Systèmes de recommandations : algorithmes de bandits et évaluation expérimentale

Jonathan Louëdec[1,2] , Max Chevalier[2], Aurélien Garivier[1], Josiane Mothe[2]

[1] *Institut de mathématiques de Toulouse, Université de Toulouse,*
prenom.nom*@math.univ-toulouse.fr*
[2] *Institut de Recherche en Informatique de Toulouse, Université de Toulouse,*
prenom.nom*@irit.fr*

Résumé. Les systèmes de recommandation à très grande échelle sont aujourd'hui omniprésents sur internet : ouvrages conseillés à l'achat dans les librairies en ligne, articles recommandés sur les sites d'information, sans parler des cadres publicitaires qui financent l'essentiel de très nombreux sites aujourd'hui... Trouver la meilleure recommandation à faire à un visiteur peut être considéré comme un "problème de bandits" : il faut en même temps apprendre ses préférences, et utiliser les interactions déjà passées pour maximiser le nombre de recommandations suivies, tout en restant capable de gérer des flux de données très importants.

Nous présentons ici quelques-uns des algorithmes les plus célèbres pour résoudre ce type de problèmes, et notamment l'algorithme UCB (Upper Confidence Bound), l'algorithme EXP3 (Exponential weights for Exploration and Exploitation) et le Thompson Sampling (du nom de l'inventeur, au début des années trente, de cette méthode d'inspiration bayésienne). Leurs mérites respectifs sont soulignés et discutés, avec la présentation des résultats théoriques les plus importants les concernant. Nous montrons en outre, dans un notebook ipython associé, comment expérimenter l'efficacité de ces méthodes pour la recommandation : ceci pose une difficulté particulière, car des jeux de données statiques rendent peu aisée l'évaluation de méthodes vouées à interagir avec des utilisateurs. Nous montrons en particulier comment mettre en place des expériences sur deux jeux de données célèbres : movielens et jester.

Mots-clés. système de recommandation, algorithmes de bandits, évaluation sur données réelles.

Abstract. Large scale recommender systems are now ubiquitous on the web: they are massively used by online bookstores, news providers and aggregators, not to mention ad servers, on which a large part of the economic viability of the net economy relies. Finding the best item to recommend to a user can be modeled as a "bandit problem": the difficulty is, in a context of massive data streams, to learn the user's preferences while, at the same time, using past interactions to maximize the number of fruitful suggestions.

We review here some famous algorithms addressing this problem: Upper-Confidence Bound strategies, the EXP3 algorithm (Exponential Weights for Exploration and Exploitation), and Thompson Sampling (named after the creator, in the early 1930s, of this Bayesian-flavored method). Their respective qualities are discussed, together with some important theoretical guarantees. In addition, we show in a companion ipython notebook how to experiment the efficiency of these methods for recommendation: there is a special difficulty because static data sets do not directly allow to evaluate the performance of methods dedicated to user interactions. We show in particular how to set up experiments using two famous data sets: movielens and jester.

Keywords. Recommender Systems, Bandit Algorithms, Evaluation on Real Data.

1 Systèmes de recommandation et modèles de bandits

Même si l'avènement d'internet et des systèmes de recommandations a fortement relancé l'intérêt qui leur est porté, l'étude des problèmes de bandits a débuté il y a longtemps, et leur intérêt applicatif est bien plus vaste. Ils modélisent des situations où un agent, plongé à chaque instant dans un certain contexte, doit choisir séquentiellement une suite d'actions qui lui assurent un certain gain aléatoire, et qui influent sur ses observations futures. Il s'agit de concevoir et d'analyser des règles de décision dynamiques, appelées *politiques*, utilisant les observations passées pour optimiser les choix futurs. Une bonne politique doit réaliser un savant équilibre entre l'*exploitation* des actions qui se sont révélées payantes par le passé et l'*exploration* de nouvelles possibilités qui pourraient s'avérer encore meilleures. Initialement motivés essentiellement par la thématique des essais cliniques, ces modèles interviennent désormais dans de nombreux autres domaines industriels, les technologies de l'information en ayant multiplié les opportunités.

L'étude mathématique des problèmes de bandits[1] remonte à l'article pionnier [1]. De nombreux travaux ont suivi, notamment dans le champ de l'apprentissage statistique, en relation avec la théorie des jeux, l'apprentissage actif, et l'agrégation d'estimateurs : on pourra par exemple se référer à l'ouvrage de référence [2]. Dans cette littérature, de nombreux problèmes tant théoriques que computationnels sont abordés, en combinant théorie des probabilités et optimisation convexe. La communauté statistique a également contribué, notamment sous la dénomination d'*inférence séquentielle* (voir [3,4] et les références citées), avec un point de vue essentiellement asymptotique.

Dans la version stochastique[2] la plus simple du problème, l'agent choisit à chaque étape

[1]Ce nom fait référence à la situation paradigmatique d'un joueur faisant face à une lignée de machines à sous et cherchant sur laquelle tenter sa chance afin de maximiser ses gains.

[2]Il existe une variante dite *adversariale*, relevant de la théorie de jeux, où les récompenses sont choisies non pas au hasard mais par un adversaire; il n'en sera pas question ici : outre [5], le lecteur intéressé pourra se réfèrer avec profit aux travaux de Gilles Stoltz pour en saisir les subtilités.

$t = 1, 2, \ldots$ un bras $A_t \in \{1, \ldots, K\}$, et il reçoit une récompense X_t telle que, conditionnellement au choix des bras $A_1, A_2, \ldots$, les récompenses soient indépendantes et identiquement distribuées, d'espérances $\mu_{A_1}, \mu_{A_2}, \ldots$ On appelle *politique* la règle de décision (potentiellement randomisée) qui, aux observations passées $(A_1, X_1, \ldots, A_{t-1}, X_{t-1})$, associe le prochain choix A_t. Le meilleur choix (inconnu de l'agent) est le bras a^* qui correspond à la récompense moyenne maximale μ_{a^*}. La performance d'une politique est mesurée par le *regret* R_n, qui est défini comme la différence moyenne entre les récompenses qu'elle permet d'accumuler jusqu'au temps $t = n$ et ce qui aurait pu être obtenu pendant la même période si le meilleur bras était connu à l'avance :

$$R_n = n\mu_{a^*} - \mathbb{E}\left[\sum_{t=1}^{n} X_t\right] .$$

Un algorithme de bandit ne peut pas être arbritrairement bon : il existe une borne inférieure au regret qu'il doit encourir dès lors qu'il offre des garanties uniformes de performance. La plus célèbre de ces bornes inférieures est celle de Lai et Robbins [6] (voir [7] pour une preuve moderne et plus générale). Dans le cas de récompenses binaires, elle stipule que si une politique assure dans tout environnement un regret sous-polynômial, alors celui-ci est toujours au moins logarithmique : quels que soient $\mu_1, \ldots, \mu_K \in]0, 1[$,

$$R_n \geq \left(\sum_{a : \mu_a < \mu_{a^*}} \frac{\mu_{a^*} - \mu_a}{\mathrm{kl}\,(\mu_a, \mu_{a^*})} \right) \log(n) \, \big(1 - \mathrm{o}(1)\big), \tag{1}$$

où kl désigne l'entropie binaire: $\mathrm{kl}(x, y) = x \log(x/y) + (1 - x) \log\big((1 - x)/(1 - y)\big)$.

2 Algorithmes de bandits

Il convient pour commencer de souligner que la première politique qui vient à l'esprit n'est pas la bonne : il est souvent catastrophique de se fier, à l'étape t, aux récompenses obtenues dans le passé pour construire des estimateurs $\hat{\mu}_1, \ldots, \hat{\mu}_K$ des paramètres $\mu_1, \ldots, \mu_K$, en faisant comme si on avait pour chaque bras un échantillon de taille déterministe, puis de choisir le bras a qui maximise $\hat{\mu}_a$ — et ce même si l'on commence par une petite phase d'exploration où chaque bras est essayé un certain nombre de fois fixé à l'avance. Une telle politique peut être trompée par des observations un peu malchanceuses lors des premiers essais du meilleur bras, et ne réalisera jamais son erreur si celui-ci n'est plus jamais tiré. Et s'il est nécessaire de se concentrer sur les bras qui ont montré leur performance dans le passé, il convient de ne jamais exclure complètement les autres.

Si de nombreux algorithmes ont été proposés pour résoudre ce *dilemme exploration-exploitation*, on se contentera de désigner ici trois grandes familles : les méthodes de type UCB (qui se fient non à la performance estimée d'un bras, mais à une borne supérieure de

confiance), les méthodes de types softmax (où l'on tire son action suivant une loi de probabilité qui favorise celles qui semblent les plus prometteuses), et le Thompson Sampling proposé dans [1] au début des années 1930 : d'une simplicité biblique, ce dernier présente toutefois de grandes qualités pratiques, tandis que des garanties théoriques d'optimalité ont récemment été montrées [8].

2.1 ϵ-greedy

La politique ϵ-greedy (également utilisée dans le cadre plus large de l'apprentissage par renforcement, voir par exemple [9]) est la plus simple que l'on puisse concevoir : pour choix de paramètres $(\epsilon_t)_{t\geq 1}$, elle choisit à l'instant t l'action qui maximise la performance moyenne sur les observations $(A_1, X_1, \ldots, A_{t-1}, X_{t-1})$ faites jusque là avec une probabilité $1-\epsilon$, et sinon avec probabilité ϵ une action uniformément au hasard.

Dans les expérimentations appliquées, il n'est pas rare de voir un choix $\epsilon_t = \epsilon$ constant. Il n'est cependant pas difficile de rendre le regret négligeable devant n, en choisissant une suite (ϵ_t) décroissant (pas trop vite) vers 0. Cependant, il n'est pas possible d'atteindre le regret asymptotiquement optimal suggéré par la borne (1) de Lai et Robbins : cette politique paye donc sa grande simplicité par une performance moindre que ses concurrentes.

2.2 UCB : Upper-Confidence Bound

Une autre idée féconde en apprentissage par renforcement s'applique de façon particulièrement convaincante pour les bandits : l'approche dite "optimiste" où l'agent fait à chaque instant comme si, parmi toutes les distributions de récompenses possibles statistiquement compatibles avec ses observations passées, il avait face à lui celle qui lui est la plus favorable.

Dans le cas présent, cela revient à se fier pour chaque bras non à un estimateur de son efficacité moyenne, mais plutôt à une *borne supérieure de confiance* (upper-confidence bound, UCB). La référence la plus citée pour l'algorithme UCB est [10] (l'idée est plus ancienne, voir les références citées), pour le cas des récompenses bornées entre 0 et 1. L'algorithme est le suivant : si jusqu'à l'instant $t-1$ on a fait $N_a(t-1)$ tirages du bras $a \in \{1, \ldots, K\}$ qui ont apporté une récompense cumulée $S_a(t-1)$, le théorème de Hoeffding suggère une borne supérieure de confiance pour μ_a de la forme[3]:

$$\mathrm{UCB}(a) = \frac{S_a(t-1)}{N_a(t-1)} + \sqrt{\frac{f(t)}{2N_a(t-1)}}$$

où $f(t)$ est un paramètre qui permet de régler le niveau de confiance. L'action A_t alors choisie est celle qui maximise $\mathrm{UCB}(a)$. Il est prouvé dans [10] une borne de regret logarithmique (non-asymptotique) pour un choix adéquat de $f(t)$.

[3]Il faut toutefois prendre garde au fait que les données disponibles sur le bras a ne constituent pas un échantillon de taille déterministe !

Toutefois, l'algorithme ci-dessus a un comportement sous-optimal qui peut s'avérer assez décevant dans le cas (fréquent en recommandation) où les récompenses moyennes sont toutes très faibles (cela se conçoit bien : la borne de Hoeffding est alors très pessimiste). Heureusement, l'analyse peut être significativement renforcée, et [11] présente une variante pour laquelle une borne non-asymptotique de regret est montrée d'où peut être déduite l'optimalité au sens de la borne (1) de Lai et Robbins. Le calcul de la borne supérieure de confiance est un peu plus complexe, mais le gain de performance n'est pas seulement théorique. Les méthodes de type UCB présentent donc de grandes qualités ; elles posent toutefois des difficultés dans les modèles plus complexes (par exemple, dans le cas où les récompenses dépendent de façon non triviale de covariables) où il devient difficile de construire des intervalles de confiance précis avec les bonnes garanties non asymptotiques.

2.3 EXP3 : Exponential Weights for Exploration and Exploitation

Une autre façon de ne pas tomber dans le piège de choisir toujours l'action qui maximise l'efficacité moyenne passée consiste à remplacer le "maximum dur" par un "maximum doux" (softmax), c'est-à-dire de maintenir à jour une loi de probabilité sur l'ensemble des bras qui ne donne pas toutes les chances au meilleur, mais qui en laisse aussi un peu aux autres, et de tirer le choix A_t sous cette loi. La politique ϵ-greedy mentionnée ci-dessus rentre dans ce cadre, mais la politique la plus représentative de cette famille est appellée EXP3 pour "Exponential Weights for Exploration and Exploitation" (voir [2,5] et les références citées). Cette politique s'inspire des idées utilisées pour la combinaison d'experts, en ajoutant une couche d'estimation. Elle peut être étudiée de nombreux points de vue : théorie des jeux, modèle pseudo-bayésien, optimisation bruitée et descente de gradient miroir... Plusieurs variantes ont été envisagées, la plus simple consiste à
- donner un poids initial $w_a^t = 1$ à chaque action $a \in \{1, \dots, K\}$;
- tirer, au temps t, l'action $A_t = a$ avec une probabilité $p_t(a)$ proportionnelle à son poids w_a^t;
- une fois obervée la récompense X_t, mettre à jour le poids de l'action A_t en le multipliant par $\exp(-\eta_t X_t / p_t(a))$, où η_t est un paramètre de l'algorithme.

Dans le cas des bandits bornés entre 0 et 1, un choix adéquat pour le paramètre est $\eta_t = \sqrt{\log(K)/(Kt)}$. Pour ce choix, on arrive à contrôler *uniformément* le regret : $R_n \leq 2\sqrt{nK\log(K)}$ (voir [2]). Cette borne de regret peut sembler décevante en regard des taux logarithmiques obtenus pour UCB, mais il convient de noter qu'il s'agit de bornes uniformes qui ne font pas intervenir les récompenses moyennes du problème considéré (et on peut prouver que, pour de telles bornes uniformes, on ne peut pas faire mieux). En outre, cette borne est en fait vraie au sens des séquences individuelles, ce qui est beaucoup plus fort (en particulier, elle ne nécessite pas de supposer que les récompenses sont i.i.d. conditionnellement aux actions). On pourra retenir que les algorithmes softmax sont

moins performants mais plus robustes. Noter que des tentatives ont été faites de combiner le “meilleur des deux mondes” (cf [11]).

2.4 Thompson Sampling

Pour finir, mentionnons ici la plus ancienne (et pourtant pour certains la plus performante, voir [8] et les références citées) de toutes les politiques : le Thompson Sampling. Il s'agit également d'une politique randomisée, mais qui utilise l'aléa d'une autre façon d'inspiration bayésienne : on suppose que l'on dispose pour chaque bras $a \in \{1, \dots, K\}$ d'une loi a priori chargeant l'ensemble des lois qui peuvent lui être associées. Sur chaque bras, cette loi est mise à jour en une loi a posteriori à chaque fois qu'il est tiré et que, par conséquent, une observation est faite pour lui. Pour déterminer l'action choisie à l'instant t, un tirage aléatoire (indépendant de tout le reste) est effectué sous la loi a posteriori de chaque bras : celui pour lequel le tirage est le plus grand est tiré.

Cette politique a l'avantage de pouvoir être mise en œuvre dans une grande variété de cas. Pas besoin de travaux fins pour construire des intervalles de confiance adaptés à la géométrie des familles de lois : la loi a posteriori s'adapte toute seule. Sa performance est en outre excellente : elle a longtemps été utilisée en pratique, en particulier pour le placement de publicités, sans que des garanties théoriques satisfaites n'aient été obtenues. Ce défaut a été palié assez récemment : il a même été montré que le Thompson Sampling atteint la borne de Lai et Robbins (voir notamment [8]).

3 Évaluation des algorithmes de bandits

Le notebook ipython associé à cette présentation contient deux types d'expériences. Les premières sont des simuations de problèmes de bandits qui permettent de se faire une idée du comportement des algorithmes présentés ci-dessus dans quelques contextes simples.

Au-delà des expériences simulées, il est intéressant de pouvoir estimer la performance des algorithmes de bandits pour la recommandation de contenu. Malheureusement, il est le plus souvent impossible de mettre en place de vraies procédures de test où des recommandations sont envoyées à des utilisateurs dont on peut enregistrer les réactions. Pour résoudre cette difficulté, une solution classique consiste à utiliser un jeu statique de préférences exprimées pour mimer synthétiquement des interactions.

Le protocole que nous présentons ici est celui de l'article [12], qui a ensuite été repris de nombreuses fois, et qui est relativement aisé à mettre en place. Il s'appuie sur des parties des jeux de données MovieLens et Jester. MovieLens-100 contient les notes (de 1 à 5) données par 943 utilisateurs à 100 films. Jester contient les appréciations (de -10 à 10) de 25000 utilisateurs sur 100 blagues.

Pour simuler l'aspect temps réel du système de recommandation, un utilisateur est choisi aléatoirement à chaque instant et m objets lui sont recommandés (le modèle

de bandits classique correspond à $m = 1$). On suppose que la recommandation est suivie si la note associée est strictement supérieure à un seuil de pertinence choisi à l'avance : la récompense obtenue est alors égale à 1. L'objectif est maximiser la somme des récompenses, autrement dit de minimiser l'abandon : c'est l'objectif classique des algorithmes de bandits[4].

En recommandation il est fréquent que le nombre m d'items proposés soit strictement supérieur à 1 : c'est alors une liste (ordonnée ou non) qu'il faut fournir. Ce problème, parfois appelé "bandits multi-actions" entre dans la catégorie des *bandits combinatoires*, une extension du modèle très étudiée pour laquelle plusieurs approches sont possibles (voir [2]). Nous montrons dans le notebook des expériences sur une approche spécifique à la recommandation de listes ordonnées de m actions : l'algorithme RBA (Ranked Bandits Algorithm) de [13]. Pour chaque position $k \in \{1, \ldots, m\}$, on utilise une instance d'algorithme de bandits dont le choix d'action, à l'instant t, fournit la k-ième suggestion. Toutefois, seule la première action cliquée de la liste reçoit une récompense : même si d'autres actions plus loin dans la liste sont cliquées, la récompense que reçoit le bandit correspondant est nulle. Ainsi, le bandit en position 1 se concentre rapidement sur l'item le plus cliqué, puis celui en position 2 sur l'action qui maximise les clics sur l'une ou l'autre de ces deux positions, etc...

Remerciements:
Ce travail a bénéficié du soutien de l'Agence Nationale de la Recherche portant les références ANR-13-BS01-0005 (projet SPADRO) et ANR-13-CORD-0020 (projet ALICIA). Jonathan Louëdec est financé par le projet ANR-11-LABX-0040-CIMI du programme ANR-11-IDEX-0002-02.

Bibliographie

[1] Thompson, W. (1933), On the likelihood that one unknown probability exceeds another in view of the evidence of two sample *Bulletin of the American mathematics society*, 25:285–294.
[2] Bubeck, S. and Cesa-Bianchi, N. (2012), Regret analysis of stochastic and nonstochastic multi-armed bandit problems, *Foundations and trends in machine learning*, 5(1):1–122.
[3] Hu, F. and Rosenberger, W. F. (2006), *The theory of response-adaptive randomization in clinical trials*, Wiley Series in Probability and Statistics. Wiley-Interscience [John Wiley & Sons], Hoboken, NJ.
[4] Jennison, C. and Turnbull, B. W. (2000), *Group sequential methods with applications to clinical trials*, Chapman & Hall/CRC, Boca Raton, FL.

[4]Un autre objectif possible est de maximiser la probabilité d'identifier le meilleur bras : les algorithmes et les bornes sont alors très différents (voir [7] et les références citées); mais ici c'est bien la minimisation du regret qui est pertinente.

[5] Cesa-Bianchi, N. and Lugosi, G. (2006), *Prediction, Learning, and Games*, Cambridge University Press.
[6] Lai, T. and Robbins, H. (1985), Asymptotically efficient adaptive allocation rules, *Advances in Applied Mathematics*, 6(1):4–22.
[7] Kaufmann, E. and Cappé, O. and Garivier, A. (2015), Complexity of Best-Arm Identification in Multi-Armed Bandits, *Journal of Machine Learning Research*.
[8] Kaufmann, E. and Korda, N. and Munos, R. (2012), Thompson Sampling : an Asymptotically Optimal Finite-Time Analysis, *Algorithmic Learning Theory (ALT)*.
[9] Sutton, R.S. and Barto, A.G. (1998), *Reinforcement learning: An introduction*, The MIT press.
[10] Auer, P. and Cesa-Bianchi, N. and Fischer, P. (2002), Finite-time analysis of the multiarmed bandit problem, *Machine Learning*, 47(2):235–256.
[11] Bubeck, S. and Slivkins, A. (2012), The best of both worlds: stochastic and adversarial bandits, *Conference On Learning Theory (COLT)*.
[12] Kohli, P. and Salek, M. and Stoddard, G. (2013), A Fast Bandit Algorithm for Recommendation to Users With Heterogeneous Tastes, *27th AAAI Conference on Artificial Intelligence*, 1135–1141.
[13] Radlinski, F. and Kleinberg, R. and Joachims, T. (2008) Learning Diverse Rankings with Multi-Armed Bandits, *25th International Conference on Machine Learning*, 784–791.