Algorithmes de bandits pour les systèmes de recommandation

Jonathan Louëdec

Institut de Recherche en Informatique de Toulouse (IRIT)
Institut de Mathématiques de Toulouse (IMT)
UMR5505, UMR5219, CNRS
118 Route de Narbonne
31062 Toulouse, France

Jonathan.Louedec@irit.fr

MOTS-CLÉS : *Recherche d'Information, Systèmes de Recommandation, Apprentissage actif, Contexte Utilisateur, Problèmes de Bandits*

KEYWORDS: *Information Retrieval, Recommender Systems, Active Learning, User Context, Bandits Problem*

ENCADREMENT. *IRIT: Josiane Mothe (PR) et Max Chevalier (MCF), IMT: Aurélien Garivier (PR)*

1. Contexte

Les systèmes de recommandation (SR) ont pour objectif de proposer à l'utilisateur des items (documents, objets, films, musiques, informations, ...) susceptibles de l'intéresser. Deux approches principales sont mises en œuvre : le filtrage basé sur le contenu (FBC) recommande à un utilisateur des items similaires à ceux qu'il a déjà aimés par le passé. Le filtrage collaboratif (FC) recommande les items appréciés par les utilisateurs qui ont auparavant fait des choix similaires à ceux de l'utilisateur. D'autres approches existent : le filtrage démographique se base sur ce que l'on sait de l'utilisateur (âge, données démographiques, sexe, ...) ; le filtrage communautaire utilise les décisions faites par les contacts de cet utilisateur (cette méthode est notamment utilisée dans les SR sociaux).

2. État de l'art

De nouveaux items et de nouveaux utilisateurs apparaissent au cours du temps, ils doivent être pris en compte par les SR. Le FC et le FBC souffrent de quelques limites bien répertoriées dans les travaux existants. Quels items proposer à un nouvel utilisateur ? Ce problème, connu sous le nom de démarrage à froid utilisateur, est commun aux FC et FBC. Par ailleurs il existe des problèmes spécifiques à chaque type d'approches. Le départ à froid côté item (à qui proposer un nouvel item ?) est une limitation du FC uniquement. Celle-ci peut être traitée en comparant le nouvel item avec les items antérieurs : c'est l'idée du FBC. Au contraire si le contenu d'un item est peu décrit, le FBC ne pourra pas établir de similarité, tandis que le FC pourra le proposer en fonction de l'information obtenue par les actions utilisateurs passées (Montaner, 2003). Ainsi, la combinaison entre FC et FBC peut permettre de corriger les problèmes spécifiques de chaque approche : actuellement, les SR les plus efficaces sont basés sur une approche hybride (Ricci *et al.*, 2011).

Koren (2010) souligne qu'un SR doit par ailleurs être capable de prendre en compte l'évolution des besoins de l'utilisateur : c'est la définition d'un SR dynamique. Deux approches pour ce problème sont l'utilisation d'une fenêtre temporelle, permettant de prendre en compte uniquement les événements les plus récents et la mise en place de méthodes de détection de rupture.

3. Problématique

Dans le contexte de la recommandation nous proposons une approche de traitement séquentielle. Lorsque l'ensemble des interactions des utilisateurs avec le SR jusqu'à l'instant t est connu, se pose la question de quels items proposer à l'instant $t + 1$. Il faut pour cela mettre en œuvre une stratégie pour acquérir de l'information sur les utilisateurs et l'ensemble des items (exploration) tout en assurant que le SR donnera de bons résultats (exploitation). Cette problématique est connue sous le nom de dilemme exploration/exploitation. La stratégie doit pouvoir s'adapter aux flux de données (nouvel utilisateur, nouveau document, évolution des goûts de l'utilisateur). Elle devra également prendre en compte le passage à l'échelle, ce qui pose deux problèmes : comment stocker l'information ? Comment optimiser le temps d'apprentissage ? Nous avons choisi d'étudier l'applicabilité de l'approche statistique « problème de bandits » à l'ensemble des problèmes des SR évoqués plus haut. Dans nos premiers travaux nous nous sommes focalisés sur l'aspect volume de données afin de pouvoir ensuite appliquer les algorithmes de bandits sur des collections réelles.

4. Actions réalisées

Fin 2013, le moteur de recherche russe Yandex a proposé un challenge dont l'objectif était de réordonner les dix premiers résultats de 797 867 requêtes soumises par des utilisateurs différents (`http://www.kaggle.com/c/`

`yandex-personalized-web-search-challenge`). Nous disposions de plus de 34 millions de requêtes passées et les actions utilisateurs liées : clics, temps passé sur la page cliquée. Chaque terme et chaque site web était remplacé par un identifiant numérique, il n'était donc pas possible de prendre en compte la sémantique des différents éléments. Les actions passées des utilisateurs constituaient les seuls éléments dont nous disposions pour traiter le problème. Dans un premier temps nous avons dû mettre en place une structure permettant de stocker et de traiter rapidement ces données. Nous avons fait le choix d'utiliser MongoDB, un système de gestion de base de données NoSQL orientée documents. Via un script Python, l'insertion en base des données a duré deux heures et, une fois les index créés, la base occupait 117 Go. Nous avons proposé plusieurs stratégies pour répondre au challenge de ré-ordonnancement. Celui présenté en Algorithme 1 à permis d'obtenir les meilleures performances, nous avons obtenu la 33-ième place sur 196 participants.

Algorithme 1 : Méthode mise en œuvre pour le ré-ordonnancement

```
pour chaque requête R pour laquelle les résultats doivent être réordonnés faire
    pour chaque document D de la liste réponse de R faire
        Récupérer le nombre de clics de l'utilisateur sur D avec au moins un
        terme de R dans la requête dans les sessions passées (apprentissage)
    fin
    Réordonner les documents en fonction de leur score, conserver l'ordre de
    départ si ex-æquo
fin
```

5. Actions futures

Pour aborder l'aspect flux de données, nous voulons mettre en place une méthode prenant en compte le dilemme exploration/exploitation. En effet nous souhaitons pouvoir détecter un changement de goûts ou prendre en compte un nouvel item sans diminuer les performances du SR. Les algorithmes de bandits sont connus pour proposer des solutions à ce dilemme, voir (Bubeck *et al.*, 2012). Cette approche est séquentielle : à partir des t premiers résultats, une décision est prise à l'instant $t + 1$. En fonction de l'item choisi et du résultat obtenu, l'estimateur associé est mis à jour et le résultat stocké. Peu de calculs sont faits à chaque étape, ce qui permet d'obtenir des temps de traitement très courts. Avec un système de stockage et une structure de données adaptés, il est possible d'obtenir des temps de réponses de l'ordre de quelques millisecondes.

Certains d'entre eux, UCB par exemple, utilise des bornes supérieures de confiance à la place des estimateurs. Une conséquence de cette approche optimiste est qu'un nouvel item sera proposé un certain nombre de fois dès son apparition pour estimer sa valeur. Concernant l'évolution des goûts d'un utilisateur et le vieillissement d'un item, il faudrait adapter la méthode de calcul de la borne supérieure pour détecter cette évolution. Pour cela nous pensons utiliser une fenêtre temporelle et les techniques de détection de rupture.

Dans les versions les plus basiques, les algorithmes de bandits traitent chaque item de manière indépendante : un clic utilisateur sur un item modifiera uniquement l'estimateur de cet item. S'il y a de nombreux items à traiter, nous souhaitons inférer de l'information sur les items similaires à celui choisi afin de limiter le temps d'apprentissage. Lorsqu'un nouvel utilisateur apparaît, il s'agit d'acquérir un maximum d'informations en un minimum de temps pour estimer ses goûts. Actuellement les méthodes d'apprentissage actif (Rubens *et al.*, 2011) traitent ce problème en choisissant à chaque instant l'item qui apportera le plus d'informations. L'interaction des méthodes d'apprentissage actif avec celles des problèmes de bandits devrait permettre d'optimiser la partie exploration au profit de l'exploitation.

En recherche d'information, plusieurs chercheurs ont adapté les algorithmes de bandits. Li *et al.* (2010) proposent un algorithme linéaire de bandits, LinUCB, prenant en compte le contexte utilisateur. Dans le cadre du ré-ordonnancement d'items, Radlinski *et al.* (2008) proposent le « Ranked Bandits Algorithm ». Les auteurs mettent en place des algorithmes de bandits pour chaque position possible dans l'ordonnancement des résultats. Ils prouvent que sur le long terme l'algorithme mis en place est meilleur qu'un algorithme basé uniquement sur la popularité des documents. La méthode d'évaluation reste tout de même très théorique, et une évaluation sur un exemple réel serait intéressant. Pour nos expérimentations, nous utiliserons des jeux de données couramment utilisés dans l'étude des SR. Par exemple le jeu de données MovieLens, contenant dix millions de notes réparties sur 10 681 films et 71 567 utilisateurs, pourrait permettre d'évaluer les aspects flux de données et de passage à l'échelle.

6. Bibliographie

Bubeck S., Cesa-Bianchi N., « Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems », *Foundations and Trends in Machine Learning*, vol. 5, n° 1, p. 1-122, 2012.

Koren Y., « Collaborative Filtering with Temporal Dynamics », *Commun. ACM*, vol. 53, n° 4, p. 89–97, avril, 2010.

Li L., Chu W., Langford J., Schapire R. E., « A Contextual-Bandit Approach to Personalized News Article Recommandation », *Proc. of 19th International World Wide Web Conference*, April, 2010.

Montaner M., « A Taxonomy of Personalized Agents on the Internet », *Artificial Intelligence Review*, 2003.

Radlinski F., Kleinberg R., Joachims T., « Learning Diverse Rankings with Multi-Armed Bandits », *Proc. of 25th International Conference on Machine Learning*, 2008.

Ricci F., Rokach L., Shapira B., « Introduction to Recommender Systems Handbook. », *in* F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (éd.), *Recommender Systems Handbook*, Springer, p. 1-35, 2011.

Rubens N., Kaplan D., Sugiyama M., « Active Learning in Recommender Systems », *in* F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (éd.), *Recommender Systems Handbook*, Springer, p. 735-767, 2011.