par Foster and Rakhlin (2020) et sa borne de regret.

On suppose qu'on dispose d'un oracle de régression en ligne. A chaque instant t, cet oracle que l'on appelle SqAlg reçoit un couple (x_t, a_t) et effectue une prédiction $\hat{y}_t$ et reçoit la vraie valeur y_t. On suppose que l'on contrôle le regret de cet oracle selon l'équation suivante :

$$\sum_{t=1}^{n} (\hat{y}_t - y_t)^2 - \inf_{f \in \mathcal{F}} \sum_{t=1}^{n} (f(x_t, a_t) - y_t)^2 \leq Reg_{Sq}(n) \tag{16}$$

L'algorithme SquareCB est le suivant :

On se donne deux paramètres $\gamma > 0$ et $\mu > 0$ ainsi qu'un oracle de régression SqAlg

- Pour t allant de 1 à T :
- On reçoit le contexte x_t
- On calcule pour chaque action $a \in \mathcal{A}$ la quantité $\hat{y}_{t,a} \overset{\text{def}}{=} \hat{y}_t(x_t, a)$.
- Soit $b_t \overset{\text{def}}{=} \arg\min_{a \in \mathcal{A}} \hat{y}_t, a$ la meilleure action selon l'oracle.
- Pour $a \neq b_t$, on définit $p_{t,a} \overset{\text{def}}{=} \frac{1}{\mu + \gamma(\hat{y}_t, a - \hat{y}_t, b_t)}$ et $p_{t,b_t} \overset{\text{def}}{=} 1 - \sum_{a \neq b_t} p_{t,a}$
- On tire $a_t \sim p_t$ et on observe une perte $l_t(a_t)$
- On mets à jour notre oracle avec l'exemple $((x_t, a_t), l_t(a_t)$

Alors, on a la borne de regret suivante :

Theorème 5. *L'algorithme SquareCB invoqué avec un oracle de régression en ligne avec un regret $Reg_{Sq}(n)$ et les paramètres $\mu = K$ et $\gamma = \sqrt{KT/(Reg_{Sq}(n) + \log(2\delta^{-1})}$ vérifie*

$$\forall \delta > 0, \mathbb{P}\left[Reg_{CB}(n) > 4\sqrt{KT \cdot Reg_{Sq}(n)} + 8\sqrt{KT \log(2\delta^{-1})} \right] < \delta \tag{17}$$

On retrouve ici la dépendance en $\sqrt{KT}$, et on peut également prouver une borne inférieure de regret qui montre que l'algorithme SquareCB est optimal (Il atteint la borne de regret MinMax de $\mathcal{F}$.

4.2 Processus de décision Markoviens et Apprentissage par renforcement

Par rapport aux bandits stochastiques, les bandits contextuels nous permettent d'intégrer une information supplémentaire à travers le contexte reçu à chaque round. Cependant, il n'existe pas de dépendance à l'historique des décisions. Pour s'intéresser à à des problèmes où la trajectoire de l'agent est prise en compte, on s'intéresse aux Processus de décisions Markoviens. Un processus de décision Markovien(MDP) est la donnée d'un quadruplet $\mathcal{M} \stackrel{\text{def}}{=} (\mathcal{S}, \mathcal{A}, P, r)$

- $\mathcal{S}$ est l'espace des états
- $\mathcal{A}$ l'epace des actions
- P une matrice de transition, $P(s'|s,a)$ est la probabilité d'arriver à l'état s' en étant à l'état s et en ayant choisit l'action a.
- r la fonction de récompense où $r(s,a)$ est la récompense reçue en choisissant l'action a en étant dans l'état s.

On s'intéresse à un agent qui interragit avec un MDP. Il commence dans un état s_1 et à chaque instant t, il va choisir une action a_t qui va l'amener à un nouvelle état s avec probabilité $P(s|s_t, a_t)$ et il va recevoir une récompense $r(s_t, a_t)$

Remarque. *Comme dans une chaîne de Markov, la dépendance temporelle est d'ordre 1. C'est à dire que le comportement de l'environnement ne dépends que de l'état dans lequel se trouve l'agent et pas de sa trajectoire depuis le début de l'historique.*

Comme dans les problématiques de bandits, on cherche à maximiser la récompense reçue par l'agent le long de sa trajectoire. Un agent agit selon une stratégie $\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ telle que $\pi(a|x)$ soit la probabilité de choisir l'action a en étant dans l'état x. On définit la distribution stationnaire d'état-action $\mu_\pi \in \Delta_{\mathcal{X}\times\mathcal{A}}$ où $\mu_\pi(x,a)$ est la probabilité d'être dans l'état x et de choisir l'action a. La stratégie π et la distribution stationnaire sont reliés par l'équation

$$\pi(a|x) = \frac{\mu_\pi(x,a)}{\sum_{a\in\mathcal{A}} \mu_\pi(x,a)} \tag{18}$$

On peut alors réécrire le problème d'optimisation comme

$$maximiser \quad \sum_{x,a} \mu(x,a) r(x,a) \tag{19}$$

tel que

$$\mu \in \Delta_{\mathcal{X}\times\mathcal{A}}$$

et

$$\forall x' \in \mathcal{X}, \sum_{a'} \mu(x',a') = \sum_{x,a} P(x'|x,a)\mu(x,a)$$

C'est un programme linéaire. La quantité qu'on maximise est égale(sous certaines conditions) au gain de la stratégie π que l'on définit par

$$\rho_s^\pi \stackrel{\text{def}}{=} \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}^\pi[r(s_t, a_t)|s_1 = s] \tag{20}$$

On note ρ^* la valeur de cette quantité pour la stratégie optimale. Le regret est alors défini par

$$\hat{R}_n = n\rho^* - \sum_{t=1}^{n} r(s_t, a_t) \tag{21}$$

on définit pour un MDP son diamètre :

$$D(M) \stackrel{\text{def}}{=} \max_{s\neq s'} \min_{\pi} \mathbb{E}^\pi[\min\{t \geq 1 : s_t = s\}|s_1 = s'] - 1 \tag{22}$$

On a le théorème suivant :

Theorème 6. *Soit $\mathcal{S}, \mathcal{A}$ un ensemble d'états et d'actions, n un entier naturel et $\delta \in (0, 1)$. Il existe une stratégie π qui lorsqu'elle interagit avec un MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbb{P}, r)$ où les récompenses sont à valeur dans [0,1] satisfait avec probabilité au moins $1 - \delta$*

$$\hat{R}_n < CD(M)S\sqrt{An\log(nSA/\delta)} \tag{23}$$

où C est une constante universelle

Remarque. *Il existe une borne inférieure en $\sqrt{D(M)SAn\log(nSA/\delta)}$, le facteur les séparant est $\sqrt{D(M)S}$ qui est assez important.*

Références

[1] Peter Auer, Thomas Jaksch, and Ronald Ortner. "Near-optimal Regret Bounds for Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Ed. by D. Koller et al. Vol. 21. Curran Associates, Inc., 2009. URL: https://proceedings.neurips.cc/paper/2008/file/e4a6222cdb5b34375400904f03d8e6a5-Paper.pdf.

[2] JA Bather and Herman Chernoff. "Sequential decisions in the control of a spaceship". In: *Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 3. 1967, pp. 181–207.

[3] Richard Bellman. "The theory of dynamic programming". In: *Bulletin of the American Mathematical Society* 60.6 (1954), pp. 503–515.

[4] Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

[5] Sébastien Bubeck and Nicolo Cesa-Bianchi. "Regret analysis of stochastic and nonstochastic multi-armed bandit problems". In: *arXiv preprint arXiv:1204.5721* (2012).

[6] Varsha Dani, Thomas P Hayes, and Sham M Kakade. "Stochastic linear optimization under bandit feedback". In: (2008).

[7] Dylan Foster and Alexander Rakhlin. "Beyond ucb: Optimal and efficient contextual bandits with regression oracles". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 3199–3210.

[8] Thomas Jaksch, Ronald Ortner, and Peter Auer. "Near-optimal Regret Bounds for Reinforcement Learning." In: *Journal of Machine Learning Research* 11.4 (2010).

[9] Tze Leung Lai. "Adaptive treatment allocation and the multi-armed bandit problem". In: *The Annals of Statistics* (1987), pp. 1091–1114.

[10] Tze Leung Lai and Herbert Robbins. "Asymptotically efficient adaptive allocation rules". In: *Advances in applied mathematics* 6.1 (1985), pp. 4–22.

[11] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[12] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[13] Herbert Robbins. "Some aspects of the sequential design of experiments". In: *Bulletin of the American Mathematical Society* 58.5 (1952), pp. 527–535.

[14] David Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587 (2016), pp. 484–489.

[15] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[16] Csaba Szepesvári. "Algorithms for reinforcement learning". In: *Synthesis lectures on artificial intelligence and machine learning* 4.1 (2010), pp. 1–103.

[17] William R Thompson. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". In: *Biometrika* 25.3/4 (1933), pp. 285–294.