TELECOM ParisTech **INFMDI341**
2015/2016 **Claire Vernade, Alexandre Gramfort, Joseph Salmon**

TP n° 6 : Apprentissage par renforcement et bandits multi-bras

Liens utiles pour ce TP :

⋆⋆⋆ Les notes de cours : https://cvernade.wp.mines-telecom.fr/teaching/

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

- Introduction -

L'objectif de cette séance est de mettre en œuvre les différents algorithmes de la section 2 du cours sur les bandits manchots. Après avoir rappelé le cadre d'apprentissage dans lequel nous nous plaçons, nous proposons successivement trois algorithmes dont certains ont été vus en cours et nous comparons leurs performances respectives.

Motivations du bandit manchot

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 manchot". Son nom vient de l'argot américain qui désigne une machine à sous par le mot "one-armed bandit". En effet, comme vu en cours, 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 ou non, et donc de choisir le plus souvent possible le bras assurant le gain maximal.

Formalisme du problème considéré

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, on perd (gain=0) avec probabilité $(1 - \mu_i)$. On suppose, sans perte de généralité, que les actions sont 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 :

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

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).

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 `nMC=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 `main.py`.

1) Avant toute chose, il est bon de savoir quelle performance minimale peut être atteinte par les stratégies que nous allons tester. Compléter la fonction `computeLowerBound` du fichier `source.py` afin que celle-ci retourne bien la valeur du coefficient de la borne inférieure de Lai et Robbins vue en cours. On s'aidera pour cela de la fonction `kl(a,b)` fournie et permettant de calculer la divergence de Kullback-Leibler de la loi $B(a)$ vers la loi $B(b)$.

- 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 à la main le regret moyen donné en (1) d'une telle politique. Compléter la ligne `coeff = #TODO` du fichier `main.py` pour afficher le gain

- 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 la première partie du script en choisissant quelques paramètres ϵ à tester.

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 `source.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.

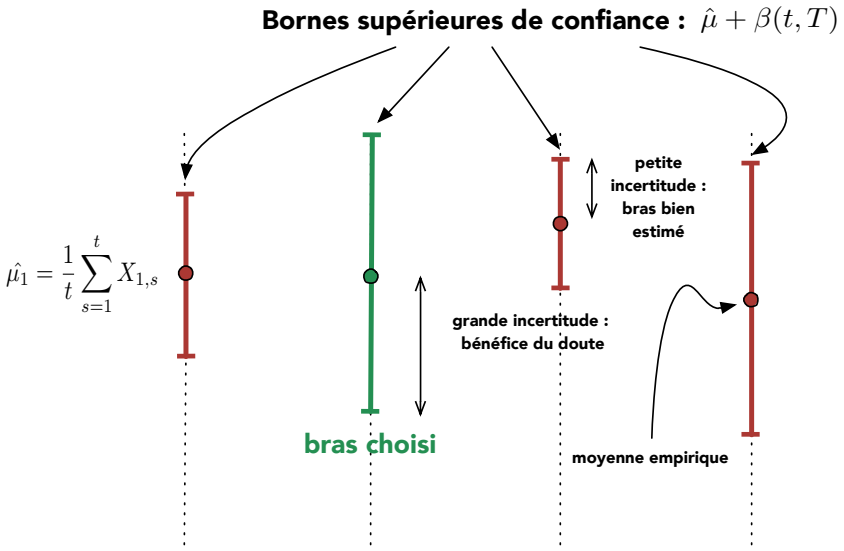

FIGURE 1 – Illustration du fonctionnement d'UCB.

- 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*

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

Dans notre cas, la moyenne empirique est estimée par

$$\hat{\mu}_k(T) = \frac{1}{N_k(T)} \sum_{s=1}^{T} X_s(I_t) \mathbb{1}_{\{I_t = k\}}.$$

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 fonction de la politique UCB dans le fichier `source.py` en utilisant les

résultats précédents. Reprendre l'expérience précédente, cette fois en calculant le regret moyen pour la méthode UCB. Comparer avec l'algorithme précédent ?

Algorithme 2 : Algorithme UCB

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 du cours pour $\alpha = 4$.

- 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

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

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.

Il a été prouvé par [1] 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).

7) Corriger le code de la fonction `Thompson` du fichier `source.py` afin de générer des échantillons sous la loi Bêta convenant à la politique Thompson Sampling et réaliser l'expérience.

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).

8) Écrire un petit script permettant de comparer cette politique avec les précédentes. Que dire du regret moyen obtenu dans chacun des cas ?

Pour aller plus loin sur les bandits, nous conseillons le livret [2] qui regroupe les différents modèles de bandits et les méthodes d'analyse du regret associées.

Références

[1] E. Kaufmann, N. Korda, and R. Munos. Thompson sampling : An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory*, pages 199–213. Springer, 2012. 4

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