Apprentissage par renforcement et bandits multibras

Liens utiles pour ce TP :

⋆⋆⋆ http://chercheurs.lille.inria.fr/~lazaric/Webpage/MVA-RL_Course14.html

- Introduction -

L'apprentissage par renforcement désigne un ensemble de méthodes qui s'appliquent dans le cadre où les données ne sont pas stockées à l'avance mais peuvent être vues comme un flux. Dans cette partie, nous présentons ce cadre et nous posons le problème que nous résoudrons par trois algorithmes différents au cours de ce TP.

Un cadre d'apprentissage différent

Aux deux cadres d'apprentissage déjà vus, le supervisé et le non-supervisé, on ajoute un nouveau type de modèle – dit par renforcement – pour lequel à chaque instant on doit prendre des décisions et l'on observe des réponses disponibles au fur et à mesure du temps.

Ce type d'apprentissage trouve de nombreuses applications en neurosciences computationnelles, en robotique, en économie, etc. En effet, dans de tels cas, l'enjeu est d'agir optimalement face à un historique de données afin d'engendrer de nouveaux retours d'expériences que l'on souhaite les plus positifs possible.

Exemple 1. *Un exemple historique est le cas de la comparaison de deux médicaments. Imaginons qu'un laboratoire dispose de deux traitements A et B pour une maladie. Son but est de réaliser des tests afin de déterminer lequel est le plus efficace. 1000 personnes sont recrutées et l'on doit décider de la procédure de test pour évaluer A et B : lorsque l'on administre l'un d'eux, la réponse du patient peut-être "guérir" (1) ou "mourir" (0). Une solution naïve (et dangereuse) consiste à traiter 500 patients avec A et 500 avec B et calculer la moyenne des résultats sur les deux populations. Hélas, si l'un des deux traitements est mortel, on aura sacrifié 500 personnes dans cette expérience. Peut-être aurait-il été plus intelligent d'agir différemment ? En apprentissage par renforcement on va plutôt administrer les traitements séquentiellement et adapter le choix au fur et à mesure des retours d'expérience afin de minimiser le nombre total de morts.*

Cet exemple est le point de départ historique des recherches sur le problème spécifique du "bandit multibras". Son nom vient de l'argot américain qui désigne une machine à sous par le mot "one-armed bandit" (en français : bandit manchot). En effet, le même problème peut être schématisé par un agent face à deux machine à sous : l'une gagne avec une probabilité p_1, l'autre avec une probabilité p_2. L'enjeu est de trouver une stratégie qui permette de gagner le plus possible dans une fenêtre de temps T fixée, et donc de choisir le plus vite possible le bras assurant le gain maximal.

Formalisme de l'apprentissage par renforcement

On se donne K actions possibles $A_1, \dots, A_K$. Pour $i = 1, \dots, K$, l'action A_i correspond à une tirer une loi de Bernoulli de paramètre μ_i, notées $B(\mu_i)$: on gagne (gain=1) avec probabilité μ_i et l'on perd (gain=0) avec probabilité $(1 - \mu_i)$. On suppose les actions triées par espérance décroissante : $\mu_1 > \mu_2 > \cdots > \mu_K$. Choisir l'action – ou le bras – A_k à l'instant t déclenche l'apparition d'une récompense que l'on nommera $X_t(k) \sim B(\mu_k)$. À chaque instant t, on tire le bras d'indice $I_t \in \{1, \dots, K\}$. Le nombre de tirages du bras k effectués jusqu'à l'instant t est noté $N_k(t)$.

Si l'horizon de temps T est fixé, le but est donc de maximiser le gain total qui est la somme des récompenses :

$$G_T = \sum_{t=1}^{T} X_t(I_t),$$

ou plutôt, de manière équivalente, de minimiser une quantité appelée *Regret* :

$$R_T = \sum_{t=1}^{T} \left(X_t(1) - X_t(I_t)\right),$$

qui correspond à la perte accumulée au fil des tirages de bras par rapport à la stratégie optimale qui aurait toujours tiré le meilleur bras A_1. On s'intéresse en fait à l'espérance de cette quantité par rapport à la randomisation des tirages :

$$\mathbb{E}R_T = \sum_{t=1}^{T} \left(\mu_1 - \mu_{I_t}\right). \tag{1}$$

Si l' on appelle $\mathcal{H}_t$ l'historique des actions et des récompenses jusqu'à l'instant t, le problème consiste à trouver une politique de décision $\pi : \mathcal{H}_t \mapsto \{1, \dots, K\}$ qui détermine l'action à tirer à l'instant suivant et qui minimise globalement le regret moyen

$$\min_{\pi} \mathbb{E}R_T(\pi) = \sum_{t=1}^{T} \left(\mu_1 - \mu_{\pi(t)}\right).$$

On note $\pi(t)$ la décision prise à l'instant t (mais qui peut dépendre de tout ce qui s'est passé avant cet instant).

Le compromis exploration-exploitation

Une première approche consiste à tirer quelques fois chacun des bras puis à toujours tirer celui qui a donné le meilleur gain jusque là. On dit alors qu'on décide d'**exploiter** le meilleur bras estimé. Mais dans ce cas, rien ne garantit que l'on ne s'est pas trompé dans l'estimation initiale et que l'on n'est pas en train d'accumuler les erreurs. Il faut donc introduire une certaine tendance à l'**exploration** : on veut se prémunir d'une éventuelle erreur d'estimation en allant tirer les bras sur lesquels on a un doute sans pour autant sacrifier le gain global. Il faut donc proposer des solutions qui satisfassent un compromis exploration-exploitation.

1. Quelle stratégie proposeriez-vous pour résoudre ce problème ? La question est ouverte, proposer au moins une stratégie. Il s'agit de comparer vos premières idées à celles que vous aurez découvertes par la suite.

Expérience numérique

On va réaliser l'expérience suivante : on considère $K = 4$ bras de moyennes respectives 0.1, 0.05, 0.02 et 0.01 que l'on tire en suivant des stratégies successives expliquées dans les prochaines parties. On tirera en tout 2000 fois les bras (*i.e.,* l'horizon est $T = 2000$) et chaque expérience est répétée `n_repetitions=100` fois afin que l'on puisse observer les résultats moyens sur ces répétitions. On pourra comparer différentes stratégies sur cet exemple en utilisant le script `bandits.py` et la fonction `Evaluation`.

- L'approche à l'aveugle -

2. En l'absence de réelle stratégie, on suppose que l'on tire les 4 bras uniformément, cela correspond à prendre $\pi(t)$ uniforme sur $\{1, 2, 3, 4\}$ pour tout t. Calculer le regret moyen donné en (1) d'une telle politique mathématiquement. Compléter la ligne `plot(time, time * ?TODO?, '--',label = 'no strategy')` du fichier `bandits.py` pour afficher le gain moyen obtenu par cette stratégie aveugle.

- L'approche gloutonne : ϵ-greedy -

La première idée raisonnable que l'on puisse avoir consiste à explicitement séparer exploitation et exploration. Après avoir tiré une fois chaque bras, à chaque itération on tire :

— Avec probabilité $1-\epsilon$, le bras ayant la meilleure moyenne empirique, i.e

$$I_{t+1} =: \underset{k\in\{1,\dots,K\}}{\arg\max}\ \hat{\mu}_k(t) = \underset{k\in\{1,\dots,K\}}{\arg\max}\ \frac{1}{N_k(t)}\sum_{s=1}^{T} X_s(I_t)\mathbb{1}_{\{I_t=k\}},$$

— Avec probabilité ϵ, un bras au hasard parmi les K proposés.

Remarque 1. *Notez que* $\sum_{s=1}^{T}\mathbb{1}_{\{I_t=k\}} = N_k(t)$.

3. Lancer le script `bandits.py`, avec l'option `scenario = 0` qui correspond au cas ϵ-greedy.

```
scenario = 0
n_repetitions = 100
horizon = 2000
env = MAB([Bernoulli(p) for p in [0.1, 0.05, 0.02, 0.01]])
```

4. Tester l'algorithme à l'aide du script Python fourni et faire une figure montrant les différents regrets cumulés obtenus pour plusieurs ϵ. Quelle est l'influence de ce paramètre sur le compromis exploration-exploitation ? Lisez l'implémentation de cette stratégie dans le fichier `policy/egreedy.py`.
5. Confirmer votre calcul théorique de la question 2) en prenant $\epsilon = 0.9999$ et en affichant la performance de cette méthode sur l'exemple introduit.

Algorithme 1 : Algorithme ϵ-greedy

Data : l'ensemble des observations correspondant aux actions prises jusqu'à l'instant t courant.
Result : Le bras à tirer à l'instant suivant $I_{t+1} \in \{1,\dots,K\}$
for $t = 1$ *to* T **do**
  for $k = 1$ *to* K **do**
    Calculer $\hat{\mu}_k(t)$
  Calculer $k^\star = \underset{k\in\{1,\dots,K\}}{\arg\max}\ \hat{\mu}_k(t)$
  Tirer avec probabilité $1-\epsilon$ le bras $k^\star$ et avec probabilité ϵ tirer au hasard un bras parmi les K.

- Implémenter l'optimisme : Upper Confidence Bounds (UCB) -

Une deuxième solution – moins naïve – consiste à se demander à chaque instant t quelle est l'ampleur de l'incertitude que l'on a sur notre estimation de la moyenne $\hat{\mu}_t = (\hat{\mu}_1(t),\dots,\hat{\mu}_K(t))$ pour chaque bras. Grâce aux inégalités de concentration, par exemple celle de Hoeffding, on peut estimer des intervalles de confiance autour de chaque estimation $\hat{\mu}_k(t)$. Être optimiste consiste à imaginer que tant que notre estimation est suffisamment incertaine, il est possible que la vraie moyenne se trouve au sommet de l'intervalle de confiance donc on doit tirer le bras dont la borne supérieure de confiance est la plus élevée (*cf.* Figure 1).

Construction mathématique des bornes de confiance

Commençons par un rappel :

Théorème 1 (Inégalité d'Hoeffding)**.** *Soient* $Z_1,\dots,Z_n$ *des variables aléatoires bornées identiquement distribuées, alors*

$$\mathbb{P}\left((Z_1+\cdots+Z_n)/n - \mathbb{E}[Z_1] \geq \sqrt{\frac{\ln(1/\delta)}{2n}}\right) \leq \delta\ .$$

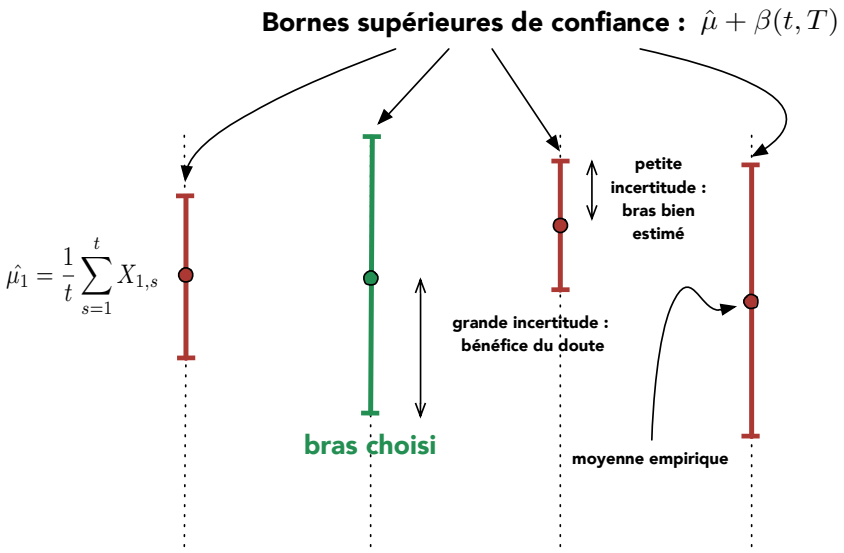

Figure 1 – Illustration du fonctionnement d'UCB.

Dans notre cas, la moyenne empirique est estimée par

$$\sum_{s=1}^{T} X_s(I_t)\mathbb{1}_{\{I_t=k\}}.$$

6. Choisir un niveau de confiance qui évolue avec t : $\delta = \frac{1}{t^\alpha}$ avec $\alpha > 1$ dans l'Algorithme 2. Corriger le code qui contrôle la classe de la politique UCB dans le fichier `policy/ucb.py` en utilisant les résultats précédents. Réaliser l'expérience. Que dire du regret moyen de cet algorithme ?

Algorithme 2 : Algorithme UCB

Data : l'ensemble des observations correspondant aux actions prises jusqu'à l'instant t courant.
Result : Le bras à tirer à l'instant suivant $I_{t+1} \in \{1, \dots, K\}$
for $t = 1$ *to* T **do**
  for $k = 1$ *to* K **do**
    Calculer $\hat{\mu}_k(t)$ et la borne supérieure de confiance pour un δ raisonnable :

$$\mathrm{UCB}_k(t) = \hat{\mu}_k(t) + \sqrt{\frac{\ln(1/\delta)}{2N_k(t)}}.$$

  Tirer le bras qui maximise cette borne supérieure de confiance :

$$I_{t+1} \in \underset{k\in\{1,\dots,K\}}{\arg\max}\ \mathrm{UCB}_k(t).$$

Preuve du comportement asymptotique logarithmique du regret

Il est possible de prouver qu'en moyenne lorsque l'horizon T tend vers l'infini, le regret de l'algorithme UCB a un comportement logarithmique : $\mathbb{E}[R(T)] = O(\ln(T))$. Cette preuve est donnée en appendice.

- Exploiter l'aléa à travers une approche bayésienne : le Thompson Sampling -

Enfin, une dernière approche du problème de bandits multibras consiste à exploiter le formalisme bayésien afin de générer des tirages de bras fondés non plus sur une estimation fréquentiste des intervalles de confiance mais sur le calcul de la loi a posteriori du paramètre μ_i qui contrôle la moyenne de chaque bras. Concrètement, comme les bras sont supposés avoir une distribution de Bernoulli, on place un a priori Bêta(α, β) sur tous les μ_i et à chaque itération on obtient une nouvelle observation qui nous permet de mettre à jour l'a posteriori de chaque bras grâce à la formule de Bayes.

Rappel : Une loi Bêta de paramètres (α, β), notée Beta(α, β), est définie par sa densité

$$p(x|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\, x^{\alpha-1}(1-x)^{\beta-1}\, \mathbb{1}_{[0,1]}(x),$$

où Γ désigne la fonction Gamma. Si les données sont supposées suivre une loi de Bernoulli de paramètre q, la vraisemblance de θ pour des observations $x_1, \dots, x_n$ s'écrit

$$l(q|x_1,\dots,x_N) = \prod_{n=1}^{N} \theta^{x_i}(1-\theta)^{1-x_i}.$$

Le formalisme bayésien consiste à imposer un a priori sur la valeur du paramètre q qui régit les observations : celui-ci devient une variable aléatoire que l'on cherche à estimer. Lorsque l'on choisit un a priori Bêta pour le paramètre q d'une loi de Bernoulli, le calcul de la loi a posteriori du paramètre q donne

$$\begin{aligned} p(\theta|x) &\propto \theta^x(1-\theta)^{1-x}\theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &\sim \text{Beta}(\alpha+x, \beta+(1-x)). \end{aligned} \tag{2}$$

On retrouve donc bien un a posteriori Bêta dont les paramètres ont changé en fonction de l'observation x. On dit que Bêta est l'a priori conjugué de la loi de Bernoulli.

Algorithme 3 : Algorithme Thomson Sampling

Data : Les paramètres initiaux α_0 et β_0.
Result : Le bras à tirer à l'instant $t+1$: $I_{t+1} \in \{1\dots, K\}$
Initialiser $\alpha(0) = \alpha_0$ et $\beta(0) = \beta_0$
for $t = 1$ *to* T **do**
  for $k = 1$ *to* K **do**
    tirer $\tilde{\mu}_k(t) \sim \text{Beta}(\alpha(t-1), \beta(t-1))$
  Tirer le bras qui a la plus grande espérance étant donné le paramètre $\tilde{\mu}_k$ tiré

$$I_{t+1} \in \underset{k\in\{1,\dots,K\}}{\arg\max}\ \tilde{\mu}_k(t). \tag{3}$$

  Observer X_{I_t} et mettre à jour $\alpha(t) = \alpha(t-1) + X_{I_t}$ et $\beta(t) = \beta(t-1) + (1 - X_{I_t})$ grâce à (2).

7. Corriger le code de la fonction `update` du fichier `posterior/beta.py` correspondant à la loi a posteriori Bêta utilisé par la politique Thompson Sampling et réaliser l'expérience. Comparer le regret moyen obtenu avec les deux précédents algorithmes.

Il a été prouvé par [2] que le Thompson Sampling a un regret logarithmique optimal pour des problèmes à récompenses binaires et pour n'importe quel choix de paramètres initiaux (α, β).

Pour aller plus loin, nous conseillons le livret [1] qui regroupe les différents modèles de bandits et les méthodes d'analyse du regret associées.

- Appendice : Contrôle du regret de l'algorithme UCB -

Notre but est de réussir à majorer le regret défini précédemment, et cela est équivalent à contrôler le nombre de bras sous-optimaux tirés au cours d'un jeu moyen. Nous détaillerons le procédé utilisé dans la preuve du théorème suivant. Pour rappel on choisit $\delta = 1/t^\alpha$ dans l'algorithme UCB.

Théorème 2. *On suppose que les récompenses de chaque bras sont bornées dans* $[0,1]$. *Alors, le pseudo-regret de l'algorithme est borné pour tout* $\alpha > 1$ *par :*

$$\bar{R}_T \leq \ln(T) \left(\sum_{i\,:\,\Delta_i > 0} \frac{2\alpha}{\Delta_i} \right) + 2\zeta(\alpha)$$

où $\Delta_i = \mu_1 - \mu_i$ *et* $\zeta(x) = \sum_{n=1}^{+\infty} 1/n^x$ *pour tout réel* $x > 1$. *Le paramètre* ϵ *introduit est appelé paramètre d'exploration.*

preuve. Avant toutes choses, la preuve de ce théorème repose sur plusieurs étapes dont nous allons décrire l'enchaînement. Il s'agit de borner le nombre de tirage de bras sous-optimaux donc nous commencerons par donner les conditions qui entraînent un tel événement. Nous en déduirons un nombre de tirages minimal permettant d'avoir une estimation raisonnable des moyennes des bras. Ensuite, nous bornerons le nombre de tirages supplémentaires effectués en nous intéressant à la probabilité de sous-estimation du bras optimal et à la probabilité de sur-estimation d'un bras sous-optimal.

Étape 1 : Une condition nécessaire et suffisante au tirage d'un bras sous-optimal

On ne tire un bras sous-optimal que lorsque sa borne supérieure UCB dépasse toutes les autres. Cela peut arriver si l'une des trois conditions ci-dessous est vérifiée :

(A) On sous-estime le bras optimal :

$$\hat{\mu}_1(t-1) + \mathrm{UCB}_1 < \mu_1.$$

(B) On sur-estime le bras sous-optimal k tiré :

$$\hat{\mu}_k(t-1) > \mu_k + \mathrm{UCB}_k(t-1).$$

(C) Ou, de manière générale, on a

$$\hat{\mu}_k(t-1) + \mathrm{UCB}_k(t-1) > \hat{\mu}_1(t-1) + \mathrm{UCB}_1(\mathrm{t}-1)$$

Ces trois conditions ne s'excluent pas entre elles mais si l'une d'entre elles est vraie alors on tire un bras sous-optimal. Réciproquement, si on a tiré un bras sous-optimal, alors au moins l'une de ces trois conditions est vraie. Cela peut aussi s'écrire

$$(A \cup B \cup C) \Longleftrightarrow (\text{un bras sous optimal est tiré à l'instant } t)$$

Notons que l'événement (A) a une probabilité plus petite que $1/t^\alpha$ (grâce à l'inégalité d'Hoeffding), idem pour l'événement (B).

Étape 2 : Un nombre de tirage minimal pour garantir une bonne estimation

La condition (C) semble être la plus commune : l'erreur d'estimation n'est pas grossière mais on sur-estime malgré tout un bras sous-optimal. A gauche et à droite, on peut respectivement majorer et minorer la condition avec probabilité au moins $1 - 1/t^\alpha$:

$$\mu_k + 2\mathrm{UCB}_k(t-1) \geq \hat{\mu}_k(t-1) + \mathrm{UCB}_k(t-1) > \hat{\mu}_1(t-1) + \mathrm{UCB}_1(\mathrm{t}-1) \geq \mu_1$$

Les deux extrémités de cette longue inégalité nous permettent d'avoir, avec probabilité au moins $1 - 1/t^\alpha$:

$$\mu_k + 2\mathrm{UCB}_k(t-1) > \mu_1 \Leftrightarrow \mathrm{UCB}_k(t-1) > \frac{\Delta_k}{2}.$$

En remplaçant $\text{UCB}_k(t-1)$ par son expression, on obtient

$$\sqrt{\frac{\alpha \ln(t-1)}{2N_k(t-1)}} > \frac{\Delta_k}{2}$$

qui équivaut à

$$\begin{aligned} N_k(t-1) <& \frac{2\alpha \ln(t-1)}{\Delta_k^2} \\ <& \frac{2\alpha \ln(T)}{\Delta_k^2} := N_{min}. \end{aligned}$$

Autrement dit, si le mauvais événement a lieu, alors avec probabilité $1 - 1/t^\alpha$ le nombre de tirages $N_k(t-1)$ est inférieur au seuil N_{min}. Pour éviter l'événement (C), on doit donc s'assurer que chaque bras est bien tiré au moins N_{min} fois.

Étape 3 : Décomposition du regret

On a montré que $\bar{R}_T = \sum_{k \neq 1} \mathbb{E}\left[N_k(T)\right] \times \Delta_k$. Afin de majorer le regret, majorons $\mathbb{E}\left[N_k(T)\right]$ pour chaque k dans le cas où on a déjà tiré au moins N_{min} fois chaque bras :

$$\begin{aligned} \mathbb{E}\left[N_k(T)\right] \leq& N_{min} + \mathbb{E}\left[\sum_{t=N_{min}+1}^{T} \mathbb{1}_{\left\{I_t = k \text{ et } N_k(t) \geq N_{min}\right\}}\right] \\ \leq& N_{min} + \mathbb{E}\left[\sum_{t=N_{min}+1}^{T} \mathbb{1}_{\{A \text{ ou } B\}}\right] \\ \leq& N_{min} + \sum_{t=N_{min}+1}^{T} \mathbb{P}\left(A\right) + \mathbb{P}\left(B\right). \end{aligned}$$

On connaît déjà la valeur de ces deux probabilités puisqu'elles correspondent exactement au niveau de confiance imposé sur les intervalles pour la construction de l'algorithme. D'ailleurs, ces deux probabilités sont égales donc on peut écrire

$$\begin{aligned} \mathbb{E}\left[N_k(T)\right] \leq& N_{min} + 2 \sum_{t=N_{min}+1}^{T} \frac{1}{t^\alpha} \\ \leq& \frac{2\alpha \ln(T)}{\Delta_k^2} + 2 \sum_{t=1}^{\infty} \frac{1}{t^\alpha}. \end{aligned}$$

La série des $\left(1/n^\alpha\right)_{n>0}$ converge pour tout $\alpha > 1$ et sa somme vaut $\zeta(\alpha)$.

Étape 4 : Conclusion

Finalement, on a obtenu

$$\mathbb{E}\left[N_k(T)\right] \leq \frac{2\alpha \ln(T)}{\Delta_k^2} + 2\zeta(\alpha)$$

Donc le pseudo-regret est borné par

$$\begin{aligned} \bar{R}_T =& \sum_{k \neq 1} \mathbb{E}\left[N_k(T)\right] \times \Delta_k \\ \leq& \ln(T) \left(\sum_{k \neq 1} \frac{2\alpha}{\Delta_k}\right) + 2 \left(\sum_{k \neq 1} \Delta_k\right) \zeta(\alpha). \end{aligned}$$

Références

[1] S. Bubeck. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1) :1–122, 2012. 5

[2] E. Kaufmann, N. Korda, and R. Munos. Thompson sampling : An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory*, pages 199–213. Springer, 2012. 5