



Systemes de Recommandation

De la Factorisation de Matrices aux Bandits



Jérémie MARY

SequeL Team

Lille Nord-Europe

MERCI
de ~~ne pas~~
m'interrompre
quand je parle à
MOI-MÊME

Overview

Méthodes Offline (CF, Factorisation)

Méthodes Online (Exploration/Exploitation)

Les systèmes de recommandation modernes

But : dans un **contexte imposé, mettre en avant le meilleur objet** possible dans une perspective à court ou long terme.

En particulier en s'appuyant sur des méthodes de :

Machine Learning à Large échelle ou non, online/offline,
exploration exploitation, ...

optimisation multi-critère CTR, fidélisation, analyse du ROI,
découverte de produit, gestion des
inventaires, auto-promo, ...

Modélisation des usagers : construction de features pour des
profils. Analyse de contenus (Langue naturelle) et de
réseaux sociaux.

Exemples de cas

Simple : Une partie de l'affichage d'un site web est constitué de produits variables et personnalisés. Comment optimiser le taux de clic (Click Through Rate CTR). Je possède une large base d'utilisateurs par rapport au nombre de produits (ex: Amazon).

Avancé : Les objets affichés proviennent de contrats (imposés) et je peux acheter de l'espace d'affichage sur une place de marché aux enchères. Les campagnes ont des limites de budget et de temps, le monde est dynamique.

Science-Fiction : Je dois effectuer la même tâche pour plusieurs zone de ma page. Des interactions peuvent exister entre les zones. Je veux fidéliser dans le temps mon utilisateur tout en maximisant mon revenu.

Netflix Challenge (2006)

Dotation d'un million de \$ -*gagné en 2009 par la BellKor Team*- portant sur la recommandation de films à des utilisateurs d'un service de VOD [4].

- ▶ **Offline**,
- ▶ 17 770 films,
- ▶ 480 189 utilisateurs,
- ▶ **100 480 507 notations** datées,
- ▶ Évaluation sur la capacité à deviner 1 408 342 notes tenues secrètes.

UNDERSTANDING ONLINE STAR RATINGS:



Formulation matricielle

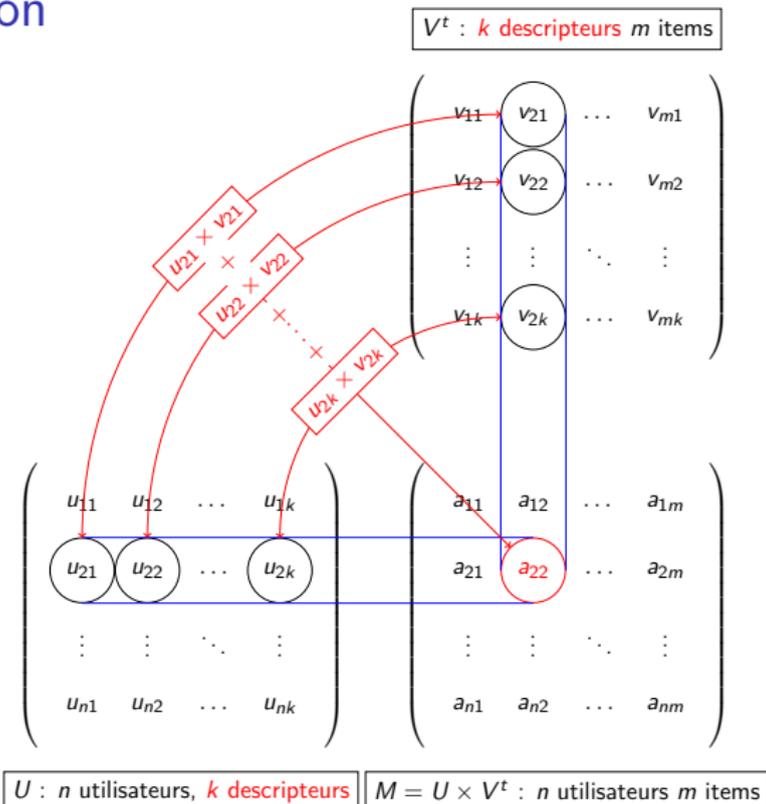
	$Item_1$	$Item_2$	$Item_3$...		$Item_m$
$User_1$	5	4	?	...	?	?
$User_2$	5	4	4	...	1	5
$User_3$?	?	?	...	?	?
\vdots	?	?	1	...	?	5
$User_n$?	?	2	...	4	5

Décomposition

- ▶ Il est considéré comme raisonnable M^* n'est pas de rang plein (i.e il existe des utilisateurs qui sont des combinaisons linéaires des autres).
- ▶ L'on cherche M^* comme le produit de deux matrices, l'une décrivant les utilisateurs, l'autre les items, le tout dans un même espace de dimension k (paramètre).

$$M^* = U \cdot V^t + \varepsilon$$

Décomposition



Décomposition

- ▶ Algébriquement si toutes les valeurs étaient connues, avec une contrainte sur le rang, la meilleure approximation au sens de la RMSE est la **SVD tronquée** à la dimension k . Cela est équivalent à minimiser :

$$\sum_{i=1}^{\#Users} \sum_{j=1}^{\#Items} (m_{ij} - \mathbf{u}_i \cdot^t \mathbf{v}_j)^2$$

SVD avec valeurs manquantes

En pratique, des valeurs sont manquantes, ce qui pousse à régulariser, l'on préfère donc minimiser :

$$\sum_{i,j \setminus m_{ij} \text{ connu}} (m_{ij} - \mathbf{u}_i \cdot^t \mathbf{v}_j - \text{biais}_{i,j})^2 + \lambda(\|\mathbf{U}\|^2 + \|\mathbf{V}\|^2)$$

Avec \mathbf{u}_i (resp \mathbf{v}_j) la i^e ligne de U (resp j^e colonne de V) et λ un paramètre réel.

La descente se fait par **gradient stochastique** en itérant sur les valeurs connues de la matrice (ou par alternating least squares - ALS).

ALS-WR

[10] regularise avec:

$$\Omega(\mathbf{U}, \mathbf{V}) \stackrel{\text{def}}{=} \sum_i \#\mathcal{J}(i) \|\mathbf{u}_i\|^2 + \sum_j \#\mathcal{I}(j) \|\mathbf{v}_j\|^2$$

$\mathcal{J}(i)$ (resp $\mathcal{I}(j)$) est le nombre de notes attribuées par l'utilisateur i (resp à l'item j).

Très efficace sur le jeu de données Netlix.

Questions ouvertes (1)

- ▶ Problème du **cold start** (users et items), utilisation d'un ε -greedy.
- ▶ choix de k , présence de minimums locaux (gradient stochastic sensible au pas et à l'initialisation).
- ▶ **Garanties** en $\frac{\|M\|_*}{m\sqrt{np}} + \frac{1}{np}$ (p proportion d'observations) sur l'erreur RMSE pour une distribution des valeurs manquantes **iid** [3].
- ▶ **Méthodologie d'évaluation** (RMSE) discutable (en pratique prédire 1 au lieu de 3 est moins gênant que 4 au lieu de 5), question de la gestion des lignes et colonnes lourdes. A noter Le prix a été gagné avec une RMSE de 0.8567 (pour des notes de 1 à 5).
- ▶ Recommandation sur des séquences (ex: écoute de musiques - cf Deezer).

Questions ouvertes (2)

- ▶ L'**introduction de biais** est délicate, conditionne fortement la performance et influe sur l'algorithme d'optimisation : comment comptabiliser un click, une mise en panier, ... ?
- ▶ Factorisation non négatives ?
- ▶ Ne modélise pas complètement le comportement d'un client (cf A/B testing par Netflix). Les personnes ont tendance à ne pas noter ce qu'elles n'aiment pas. Gestion des feedbacks implicites: optimisation de rangs [9].
- ▶ **Inclusion du temps** problématique (décomposition de tenseurs [8].).
- ▶ Problème du besoin d'oubli/**respect de la vie privée**.

Implémentation

Mahout propose une implémentation GPL de décomposition pouvant utiliser sur un cluster Hadoop. 4-Tell est une version commerciale par Netflix.



PROPACK (BSD - Stanford) implémente une version sans les termes de biais avec une descente basée sur une variante de Lanczos. **Vowpal Wabbit** propose également une implémentation depuis quelques semaines.

Bandits et regret cumulé



Objectif : minimiser le **regret cumulé** au pas de temps T .

$$R_T = \sum_{i=1}^T (\mu^* - \mu_{\pi_i})$$

Ou μ^* est l'espérance de gain du meilleur bandit et μ_{π_i} l'espérance du bandit choisi par la politique au pas de temps i .

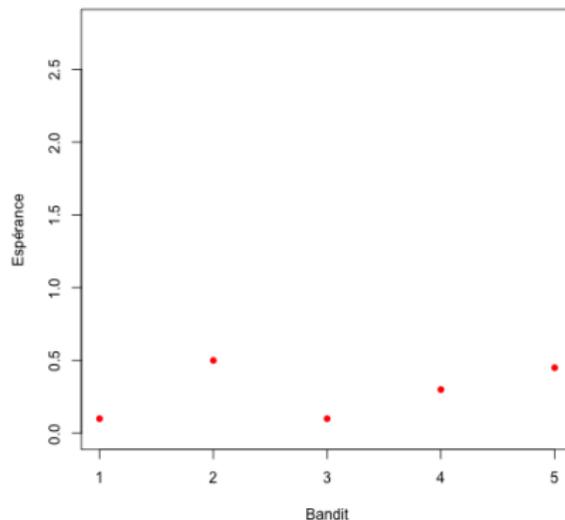
Regret cumulé

- ▶ Le regret cumulé croît linéairement en T si l'on utilise une politique aléatoire uniforme.
- ▶ Une borne inf (asymptotique) :

$$\liminf_{T \rightarrow \infty} \frac{E(R_T)}{\log(T)} \geq \sum_{i=1}^k \frac{\Delta_i}{KL(P_i || P^*)}$$

- ▶ Des algorithmes : ϵ_n -greedy, EXT3, UCB, UCB-v, KL-UCB, **Thompson Sampling**, Gitting indexes, ... réalisant la borne (parfois à une constante près et pour t assez grand).
- ▶ Take at home : **l'exploration a un coût logarithmique en T .**

Principe général pour les méthodes optimistes



Exemple avec la borne UCB : $\hat{\mu}_i + \alpha \cdot \sqrt{\frac{\log(T)}{n_i}}$

Bandits contextuels

Web + Ads Servers Ensemble de pubs A

User in Context x

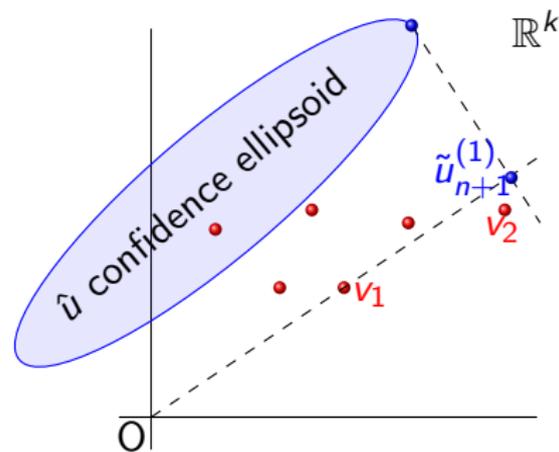


Le choix de la publicité affichée se fait selon une politique π .

Le **contexte est connu** et contient entre autres le profil de l'utilisateur, la page visitée ainsi que la position dans la page.

Besoin d'évaluation Offline alors que **le problème est Online**. Dans les cas réels il faut ajouter une phase d'enchère pour les emplacements.

LinUCB / Kernel UCB



$$\operatorname{argmax}_j \hat{\mathbf{u}} \cdot \mathbf{v}_j(t)^T + \alpha \sqrt{\mathbf{v}_j(t)^T \mathbf{A}^{-1} \mathbf{v}_j(t)^T},$$

α paramètre d'exploration

$$\mathbf{A} = \sum_{t'=1}^{t-1} \mathbf{v}_{j_{t'}}(t') \cdot \mathbf{v}_{j_{t'}}(t')^T + \mathbf{Id}.$$

Partial analysis in [1]. Version kernel at ICML'13.

ICML'11 Challenge

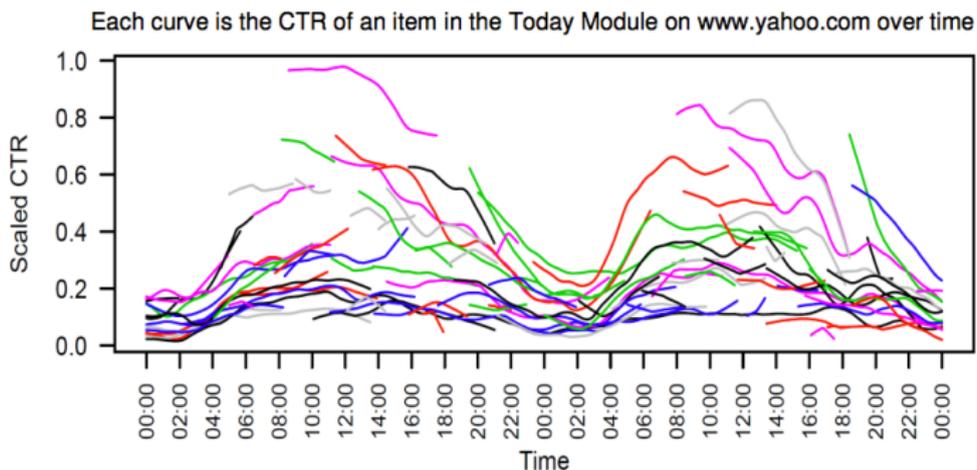
- ▶ Présentation de manière répétée de 6 tuples (contexte, action, récompense) en cachant l'information de récompense. Contexte de dimension 99 et hétérogène.
- ▶ Choisir un des tuples dans le but de **maximiser la somme des récompenses**. La récompense associée au tuple choisie est révélée et l'on continue avec 6 nouveaux tuples.
- ▶ Méthode gagnante ayant pour clé de **s'adapter rapidement aux changements** (larges updates en cas de "surprise")...
- ▶ ... mais sans jamais tenir compte des actions. Il est plus important de **reconnaitre les "bons" utilisateurs** que de personnaliser [7].

ICML'12 Challenge

Yahoo! provided some data of their frontpage with **random uniform allocation** of news.

Context (137 features)	Pool of current articles (around 30)	displayed article	Clic
x_1	P_1	a_1	r_1
\vdots	\vdots	\vdots	\vdots
x_T	P_T	a_T	r_T

CTR over Time



Graphique de Bee-Chung Chen, présence d'effets de l'heure, de nouvelles majeures et de durées variables.

Calcul du score

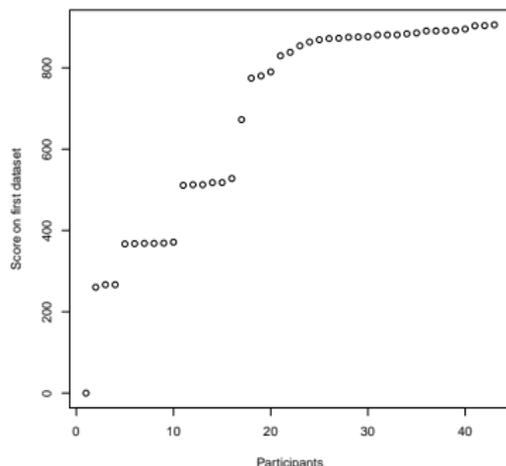
Pour une politique d'allocation π l'estimateur du CTR est calculé par rejet comme proposé par [6]. Une trentaine de news sélectionnables à chaque pas de temps.

- ▶ Phase 1 : plusieurs soumissions d'algorithmes autorisée avec un retour sur la performance.
- ▶ Phase 2 : nouveau jeu de données et une unique soumission.

Le score affiché est le $CTR * 10\ 000$.

Resultats de la Phase 1

NAME	AFFILIATION	BEST SCORE (CTR * 10 000)	RANK
Ku-Chun	NTU	905.9	1
tviro	MIT	903.9	2
edjoesu	MIT	903.4	3
Francis	ULg	895.4	4
jamh	UCM	891.9	5
exploreit	untitled	891.4	6
EpsilonGreedyRocks	U of A	890.9	7

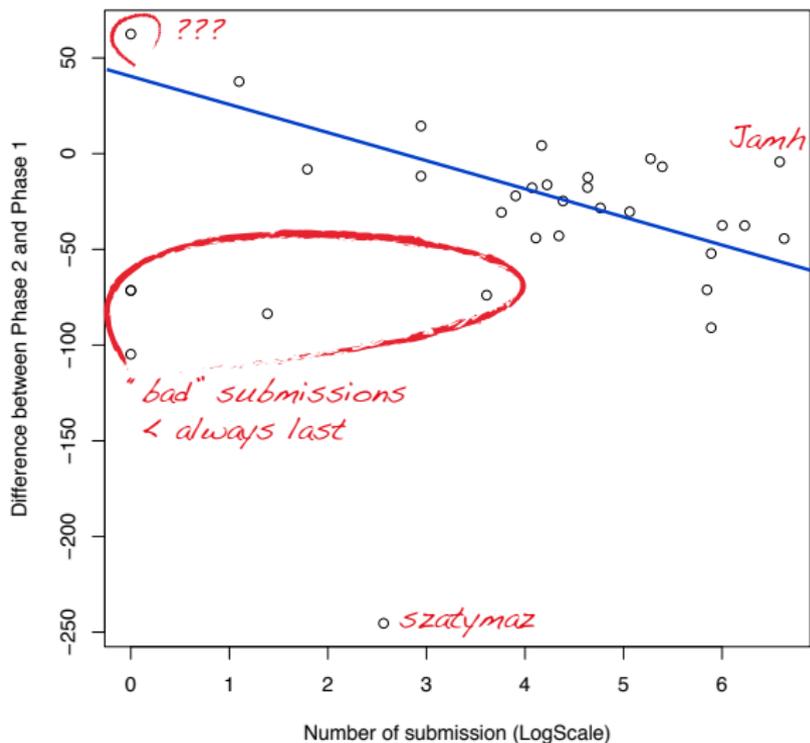


Complete list:

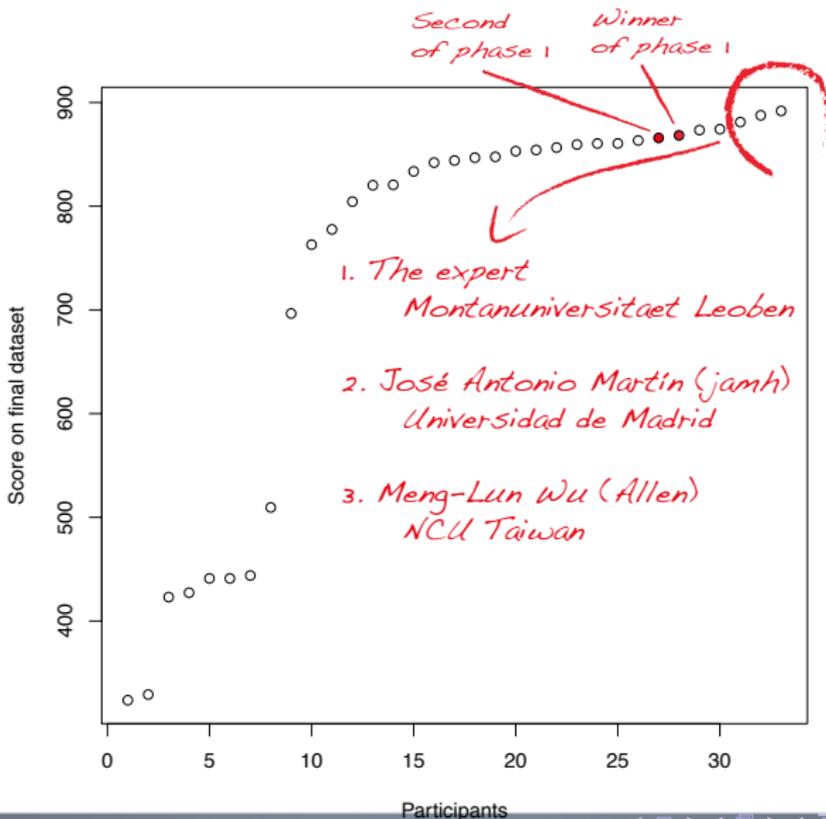
<http://explochallenge.inria.fr/leaderboard/>

Contexte pas toujours utilisé par les candidats !.

Difference entre les Scores en Phase 1 et 2



Résultats Phase 2



ICML'12 Challenge - UCB-v

Décrit dans [2]

$$\hat{\mu} = \mu + \sqrt{\frac{c \cdot \mu \cdot (1 - \mu) \cdot \log(t)}{n}} + c \cdot \left(\frac{0.5 - \mu}{n}\right) \log(t)$$

avec, t pas de temps courant, n nombre de fois ou l'article a été choisi, μ moyenne empirique, c constante à calibrer.

Open questions

- ▶ Résultats pour des valeurs de t petites,
- ▶ Analyse complète du Thompson sampling (faite par E. Kofmann dans le cas Bernouilli [5])
- ▶ Utilisation de données hétérogènes,
- ▶ Cas des grand nombres de bras (ajustement du β de UCB-air, liens avec la théorie des valeurs extrêmes),
- ▶ Construction à la volée du contexte en utilisant les observations (i.e rapprocher de la factorisation de matrices).
- ▶ Optimisation d'un contrôle en ligne sans forte dégradation.

Autres Applications

- ▶ Optimisation des tests de type **A/B testing** (utilisé par Google entre autres pour de l'allocation de ressources),
- ▶ Gestion de tournois et classement (Bayes Elo),
- ▶ Optimisation des **recherches arborescentes** (variante UCT), a servi dans l'équipe à développer des IA pour le Go et le Poker, utilisé pour des politiques de routage dynamique.

Software

- ▶ CrazyStone par Rémi Coulom (Jeu de Go)
- ▶ Vowpal Wabbit (encore) destiné à faire des séparations linéaire sur des Terraoctets de données (utile pour LinUCB). Bandits maintenant implémentés ainsi que des politiques d'évaluation.
- ▶ De manière générale ce sont des **algorithmes rapides et incrémentaux** (passage à l'échelle possible).

Bibliographie I



Yasin Abbasi-yadkori, David Pal, and Csaba Szepesvari.

Improved algorithms for linear stochastic bandits.

In [Advances in Neural Information Processing Systems 24th \(NIPS\)](#), pages 2312–2320, 2011.



Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári.

Exploration-exploitation tradeoff using variance estimates in multi-armed bandits.

[Theor. Comput. Sci.](#), 410(19):1876–1902, April 2009.



Sourav Chatterjee.

Matrix estimation by universal singular value thresholding.

[pre-print](#), 2012.



Gideon Dror, Noam Koenigstein, Yehuda Koren, and Markus Weimer.

The Yahoo! music dataset and kdd-cup'11.

In [Proceedings of KDDCup 2011](#), 2011.



E. Kaufmann, N. Korda, and R. Munos.

Thompson Sampling: An Asymptotically Optimal Finite Time Analysis.

[ArXiv e-prints](#), May 2012.



Lihong Li, Wei Chu, John Langford, and Xuanhui Wang.

Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms.

In [Proc. Web Search and Data Mining \(WSDM\)](#), pages 297–306. ACM, 2011.

Bibliographie II



O. Nicol, J. Mary, and Ph. Preux.

Icml exploration ans exploitation challenge: Keep it simple !

In [Journal of Machine Learning Research \(JMLR\)](#), 2012.



Steffen Rendle, Christoph Freudenthaler, and Lars S. Thieme.

Factorizing personalized Markov chains for next-basket recommendation.

In [Proceedings of the 19th international conference on World wide web, WWW '10](#), pages 811–820, New York, NY, USA, 2010. ACM.



Gábor Takács and Domonkos Tikk.

Alternating least squares for personalized ranking.

In [Proceedings of the sixth ACM conference on Recommender systems, RecSys '12](#), pages 83–90, New York, NY, USA, 2012. ACM.



Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan.

Large-scale parallel collaborative filtering for the netflix prize.

In [Proceedings of the 4th international conference on Algorithmic Aspects in Information and Management \(AAIM\)](#), pages 337–348, Berlin, Heidelberg, 2008. Springer-Verlag.

MERCI

Inria
INVENTEURS DU MONDE NUMÉRIQUE