

Efficient Context-Aware Sequential Recommender System

Leonardo Cella

Dipartimento di Informatica, Università degli Studi di Milano

Milan, Italy

leonardo.cella@unimi.it

ABSTRACT

Traditional collaborative filtering, and content-based approaches attempt to learn a static recommendation model in a batch fashion. These approaches are not suitable in highly dynamic recommendation scenarios, like news recommendation and computational advertisement. Due to this well-known limitation, in the last decade a lot of efforts have been spent over the study of online learning techniques. Currently, a lot of attention has been devoted to improvements on the theoretical guarantees, without caring too much about computational cost and memory footprint. However, in the era of big-data content features tend to be high-dimensional, which leads to a direct challenge for traditional on-line learning algorithms (e.g., multi-armed bandits) since these are mostly designed for low-dimensional feature spaces. In this work we face the aforementioned problem, investigating an approximated context-aware bandit learner. Our model takes into account the problem of finding the actual low-dimensional manifold spanned by data content-features. In particular, we propose to store the covariance matrix of the previously seen contexts in a compressed space, without losing too much in terms of recommendation quality. With this work we provide an overview over the main properties, describe the adopted techniques, and report on preliminary experimental results on a synthetic dataset. We also discuss a drawback of the proposed method that may appear in typical scenarios and suggest future research avenues.

CCS CONCEPTS

• **Computing methodologies** → **Online learning settings; Sequential decision making; Spectral methods;** • **Theory of computation** → **Online learning algorithms;**

KEYWORDS

Online learning theory, Incremental learning, Recommender Systems, Curse of dimensionality

ACM Reference Format:

Leonardo Cella. 2018. Efficient Context-Aware Sequential Recommender System. In *The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France*. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3184558.3191581>

This paper is published under the Creative Commons Attribution-NonCommercial-NoDerivs 4.0 International (CC BY-NC-ND 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW'18 Companion, April 23–27, 2018, Lyon, France

© 2018 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC BY-NC-ND 4.0 License.

ACM ISBN 978-1-4503-5640-4/18/04.
<https://doi.org/10.1145/3184558.3191581>

1 INTRODUCTION

The recommender system field has been largely studied since early 90s [18]. Recommendation engines are now an essential part of many on-line businesses such as Amazon [20] and Netflix [11]. Typical examples of modern recommendation problems are news articles [13] and computational advertisement [14]. Due to their sequential nature and the dependence on the current *context*, these application domains are not suited to standard batch algorithms that are commonly used in static settings. Static settings assumes a set of users with fixed preferences interacting with a fixed set of items. In this scenario, the well known cold-start issue, namely the lack of any information regarding the new user (or item), is usually addressed with *hybrid* approaches (e.g [2, 4]). As mentioned earlier, this static assumption is not always true: in online scenarios user preferences have to be computed sequentially according to previous choices. If we consider highly dynamic recommendation domains, such as newsfeeds and online advertising, the user preferences and item sets tend to be quite fluid. As a consequence, standard collaborative filtering methods as Matrix Factorization break down.

Multi-armed bandit model have been the state-of-the-art for modeling online scenarios for a decade. Recently, a lot of efforts have been spent in building context-aware bandit algorithms with strong theoretical guarantees. However, there is still a certain lack of attention on the scalability issues, which makes it difficult to use these algorithms in scenarios with large amount of content features. Common scenarios that may benefit from this solution are recommender systems that exploit a Reproduced Kernel Hilbert Space for deriving product similarities [3, 6], and more generally recommendation algorithms that operate in scenarios with many content features [13, 14].

We propose a sketched on-line contextual learner capable to preserve most of past spectral information while learning. More specifically, we introduce a first sketching model over a specific contextual learning algorithm, [16]. The adopted sketching technique not only has good empirical guarantees, but also strong theoretical properties that are suited to a future theoretical analysis. We describe preliminary results derived with different sketching configurations on a synthetic dataset. We finally give some intuitions on drawbacks and possible solutions to be taken into account in future investigations.

For any positive integer T , we will indicate with $[T]$ the set $1, \dots, T$. We indicate matrices with bold upper-case letters \mathbf{W} , vectors with bold lower-case letters \mathbf{x} . We denote with $\|\mathbf{x}\|_p$ the p -norm of the vector $\mathbf{x} \in \mathbb{R}^d$. Vectors are always column-vectors, thus \mathbf{x}^\top is a row-vector. The inner-product of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ is defined as $\langle \mathbf{x}, \mathbf{y} \rangle$. Finally, given a positive definite matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$, the matrix-norm of a vector $\mathbf{x} \in \mathbb{R}^d$ is defined as $\|\mathbf{x}\|_{\mathbf{W}} = \sqrt{\mathbf{x}^\top \mathbf{W} \mathbf{x}}$.

2 RELATED WORKS

The main goal of a well-tuned multi armed bandit model (MAB) is to manage the exploration-exploitation dilemma which typically arise in a sequential setting with partial feedback. To act eventually optimally, the learner *exploits* past observation to select the arm (item) which appears best. On the other hand, such arm could instead be suboptimal, due to lack of knowledge. To avoid selecting always a possible suboptimal arm, the learner has to consider also the option of *exploring* by selecting a so far unseen arm (e.g., an item whose feature vector is orthogonal to all the vectors selected until current trial) with the objective of gather more information about the new direction. Clearly, naive strategies of pure-exploration or pure-exploitation are far to be effective. The reason is that *exploration* is necessary to refine the learner estimation even if it may decrease the *short-term rewards*, while *exploitation* is necessary in order to trying selecting the optimal arm. This ability allow MAB algorithms to break-down also trendy deep-learning approaches when there is a continuous need of recommendations over cold-items. The MAB model was originally proposed and investigated by Robbins [15], attracting the interest of several researchers among different communities. Contextual MAB learners, which exploit the context as additional information in order to maximize their reward, are sequential decision-making models that extend Robbins' model and are largely adopted in the Recommender System community [9]. [1, 5, 10, 21] try to solve the personalized recommendation problem for fluid (new) users when no content user features are available (cold-user), with the aim of understanding as quickly as possible the class profile the user belongs to. Collaborative strategies [3, 7, 12, 22, 23] try instead to speed-up the learning process by collaboratively sharing information. In particular they strengthen the learning phase by agglomerating similar user profiles in clusters, requiring less rewards per user and less exploration.

In the big data era, it is common to deal with high-dimensional content information, while most algorithms developed so far are designed for low-dimensional data. This creates two issues. The first one are the high computational costs, the second one consists of having extremely large upper bounds over the error metric (regret) with high-dimensional data. The most popular contextual learners [9, 16] assume a linear relation governing the reward in terms of content features, and are based on ridge-regression for the estimation of the hidden vector parameter. This means that at each round, they invert a covariance matrix defined in the original space, an operation that is computationally very costly in most practical cases. More in detail, costs are linear in $O(Td)$, where T is the number of rounds, and d is the dimension of the feature space. As mentioned, the regret bounds are also defined in terms of the dimension of the feature space, and thus affected by the curse of dimensionality. In the machine learning community, similar problems have been already encountered in different settings, such as the online-newton-step algorithm [8, 17]. In these works the authors proposed compressing techniques to store the previous observed gradient vectors. As done in [8], we will adopt Frequent Directions [19] as compressing strategy. That consists of a deterministic sketching method that maintains the invariant that the last row of the stored matrix is always 0.

3 PROBLEM FORMULATION AND SETTING

In this section we formally present the sequential user recommendation problem, and we also make some examples in order to show how it could be adopted to solve the recommendation problem with fluid items.

Following previous work [9, 16] we call this multi-armed bandit model, *Contextual Bandit*. Formally, a contextual-bandit algorithm operates in sequential discrete rounds $t = 1, \dots, T$. In each trial t :

- The learner observes the current user u_t and a set of arms I_t together with their feature vectors $\mathbf{x}_{i,t} \forall i \in I_t$. The vector $\mathbf{x}_{i,t}$ embeds information of both user u_t and item i , and it will be referred to as *context*.
- Based on the payoffs observed in previous trials $[t - 1]$ with respect to the chosen contexts, the learner chooses the item to be recommended $i(t)$.
- According to the recommended product $\mathbf{x}_{i(t)}$, the learner observes a reward $r_{t,i(t)}$. The expected value on the reward depends on both the user and the recommended item. It is important to emphasize that in the **bandit** setting, the only observed reward per round is the one corresponding to the recommended arm, no feedback is given for unchosen arms $i \neq i(t)$.

In this setting the performance will be measured as $\sum_{t=1}^T r_{t,i(t)}$, corresponding to the cumulative reward observed over the T rounds. In particular we will compare that to the optimal reward $\sum_{t=1}^T r_{t,i^*}$, where i^* is the optimal arm at each round. Clearly our goal is to design a scalable strategy that maximizes the cumulative reward $\sum_{t=1}^T r_{t,i(t)}$. Since the optimal cumulative reward is always the same independently of the adopted learner, we consider the *regret* as the objective function to be *minimized*:

$$R_T = \sum_{t=1}^T (r_{t,i^*} - r_{t,i(t)}) \quad (1)$$

In our setting we will refer to a special case of the general contextual bandit problem known as *K-armed bandit* in which at each round, the set of available arms has fixed cardinality K .

Coming back to our recommendation problem, we can consider item contexts as the arm set. In particular each item context could be viewed as the concatenation of both real *context* information regarding the user and the environment, and pure content properties of the user and the respective item. As mentioned before the linearity assumption consists of assuming a linear function mapping the expected value of the reward (loss) with the item context. We indicate with θ^* the hidden vector that specifies this linear relation.

4 PROPOSED ALGORITHM

4.1 Optimal Algorithm

According to the notation of Section 1, we assume that the reward (loss) of an arm i is linear in its d -dimensional feature vector $\mathbf{x}_{t,i}$:

$$r_{t,i} = \mathbf{x}_{t,i}^\top \theta^* + \eta_t \quad (2)$$

where: $\theta^* \in \mathbb{R}^d$ is the unknown d -dimensional coefficient vector governing the linear mapping, η_t is a 1-subGaussian random-noise variable. Let $\mathbf{X}_t \in \mathbb{R}^{t \times d}$ be the matrix of observed contexts of $t \times d$ dimension, whose rows correspond to the $\mathbf{x}_{i(1)}, \dots, \mathbf{x}_{i(t)}$ contexts

Algorithm 1 Online Contextual Algorithm

Init: $\mathbf{b}_0 = \mathbf{0}$,
for $t = 1, 2, \dots, T$ **do**
 compute $\hat{\theta}$ as specified in Equation 4
 receive the set of contents $I_t := \{\mathbf{x}_{t,1}, \dots, \mathbf{x}_{t,K}\}$
 select arm i_t according to Equation 5
 observe reward r_t associated with arm i_t
 Design matrix and vector of rewards updates

observed until round t , and $\mathbf{r}_t \in \mathbb{R}^d$ the column-vector of corresponding feedbacks (e.g., click/no-click user feedback). Applying regularized least-square regression to (X_t, \mathbf{r}_t) , the unknown vector parameter estimate is defined as:

$$\hat{\theta}_t = (X_t^\top X_t + \mathbf{I}_d)^{-1} X_t^\top \mathbf{r}_t \quad (3)$$

where \mathbf{I}_d is the identity matrix with d components. As mentioned in [16], the exploration-exploitation management can be summarized with the following formula:

$$\mathbf{x}_{t+1} = \arg \max_{\mathbf{x}_i \in I_{t+1}} [\mu(\mathbf{x}_i) + \beta_t \sigma(\mathbf{x}_i, \mathbf{x}_{[t]})]$$

For the moment we can consider the first *exploitation* term $\mu(\mathbf{x}_i)$ as the current estimate of the mean reward observed, the second term $\sigma(\mathbf{x}_i, \mathbf{x}_{[t]})$ as the *exploration* bonus term times a time-dependent hyper-parameter β_t . Let $\boldsymbol{\eta}_t = (\eta_1, \dots, \eta_t)^\top$, $\mathbf{X} = \mathbf{X}_{1:t}$, thanks to the linearity assumption (Equation 2), the original form of the ridge-regression estimation for $\hat{\theta}$ can be rewritten as:

$$\hat{\theta}_t = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top (\mathbf{X} \boldsymbol{\theta}_* + \boldsymbol{\eta}_t) \quad (4)$$

where $(\mathbf{X} \boldsymbol{\theta}_* + \boldsymbol{\eta}_t)$ is the sequence of previously observed rewards. According to [16] the arm selection phase, could be written as:

$$\mathbf{x}_{(t)} = \arg \max_{\mathbf{x} \in I_t} \langle \mathbf{x}, \hat{\theta} \rangle + \log \left(\frac{t}{\delta} \right)^{\frac{1}{2}} \|\mathbf{x}\|_{V_t^{-1}} \quad (5)$$

where $\mathbf{V}_t = \mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_d$ and λ is a regularization parameter.

4.2 Efficiency via Sketching

The just presented model requires $\Omega(d^2)$ time and space for storing the covariance matrix \mathbf{V}_t , and has $\Omega(d^3)$ time complexity for its inverse computation. The adoption of sketching is motivated by its ability of maintaining an approximation of the *to-sketch* matrix \mathbf{X}_t , that we will denote as $\mathbf{B}_t \in \mathbb{R}^{m \times d}$, where m is the *sketch size* ($m \ll d$) whose optimal value is the data really spanned dimension.

$$\begin{aligned} \mathbf{V}_t &= \mathbf{V}_0 + \sum_{s=1}^{s=t} \mathbf{x}_s \mathbf{x}_s^\top \\ &= \mathbf{V}_0 + \mathbf{X}_t^\top \mathbf{X}_t \\ &\approx \mathbf{V}_0 + \mathbf{B}_t^\top \mathbf{B}_t \end{aligned} \quad (6)$$

Regarding its inversion, we leverage on the Sherman-Morrison formula, with $\mathbf{V}_0 = \lambda \mathbf{I}_d$:

$$\begin{aligned} V_t^{-1} &= \frac{1}{\lambda} (\mathbf{I}_d - \mathbf{B}_t^\top (\lambda \mathbf{I}_m + \mathbf{B}_t \mathbf{B}_t^\top)^{-1} \mathbf{B}_t) \\ &= \frac{1}{\lambda} (\mathbf{I}_d + \mathbf{B}_t^\top \mathbf{H}_t \mathbf{B}_t) \end{aligned} \quad (7)$$

To see why this gives an efficient algorithm we can notice that

Table 1: Statistic of used datasets

Dataset - name	original-dimension	sketched-dimension
Synthetic 1 - s1	100	70
Synthetic 2 - s2	100	80

Table 2: Policies performance measured as regret mean \pm regret standard deviation.

Regret:	LinUCB	Sketched LinUCB
s1	1762.8480 \pm 29.8683	1936.7079 \pm 36.0234
s2	1764.7769 \pm 7.5479	1864.7969 \pm 13.9649

Equation 4 becomes: $\hat{\theta}_t = \frac{1}{\lambda} (\mathbf{X}^\top \mathbf{Y}_t + \mathbf{B}_t^\top \mathbf{H}_t \mathbf{B}_t^\top \mathbf{X}^\top \mathbf{Y}_t)$. Regarding the arm selection Equation 5 we now have:

$$\mathbf{x}_t = \arg \max_{\mathbf{x} \in I_t} \langle \mathbf{x}, \frac{1}{\lambda} (\mathbf{X}^\top \mathbf{Y}_t + \mathbf{B}_t^\top \mathbf{H}_t \mathbf{B}_t^\top \mathbf{X}^\top \mathbf{Y}_t) \rangle + \log \left(\frac{t}{\delta} \right)^{\frac{1}{2}} \|\mathbf{x}\|_{V_{t-1}}$$

The operations involving $\mathbf{B}_t \mathbf{x}_t$ require only $O(md)$ time, while matrix vector product with \mathbf{H}_t requires only $O(d^2)$. We call this algorithm **Sketched OFUL**. For the experiment we have adopted [19] as sketching technique, its main drawback is that requires a predefined sketch-size. [17] proposes a new sketching learning, based on ridge leverage scores that avoids high-regret when the dataset rank is bigger than the predefined sketched-dimension. It adaptively increases the size of its sketch, providing a favorable regret-performance trade-off. This is advantageous when the context vectors do not span a low-dimensional space but range uniformly over the whole d -dimensional space.

5 EXPERIMENTAL ANALYSIS

We compare our algorithm to [16] without sketching. The comparison is made on an artificially generated dataset. We adopt two different dataset versions according to the data really-spanned manifold dimension (Table 1). All of them have been used to evaluate the algorithm over $T = 5000$ trials comparing to the performance obtained by the original linear contextual learner, that is the objective of our approximation. The artificial contexts have been generated starting from a low-dimensional manifold, expressed in terms of orthonormal basis. This basis has been enlarged to generate the bigger d -dimensional space, by linear combinations of its vectors. All the context vectors have finally been normalized according to the l2-norm. The same holds for the generation of the hidden vector $\boldsymbol{\theta}^* \in \mathbb{R}^d$. Finally, the rewards have been generated as $\mathbf{x}^\top \boldsymbol{\theta}^* + \eta_{i,t}$ where $\eta_{i,t}$ is random noise with 0 mean and 0.5 variance. The performance have been measured in terms of difference between the best expectation, defined as the cumulative reward obtained knowing the hidden vector $\boldsymbol{\theta}^* \in \mathbb{R}^d$ and the cumulative reward collected by the learning policy:

$$\sum_{t=1}^T \arg \max_{i \in I(t)} (\mathbf{x}_i^\top \boldsymbol{\theta}^* + \eta_i) - r_{policy(t)} \quad (8)$$

The results are shown in Table 2, and Figures 1, 2. In particular, we can notice that the empirical regret achieved by the sketched learner is close to the original one.

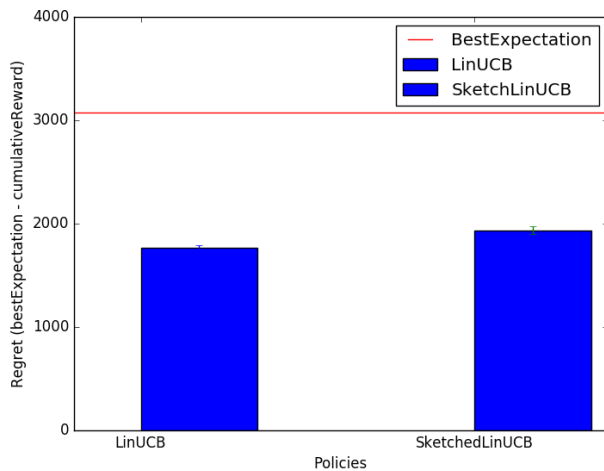


Figure 1: Mean regret obtained with compression ratio of 30%. On the left the mean regret obtained by the original contextual learner, on the right the one obtained by the sketched version.

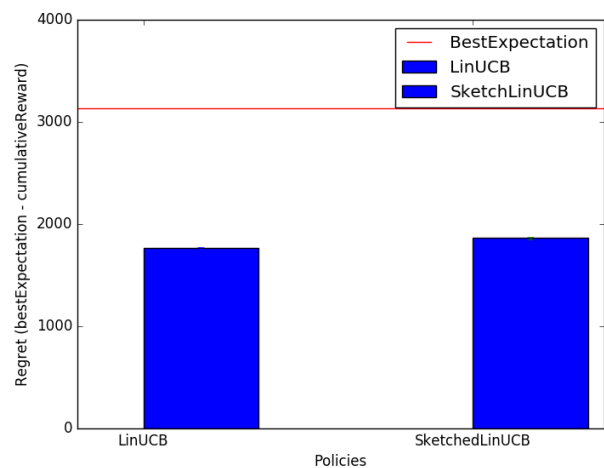


Figure 2: Mean regret obtained with compression ratio of 20%. On the left the mean regret obtained by the original contextual learner, on the right the one obtained by the sketched version.

6 CONCLUSIONS

We have initiated an investigation of approximation in contextual linear bandit algorithms operating in relevant scenarios where vendors acquire and maintain a large amount of content in their repository, or may be using a Reproduced Kernel Hilbert Space for storing similarities (which, theoretically, may have infinitely many feature dimensions). We have proposed the adoption of a deterministic sketching technique for compressing the original feature space. We carried out a preliminary experimental evaluation with very encouraging results by using high-dimensional artificially generated data. Future investigations regard the evaluation over real

dataset, the study of the theoretical guarantees, and the adoption of size-adaptive sketching techniques as in [17].

REFERENCES

- [1] Li Zhou; Emma Brunskill. July 2016. Latent contextual bandits and their application to personalized recommendations for new users. In *IJCAI'16 Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. 3646–3653.
- [2] Mattia Bianchi; Federico Cesaro; Filippo Ciceri; Mattia Dagrada; Alberto Gasparin; Daniele Grattarola; Ilyas Inajjar; Alberto Maria Metelli; Leonardo Cella. August 2017. Content-Based approaches for Cold-Start Job Recommendations. In *RecSys'17, Proceeding RecSys Challenge '17 Proceedings of the Recommender Systems Challenge 2017*. <https://doi.org/10.1145/3124791.3124793>
- [3] Leonardo Cella; Romaric Gaudel; Paolo Cremonesi. August 2017. Kernelized Collaborative Contextual Bandits. In *In Proceedings of RecSys 2017 Posters*.
- [4] Leonardo Cella; Stefano Cereda; Massimo Quadrona; Paolo Cremonesi. July 2017. Deriving Item Features Relevance from Past User Interactions. In *UMAP'17 Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. 275–279. <https://doi.org/10.1145/3079628.3079695>
- [5] Philippe Preux Cricia Z. Felicio; Klerisson V.R. Paixao; Celia A.Z. Barcelos. July 2017. A Multi-Armed Bandit Model Selection for Cold-Start User Recommendation. In *UMAP'17 Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. 32–40. <https://doi.org/10.1145/3079628.3079681>
- [6] Michal Valko; Nathaniel Korda; Remi Munos; Ilias Flouounas; Nelo Cristianini. August 2013. Finite-Time Analysis of Kernelised Contextual Bandits. In *UAI13 Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*. 654–663.
- [7] Shuai Li; Alexandros Karatzoglou; Claudio Gentile. July 2016. Collaborative Filtering Bandits. In *SIGIR '16 Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*.
- [8] Haipeng Luo; Alekh Agarwal; Nicolo Cesa-Bianchi; John Langford. December 2016. Efficient Second Order Online Learning by Sketching. In *NIPS'16 30th Conference on Neural Information Processing Systems*.
- [9] Lihong Li and Robert E. Schapire Wei Chu, John Langford. April 2010. A contextual-bandit approach to personalized news article recommendation. In *WWW'10 Proceedings of the 19th international conference on World wide web*. 661–670. <https://doi.org/10.1145/1772690.1772758>
- [10] Branislav Kveton; Csaba Szepesvari; Anup Rao; Zheng Wen; Yasin Abbasi-Yadkori; S. Muthukrishnan. December 2017. Stochastic Low-Rank Bandits. In *arXiv:1712.04644*.
- [11] James Bennett; Stan Lanning; Netflix Netflix. 2007. The Netflix Prize (2007). In *In KDD Cup and Workshop in conjunction with KDD*.
- [12] Giovanni Zappella Nicolo Cesa-Bianchi, Claudio Gentile. December 2013. A Gang of Bandits. In *NIPS'13 Advances in Neural Information Processing Systems*. 737–745.
- [13] Abhinandan S. Das; Mayur Datar; Ashutosh Garg; Shyam Rajaram. [n. d.]. Google news personalization: scalable online collaborative filtering. In *WWW '07 Proceedings of the 16th international conference on World Wide Web*. 271.
- [14] Aris Anagnostopoulos; Andrei Z. Broder; Evgeniy Gabrilovich; Vanja Josifovski; Lance Riedel. [n. d.]. Just-in-time contextual advertising. In *CIKM '07 Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*.
- [15] Herbert Robbins. 1952. *Some aspects of the sequential design of experiments*. Vol. 58. Bulletin of the American Mathematical Society, 527–535.
- [16] Yasin Abbasi Yadkori; David Pal; Csaba Szepesvari. December 2011. Improved algorithms for linear stochastic bandits. In *NIPS'11 Proceedings of the 24th International Conference on Neural Information Processing Systems*.
- [17] Daniele Calandriello; Alessandro Lazaric; Michal Valko. 2017. Second-Order Kernel Online Convex Optimization with Adaptive Sketching. In *ICML'17 Proceedings of the 34th International Conference on Machine Learning*.
- [18] Paul Resnick; Hal R. Varian. March 1997. *Recommender Systems*. Vol. 40. Magazine Communications of the ACM, 56–58. <https://doi.org/10.1145/245108.245121>
- [19] Mina Ghashami; Edo Liberty; Jeff M. Phillips; David P. Woodruff. September 2016. Frequent Directions: simple and deterministic matrix sketching. (September 2016), 1762–1792 pages.
- [20] G. Linden ; B. Smith ; J. York. January 2003. Amazon.com recommendations: item-to-item collaborative filtering. In *IEEE Internet Computing*. 76 – 80. <https://doi.org/10.1109/MIC.2003.1167344>
- [21] Aditya Gopalan; Odalric-Ambrym Maillard; Mohammadi Zaki. September 2016. Low-rank Bandits with Latent Mixtures. In *arXiv:1609.01508*.
- [22] Claudio Gentile; Shuai Li; Giovanni Zappella. June 2014. Online clustering of bandits. In *ICML '14 Proceedings of the 31st International Conference on International Conference on Machine Learning*. 757–765.
- [23] Claudio Gentile; Shuai Li; Purushottam Kar; Alexandros Karatzoglou; Evans Etrue; Giovanni Zappella. February 2017. On Context-Dependent Clustering of Bandits. In *Proceedings of the 34th International Conference on Machine Learning*.