Stochastic Rank-1 Bandits

Sumeet Katariya
Department of ECE
University of Wisconsin-Madison
katariya@wisc.edu

Branislav Kveton
Adobe Research
San Jose, CA
kveton@adobe.com

Csaba Szepesvári
Department of Computing Science
University of Alberta
szepesva@cs.ualberta.ca

Claire Vernade
Telecom ParisTech
Paris, France
claire.vernade@telecom-paristech.fr

Zheng Wen
Adobe Research
San Jose, CA
zwen@adobe.com

Abstract

We propose stochastic rank-1 bandits, a class of online learning problems where at each step a learning agent chooses a pair of row and column arms, and receives the product of their values as a reward. The main challenge of the problem is that the individual values of the row and column are unobserved. We assume that these values are stochastic and drawn independently. We propose a computationally-efficient algorithm for solving our problem, which we call `Rank1Elim`. We derive a $O((K + L)(1/\Delta) \log n)$ upper bound on its n-step regret, where K is the number of rows, L is the number of columns, and Δ is the minimum of the row and column gaps; under the assumption that the mean row and column rewards are bounded away from zero. To the best of our knowledge, we present the first bandit algorithm that finds the maximum entry of a rank-1 matrix whose regret is linear in $K + L$, $1/\Delta$, and $\log n$. We also derive a nearly matching lower bound. Finally, we evaluate `Rank1Elim` empirically on multiple problems. We observe that it leverages the structure of our problems and can learn near-optimal solutions even if our modeling assumptions are mildly violated.

1 Introduction

We study the problem of finding the maximum entry of a stochastic rank-1 matrix from noisy and adaptively-chosen observations. This problem is motivated by two problems, ranking in the position-based model [27] and online advertising.

The *position-based model (PBM)* [27] is one of the most fundamental click models [5], a model of how people click on a list of K items out of L. This model is defined as follows. Each *item* is associated with its *attraction* and each *position* in the list is associated with its *examination*. The attraction of any item and the examination of any position are i.i.d. Bernoulli random variables. The item in the list is *clicked* only if it is attractive and its position is examined. Under these assumptions, the pair of the item and position that maximizes the probability of clicking is the maximum entry of a rank-1 matrix, which is the outer product of the attraction probabilities of items and the examination probabilities of positions.

As another example, consider a marketer of a product who has two sets of actions, K population *segments* and L marketing *channels*. Given a product, some segments are *easier to market to* and some channels are *more appropriate*. Now suppose that the conversion happens only if both actions are successful and that the successes of these actions are independent. Then similarly to our earlier example, the pair of the population segment and marketing channel that maximizes the conversion rate is the maximum entry of a rank-1 matrix.

We propose an online learning model for solving our motivating problems, which we call a *stochastic rank-1 bandit*. The learning agent interacts with our problem as follows. At time t, the agent selects a pair of row and column arms, and receives the product of their individual values as a reward. The values are stochastic, drawn independently, and not observed. The goal of the agent is to maximize its expected cumulative reward, or equivalently to minimize its expected cumulative regret with respect to the optimal solution, the most rewarding pair of row and column arms.

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

We make five contributions. First, we precisely formulate the online learning problem of *stochastic rank-1 bandits*. Second, we design an elimination algorithm for solving it, which we call `Rank1Elim`. The key idea in `Rank1Elim` is to explore all remaining rows and columns randomly over all remaining columns and rows, respectively, to estimate their expected rewards; and then eliminate those rows and columns that seem suboptimal. This algorithm is computationally efficient and easy to implement. Third, we derive a $O((K+L)(1/\Delta)\log n)$ gap-dependent upper bound on its n-step regret, where K is the number of rows, L is the number of columns, and Δ is the minimum of the row and column gaps; under the assumption that the mean row and column rewards are bounded away from zero. Fourth, we derive a nearly matching gap-dependent lower bound. Finally, we evaluate our algorithm empirically. In particular, we validate the scaling of its regret, compare it to multiple baselines, and show that it can learn near-optimal solutions even if our modeling assumptions are mildly violated.

We denote random variables by boldface letters and define $[n] = \{1, \ldots, n\}$. For any sets A and B, we denote by A^B the set of all vectors whose entries are indexed by B and take values from A.

2 Setting

We formulate our online learning problem as a *stochastic rank-1 bandit*. An instance of this problem is defined by a tuple $(K, L, P_{\text{U}}, P_{\text{V}})$, where K is the number of rows, L is the number of columns, P_{U} is a probability distribution over a unit hypercube $[0,1]^K$, and P_{V} is a probability distribution over a unit hypercube $[0,1]^L$.

Let $(\mathbf{u}_t)_{t=1}^n$ be an i.i.d. sequence of n vectors drawn from distribution P_{U} and $(\mathbf{v}_t)_{t=1}^n$ be an i.i.d. sequence of n vectors drawn from distribution P_{V}, such that $\mathbf{u}_t$ and $\mathbf{v}_t$ are drawn independently at any time t. The learning agent interacts with our problem as follows. At time t, it chooses *arm* $(\mathbf{i}_t, \mathbf{j}_t) \in [K] \times [L]$ based on its history up to time t; and then *observes* $\mathbf{u}_t(\mathbf{i}_t)\mathbf{v}_t(\mathbf{j}_t)$, which is also its *reward*.

The goal of the agent is to maximize its expected cumulative reward in n steps. This is equivalent to minimizing the *expected cumulative regret* in n steps

$$R(n) = \mathbb{E}\left[\sum_{t=1}^{n} R(\mathbf{i}_t, \mathbf{j}_t, \mathbf{u}_t, \mathbf{v}_t)\right],$$

where $R(\mathbf{i}_t, \mathbf{j}_t, \mathbf{u}_t, \mathbf{v}_t) = \mathbf{u}_t(i^*)\mathbf{v}_t(j^*) - \mathbf{u}_t(\mathbf{i}_t)\mathbf{v}_t(\mathbf{j}_t)$ is the *instantaneous stochastic regret* of the agent at time t and

$$(i^*, j^*) = \operatorname*{arg\,max}_{(i,j) \in [K] \times [L]} \mathbb{E}\left[\mathbf{u}_1(i)\mathbf{v}_1(j)\right]$$

is the *optimal solution* in hindsight of knowing P_{U} and P_{V}. Since $\mathbf{u}_1$ and $\mathbf{v}_1$ are drawn independently, and $\mathbf{u}_1(i) \geq 0$ for all $i \in [K]$ and $\mathbf{v}_1(j) \geq 0$ for all $j \in [L]$, we get that

$$i^* = \operatorname*{arg\,max}_{i \in [K]} \mu\bar{u}(i), \quad j^* = \operatorname*{arg\,max}_{j \in [L]} \mu\bar{v}(j),$$

for any $\mu > 0$, where $\bar{u} = \mathbb{E}[\mathbf{u}_1]$ and $\bar{v} = \mathbb{E}[\mathbf{v}_1]$. This is the key idea in our solution.

Note that the problem of learning $\bar{u}$ and $\bar{v}$ from stochastic observations $\{\mathbf{u}_t(\mathbf{i}_t)\mathbf{v}_t(\mathbf{j}_t)\}_{t=1}^n$ is a special case of *matrix completion from noisy observations* [15]. This problem is harder than that of learning (i^*, j^*). In particular, the most popular approach to matrix completion is alternating minimization of a non-convex function [17], where the observations are corrupted with Gaussian noise. In contrast, our proposed algorithm is guaranteed to learn the optimal solution with a high probability, and does not make any strong assumptions on P_{U} and P_{V}.

3 Naive Solutions

Our learning problem is a KL-arm bandit with $K + L$ parameters, $\bar{u} \in [0,1]^K$ and $\bar{v} \in [0,1]^L$. The main challenge is to leverage this structure to learn efficiently. In this section, we discuss the challenges of solving our problem by existing algorithms. We conclude that a new algorithm is necessary and present it in Section 4.

Any rank-1 bandit is a multi-armed bandit with KL arms. As such, it can be solved by `UCB1` [2]. The n-step regret of `UCB1` in rank-1 bandits is $O(KL(1/\Delta)\log n)$. Therefore, `UCB1` is impractical when both K and L are large.

Note that $\log(\bar{u}(i)\bar{v}(j)) = \log(\bar{u}(i)) + \log(\bar{v}(j))$ for any $\bar{u}(i), \bar{v}(j) > 0$. Therefore, a rank-1 bandit can be viewed as a stochastic linear bandit and solved by `LinUCB` [8, 1], where the reward of arm (i,j) is $\log(\mathbf{u}_t(i)) + \log(\mathbf{v}_t(j))$ and its features $x_{i,j} \in \{0,1\}^{K+L}$ are

$$x_{i,j}(e) = \begin{cases} \mathbb{1}\{e = i\}, & e \leq K; \\ \mathbb{1}\{e - K = j\}, & e > K, \end{cases} \tag{1}$$

for any $e \in [K + L]$. This approach is problematic for at least two reasons. First, the reward is not properly defined when either $\mathbf{u}_t(i) = 0$ or $\mathbf{v}_t(j) = 0$. Second,

$$\mathbb{E}\left[\log(\mathbf{u}_t(i)) + \log(\mathbf{v}_t(j))\right] \neq \log(\bar{u}(i)) + \log(\bar{v}(j)).$$

Nevertheless, note that both sides of the above inequality have maxima at (i^*, j^*), and therefore `LinUCB` should perform well. We compare to it in Section 6.2.

Also note that $\bar{u}(i)\bar{v}(j) = \exp[\log(\bar{u}(i)) + \log(\bar{v}(j))]$ for $\bar{u}(i), \bar{v}(j) > 0$. Therefore, a rank-1 bandit can be viewed as a generalized linear bandit and solved by `GLM-UCB` [9], where the mean function is $\exp[\cdot]$ and the feature vector of arm (i,j) is in (1). This approach is not practical for three reasons. First, the parameter space is unbounded, because

$\log(\bar{u}(i)) \to -\infty$ as $\bar{u}(i) \to 0$ and $\log(\bar{v}(j)) \to -\infty$ as $\bar{v}(j) \to 0$. Second, the confidence intervals of `GLM-UCB` are scaled by the reciprocal of the minimum derivative of the mean function c_μ^{-1}, which can be very large in our setting. In particular, $c_\mu = \min_{(i,j) \in [K] \times [L]} \bar{u}(i) \bar{v}(j)$. In addition, the gap-dependent upper bound on the regret of `GLM-UCB` is $O((K+L)^2 c_\mu^{-2})$, which further indicates that `GLM-UCB` is not practical. Our upper bound in Theorem 1 scales much better with all quantities of interest. Third, `GLM-UCB` needs to compute the maximum-likelihood estimates of $\bar{u}$ and $\bar{v}$ at each step, which is a non-convex optimization problem (Section 2).

Some variants of our problem can be solved trivially. For instance, let $\mathbf{u}_t(i) \in \{0.1, 0.5\}$ for all $i \in [K]$ and $\mathbf{v}_t(j) \in \{0.5, 0.9\}$ for all $j \in [L]$. Then $(\mathbf{u}_t(i), \mathbf{v}_t(j))$ can be identified from $\mathbf{u}_t(i) \mathbf{v}_t(j)$, and the learning problem does not seem more difficult than a stochastic combinatorial semi-bandit [20]. We do not focus on such degenerate cases in this paper.

4 `Rank1Elim` Algorithm

Our algorithm, `Rank1Elim`, is shown in Algorithm 1. It is an elimination algorithm [3], which maintains `UCB1` confidence intervals [2] on the expected rewards of all rows and columns. `Rank1Elim` operates in stages, which quadruple in length. In each stage, it explores all remaining rows and columns randomly over all remaining columns and rows, respectively. At the end of the stage, it eliminates all rows and columns that cannot be optimal.

The eliminated rows and columns are tracked as follows. We denote by $\mathbf{h}_\ell^{\mathrm{U}}(i)$ the index of the most rewarding row whose expected reward is believed by `Rank1Elim` to be at least as high as that of row i in stage ℓ. Initially, $\mathbf{h}_0^{\mathrm{U}}(i) = i$. When row i is eliminated by row $\mathbf{i}_\ell$ in stage ℓ, $\mathbf{h}_{\ell+1}^{\mathrm{U}}(i)$ is set to $\mathbf{i}_\ell$; then when row $\mathbf{i}_\ell$ is eliminated by row $\mathbf{i}_{\ell'}$ in stage $\ell' > \ell$, $\mathbf{h}_{\ell'+1}^{\mathrm{U}}(i)$ is set to $\mathbf{i}_{\ell'}$; and so on. The corresponding column quantity, $\mathbf{h}_\ell^{\mathrm{V}}(j)$, is defined and updated analogously. The *remaining rows and columns in stage* ℓ, $\mathbf{I}_\ell$ and $\mathbf{J}_\ell$, are then the unique values in $\mathbf{h}_\ell^{\mathrm{U}}$ and $\mathbf{h}_\ell^{\mathrm{V}}$, respectively; and we set these in line 7 of Algorithm 1.

Each stage of Algorithm 1 has two main steps: exploration (lines 9–20) and elimination (lines 22–41). In the row exploration step, each row $i \in \mathbf{I}_\ell$ is explored randomly over all remaining columns $\mathbf{J}_\ell$ such that its expected reward up to stage ℓ is at least $\mu \bar{u}(i)$, where μ is in (4). To guarantee this, we sample column $j \in [L]$ randomly and then substitute it with column $\mathbf{h}_\ell^{\mathrm{V}}(j)$, which is at least as rewarding as column j. This is critical to avoid $1/\min_{j \in [L]} \bar{v}(j)$ in our regret bound, which can be large and is not necessary. The observations are stored in *reward matrix* $\mathbf{C}_\ell^{\mathrm{U}} \in \mathbb{R}^{K \times L}$. As all rows are explored similarly, their expected rewards are scaled similarly, and this permits elimination. The column exploration step is analogous.

Algorithm 1 `Rank1Elim` for stochastic rank-1 bandits.

1: // Initialization
2: $t \leftarrow 1,\ \tilde{\Delta}_0 \leftarrow 1,\ \mathbf{C}_0^{\mathrm{U}} \leftarrow \{0\}^{K \times L},\ \mathbf{C}_0^{\mathrm{V}} \leftarrow \{0\}^{K \times L},$
3: $\mathbf{h}_0^{\mathrm{U}} \leftarrow (1, \ldots, K),\ \mathbf{h}_0^{\mathrm{V}} \leftarrow (1, \ldots, L),\ n_{-1} \leftarrow 0$
4:
5: **for all** $\ell = 0, 1, \ldots$ **do**
6: $\quad n_\ell \leftarrow \left\lceil 4 \tilde{\Delta}_\ell^{-2} \log n \right\rceil$
7: $\quad \mathbf{I}_\ell \leftarrow \bigcup_{i \in [K]} \{\mathbf{h}_\ell^{\mathrm{U}}(i)\},\ \mathbf{J}_\ell \leftarrow \bigcup_{j \in [L]} \{\mathbf{h}_\ell^{\mathrm{V}}(j)\}$
8:
9: $\quad$ // Row and column exploration
10: $\quad$ **for** $n_\ell - n_{\ell-1}$ times **do**
11: $\quad\quad$ Choose uniformly at random column $j \in [L]$
12: $\quad\quad j \leftarrow \mathbf{h}_\ell^{\mathrm{V}}(j)$
13: $\quad\quad$ **for all** $i \in \mathbf{I}_\ell$ **do**
14: $\quad\quad\quad \mathbf{C}_\ell^{\mathrm{U}}(i, j) \leftarrow \mathbf{C}_\ell^{\mathrm{U}}(i, j) + \mathbf{u}_t(i) \mathbf{v}_t(j)$
15: $\quad\quad\quad t \leftarrow t + 1$
16: $\quad\quad$ Choose uniformly at random row $i \in [K]$
17: $\quad\quad i \leftarrow \mathbf{h}_\ell^{\mathrm{U}}(i)$
18: $\quad\quad$ **for all** $j \in \mathbf{J}_\ell$ **do**
19: $\quad\quad\quad \mathbf{C}_\ell^{\mathrm{V}}(i, j) \leftarrow \mathbf{C}_\ell^{\mathrm{V}}(i, j) + \mathbf{u}_t(i) \mathbf{v}_t(j)$
20: $\quad\quad\quad t \leftarrow t + 1$
21:
22: $\quad$ // UCBs and LCBs on the expected rewards of all remaining rows and columns
23: $\quad$ **for all** $i \in \mathbf{I}_\ell$ **do**
24: $\quad\quad \mathbf{U}_\ell^{\mathrm{U}}(i) \leftarrow \frac{1}{n_\ell} \sum_{j=1}^{L} \mathbf{C}_\ell^{\mathrm{U}}(i, j) + \sqrt{\frac{\log n}{n_\ell}}$
25: $\quad\quad \mathbf{L}_\ell^{\mathrm{U}}(i) \leftarrow \frac{1}{n_\ell} \sum_{j=1}^{L} \mathbf{C}_\ell^{\mathrm{U}}(i, j) - \sqrt{\frac{\log n}{n_\ell}}$
26: $\quad$ **for all** $j \in \mathbf{J}_\ell$ **do**
27: $\quad\quad \mathbf{U}_\ell^{\mathrm{V}}(j) \leftarrow \frac{1}{n_\ell} \sum_{i=1}^{K} \mathbf{C}_\ell^{\mathrm{V}}(i, j) + \sqrt{\frac{\log n}{n_\ell}}$
28: $\quad\quad \mathbf{L}_\ell^{\mathrm{V}}(j) \leftarrow \frac{1}{n_\ell} \sum_{i=1}^{K} \mathbf{C}_\ell^{\mathrm{V}}(i, j) - \sqrt{\frac{\log n}{n_\ell}}$
29:
30: $\quad$ // Row and column elimination
31: $\quad \mathbf{i}_\ell \leftarrow \arg\max_{i \in \mathbf{I}_\ell} \mathbf{L}_\ell^{\mathrm{U}}(i)$
32: $\quad \mathbf{h}_{\ell+1}^{\mathrm{U}} \leftarrow \mathbf{h}_\ell^{\mathrm{U}}$
33: $\quad$ **for all** $i = 1, \ldots, K$ **do**
34: $\quad\quad$ **if** $\mathbf{U}_\ell^{\mathrm{U}}(\mathbf{h}_\ell^{\mathrm{U}}(i)) \leq \mathbf{L}_\ell^{\mathrm{U}}(\mathbf{i}_\ell)$ **then**
35: $\quad\quad\quad \mathbf{h}_{\ell+1}^{\mathrm{U}}(i) \leftarrow \mathbf{i}_\ell$
36:
37: $\quad \mathbf{j}_\ell \leftarrow \arg\max_{j \in \mathbf{J}_\ell} \mathbf{L}_\ell^{\mathrm{V}}(j)$
38: $\quad \mathbf{h}_{\ell+1}^{\mathrm{V}} \leftarrow \mathbf{h}_\ell^{\mathrm{V}}$
39: $\quad$ **for all** $j = 1, \ldots, L$ **do**
40: $\quad\quad$ **if** $\mathbf{U}_\ell^{\mathrm{V}}(\mathbf{h}_\ell^{\mathrm{V}}(j)) \leq \mathbf{L}_\ell^{\mathrm{V}}(\mathbf{j}_\ell)$ **then**
41: $\quad\quad\quad \mathbf{h}_{\ell+1}^{\mathrm{V}}(j) \leftarrow \mathbf{j}_\ell$
42:
43: $\quad \tilde{\Delta}_{\ell+1} \leftarrow \tilde{\Delta}_\ell / 2,\ \mathbf{C}_{\ell+1}^{\mathrm{U}} \leftarrow \mathbf{C}_\ell^{\mathrm{U}},\ \mathbf{C}_{\ell+1}^{\mathrm{V}} \leftarrow \mathbf{C}_\ell^{\mathrm{V}}$

In the elimination step, the confidence intervals of all re-

maining rows, $[\mathbf{L}^{\text{U}}_\ell(i), \mathbf{U}^{\text{U}}_\ell(i)]$ for any $i \in \mathbf{I}_\ell$, are estimated from matrix $\mathbf{C}^{\text{U}}_\ell \in \mathbb{R}^{K \times L}$; and the confidence intervals of all remaining columns, $[\mathbf{L}^{\text{V}}_\ell(j), \mathbf{U}^{\text{V}}_\ell(j)]$ for any $j \in \mathbf{J}_\ell$, are estimated from $\mathbf{C}^{\text{V}}_\ell \in \mathbb{R}^{K \times L}$. This separation is needed to guarantee that the expected rewards of all remaining rows and columns are scaled similarly. The confidence intervals are designed such that

$$\mathbf{U}^{\text{U}}_\ell(i) \leq \mathbf{L}^{\text{U}}_\ell(\mathbf{i}_\ell) = \max_{i \in \mathbf{I}_\ell} \mathbf{L}^{\text{U}}_\ell(i)$$

implies that row i is suboptimal with a high probability for any column elimination policy up to the end of stage ℓ, and

$$\mathbf{U}^{\text{V}}_\ell(j) \leq \mathbf{L}^{\text{V}}_\ell(\mathbf{j}_\ell) = \max_{j \in \mathbf{J}_\ell} \mathbf{L}^{\text{V}}_\ell(j)$$

implies that column j is suboptimal with a high probability for any row elimination policy up to the end of stage ℓ. As a result, all suboptimal rows and columns are eliminated correctly with a high probability.

5 Analysis

This section has three subsections. In Section 5.1, we derive a gap-dependent upper bound on the n-step regret of `Rank1Elim`. In Section 5.2, we derive a gap-dependent lower bound that nearly matches our upper bound. In Section 5.3, we discuss the results of our analysis.

5.1 Upper Bound

The hardness of our learning problem is measured by two sets of metrics. The first metrics are gaps. The *gaps* of row $i \in [K]$ and column $j \in [L]$ are defined as

$$\Delta^{\text{U}}_i = \bar{u}(i^*) - \bar{u}(i)\,, \quad \Delta^{\text{V}}_j = \bar{v}(j^*) - \bar{v}(j)\,, \tag{2}$$

respectively; and the *minimum row and column gaps* are defined as

$$\Delta^{\text{U}}_{\min} = \min_{i \in [K]: \Delta^{\text{U}}_i > 0} \Delta^{\text{U}}_i\,, \quad \Delta^{\text{V}}_{\min} = \min_{j \in [L]: \Delta^{\text{V}}_j > 0} \Delta^{\text{V}}_j\,, \tag{3}$$

respectively. Roughly speaking, the smaller the gaps, the harder the problem. The second metric is the minimum of the average of entries in $\bar{u}$ and $\bar{v}$, which is defined as

$$\mu = \min \left\{ \frac{1}{K} \sum_{i=1}^{K} \bar{u}(i),\ \frac{1}{L} \sum_{j=1}^{L} \bar{v}(j) \right\}. \tag{4}$$

The smaller the value of μ, the harder the problem. This quantity appears in our regret bound due to the averaging character of `Rank1Elim` (Section 4). Our upper bound on the regret of `Rank1Elim` is stated and proved below.

Theorem 1. *The expected n-step regret of* `Rank1Elim` *is bounded as*

$$R(n) \leq \frac{1}{\mu^2} \left(\sum_{i=1}^{K} \frac{384}{\bar{\Delta}^{\text{U}}_i} + \sum_{j=1}^{L} \frac{384}{\bar{\Delta}^{\text{V}}_j} \right) \log n + 3(K+L)\,,$$

where

$$\bar{\Delta}^{\text{U}}_i = \Delta^{\text{U}}_i + \mathbb{1}\{\Delta^{\text{U}}_i = 0\}\, \Delta^{\text{V}}_{\min}\,,$$
$$\bar{\Delta}^{\text{V}}_j = \Delta^{\text{V}}_j + \mathbb{1}\{\Delta^{\text{V}}_j = 0\}\, \Delta^{\text{U}}_{\min}\,.$$

The proof of Theorem 1 is organized as follows. First, we bound the probability that at least one confidence interval is violated. The corresponding regret is small, $O(K+L)$. Second, by the design of `Rank1Elim` and because all confidence intervals hold, the expected reward of any row $i \in [K]$ is at least $\mu\bar{u}(i)$. Because all rows are explored in the same way, any suboptimal row i is guaranteed to be eliminated after $O([1/(\mu\Delta^{\text{U}}_i)^2] \log n)$ observations. Third, we factorize the regret due to exploring row i into its row and column components, and bound both of them. This is possible because `Rank1Elim` eliminates rows and columns simultaneously. Finally, we sum up the regret of all explored rows and columns.

Note that the gaps in Theorem 1, $\bar{\Delta}^{\text{U}}_i$ and $\bar{\Delta}^{\text{V}}_j$, are slightly different from those in (2). In particular, all zero row and column gaps in (2) are substituted with the minimum column and row gaps, respectively. The reason is that the regret due to exploring optimal rows and columns is positive until all suboptimal columns and rows are eliminated, respectively. The proof of Theorem 1 is below.

Proof. Let $\mathbf{R}^{\text{U}}_\ell(i)$ and $\mathbf{R}^{\text{V}}_\ell(j)$ be the stochastic regret associated with exploring row i and column j, respectively, in stage ℓ. Then the expected n-step regret of `Rank1Elim` is bounded as

$$R(n) \leq \mathbb{E}\left[\sum_{\ell=0}^{n-1} \left(\sum_{i=1}^{K} \mathbf{R}^{\text{U}}_\ell(i) + \sum_{j=1}^{L} \mathbf{R}^{\text{V}}_\ell(j) \right) \right],$$

where the outer sum is over possibly n stages. Let

$$\begin{aligned} \bar{\mathbf{u}}_\ell(i) &= \sum_{t=0}^{\ell} \mathbb{E}\left[\sum_{j=1}^{L} \frac{\mathbf{C}^{\text{U}}_t(i,j) - \mathbf{C}^{\text{U}}_{t-1}(i,j)}{n_\ell} \,\middle|\, \mathbf{h}^{\text{V}}_t \right] \\ &= \bar{u}(i) \sum_{t=0}^{\ell} \frac{n_t - n_{t-1}}{n_\ell} \sum_{j=1}^{L} \frac{\bar{v}(\mathbf{h}^{\text{V}}_t(j))}{L} \end{aligned}$$

be the expected reward of row $i \in \mathbf{I}_\ell$ in the first ℓ stages, where $n_{-1} = 0$ and $\mathbf{C}^{\text{U}}_{-1}(i,j) = 0$; and let

$$\mathcal{E}^{\text{U}}_\ell = \{\forall i \in \mathbf{I}_\ell : \bar{\mathbf{u}}_\ell(i) \in [\mathbf{L}^{\text{U}}_\ell(i), \mathbf{U}^{\text{U}}_\ell(i)],\ \bar{\mathbf{u}}_\ell(i) \geq \mu\bar{u}(i)\}$$

be the event that for all remaining rows $i \in \mathbf{I}_\ell$ at the end of stage ℓ, the confidence interval on the expected reward holds and that this reward is at least $\mu\bar{u}(i)$. Let $\overline{\mathcal{E}^{\text{U}}_\ell}$ be the complement of event $\mathcal{E}^{\text{U}}_\ell$. Let

$$\begin{aligned} \bar{\mathbf{v}}_\ell(j) &= \sum_{t=0}^{\ell} \mathbb{E}\left[\sum_{i=1}^{K} \frac{\mathbf{C}^{\text{V}}_t(i,j) - \mathbf{C}^{\text{V}}_{t-1}(i,j)}{n_\ell} \,\middle|\, \mathbf{h}^{\text{U}}_t \right] \\ &= \bar{v}(j) \sum_{t=0}^{\ell} \frac{n_t - n_{t-1}}{n_\ell} \sum_{i=1}^{K} \frac{\bar{u}(\mathbf{h}^{\text{U}}_t(i))}{K} \end{aligned}$$

denote the expected reward of column $j \in \mathbf{J}_\ell$ in the first ℓ stages, where $n_{-1} = 0$ and $\mathbf{C}^{\mathrm{V}}_{-1}(i,j) = 0$; and let

$$\mathcal{E}^{\mathrm{V}}_\ell = \{\forall j \in \mathbf{J}_\ell : \bar{\mathbf{v}}_\ell(j) \in [\mathbf{L}^{\mathrm{V}}_\ell(j), \mathbf{U}^{\mathrm{V}}_\ell(j)],\ \bar{\mathbf{v}}_\ell(j) \geq \mu\bar{v}(j)\}$$

be the event that for all remaining columns $j \in \mathbf{J}_\ell$ at the end of stage ℓ, the confidence interval on the expected reward holds and that this reward is at least $\mu\bar{v}(j)$. Let $\overline{\mathcal{E}^{\mathrm{V}}_\ell}$ be the complement of event $\mathcal{E}^{\mathrm{V}}_\ell$. Let $\mathcal{E}$ be the event that all events $\mathcal{E}^{\mathrm{U}}_\ell$ and $\mathcal{E}^{\mathrm{V}}_\ell$ happen; and $\overline{\mathcal{E}}$ be the complement of $\mathcal{E}$, the event that at least one of $\mathcal{E}^{\mathrm{U}}_\ell$ and $\mathcal{E}^{\mathrm{V}}_\ell$ does not happen. Then the expected n-step regret of `Rank1Elim` is bounded from above as

$$\begin{aligned}
R(n) \leq\ & \mathbb{E}\left[\left(\sum_{\ell=0}^{n-1}\left(\sum_{i=1}^{K}\mathbf{R}^{\mathrm{U}}_\ell(i) + \sum_{j=1}^{L}\mathbf{R}^{\mathrm{V}}_\ell(j)\right)\right)\mathbb{1}\{\mathcal{E}\}\right] + \\
& nP(\overline{\mathcal{E}}) \\
\leq\ & \sum_{i=1}^{K}\mathbb{E}\left[\sum_{\ell=0}^{n-1}\mathbf{R}^{\mathrm{U}}_\ell(i)\mathbb{1}\{\mathcal{E}\}\right] + \\
& \sum_{j=1}^{L}\mathbb{E}\left[\sum_{\ell=0}^{n-1}\mathbf{R}^{\mathrm{V}}_\ell(j)\mathbb{1}\{\mathcal{E}\}\right] + 2(K+L)\,,
\end{aligned}$$

where the last inequality is from Lemma 1 in Appendix A.

Let $\mathcal{H}_\ell = (\mathbf{I}_\ell, \mathbf{J}_\ell)$ be the rows and columns in stage ℓ, and

$$\mathcal{F}_\ell = \left\{\forall i \in \mathbf{I}_\ell, j \in \mathbf{J}_\ell : \Delta^{\mathrm{U}}_i \leq \frac{2\tilde{\Delta}_{\ell-1}}{\mu},\ \Delta^{\mathrm{V}}_j \leq \frac{2\tilde{\Delta}_{\ell-1}}{\mu}\right\}$$

be the event that all rows and columns with "large gaps" are eliminated by the beginning of stage ℓ. By Lemma 2 in Appendix A, event $\mathcal{E}$ causes event $\mathcal{F}_\ell$. Now note that the expected regret in stage ℓ is independent of $\mathcal{F}_\ell$ given $\mathcal{H}_\ell$. Therefore, the regret can be further bounded as

$$\begin{aligned}
R(n) \leq\ & \sum_{i=1}^{K}\mathbb{E}\left[\sum_{\ell=0}^{n-1}\mathbb{E}\left[\mathbf{R}^{\mathrm{U}}_\ell(i)\,|\,\mathcal{H}_\ell\right]\mathbb{1}\{\mathcal{F}_\ell\}\right] + \\
& \sum_{j=1}^{L}\mathbb{E}\left[\sum_{\ell=0}^{n-1}\mathbb{E}\left[\mathbf{R}^{\mathrm{V}}_\ell(j)\,|\,\mathcal{H}_\ell\right]\mathbb{1}\{\mathcal{F}_\ell\}\right] + \\
& 2(K+L)\,.
\end{aligned} \tag{5}$$

By Lemma 3 in Appendix A,

$$\begin{aligned}
\mathbb{E}\left[\sum_{\ell=0}^{n-1}\mathbb{E}\left[\mathbf{R}^{\mathrm{U}}_\ell(i)\,|\,\mathcal{H}_\ell\right]\mathbb{1}\{\mathcal{F}_\ell\}\right] &\leq \frac{384}{\mu^2\bar{\Delta}^{\mathrm{U}}_i}\log n + 1\,, \\
\mathbb{E}\left[\sum_{\ell=0}^{n-1}\mathbb{E}\left[\mathbf{R}^{\mathrm{V}}_\ell(j)\,|\,\mathcal{H}_\ell\right]\mathbb{1}\{\mathcal{F}_\ell\}\right] &\leq \frac{384}{\mu^2\bar{\Delta}^{\mathrm{V}}_j}\log n + 1\,,
\end{aligned}$$

for any row $i \in [K]$ and column $j \in [L]$. Finally, we apply the above upper bounds to (5) and get our main claim. ■

5.2 Lower Bound

We derive a gap-dependent lower bound on the family of rank-1 bandits where P_{U} and P_{V} are products of independent Bernoulli variables, which are parameterized by their means $\bar{u}$ and $\bar{v}$, respectively. The lower bound is derived for any *uniformly efficient algorithm* $\mathcal{A}$, which is any algorithm such that for any $(\bar{u}, \bar{v}) \in [0,1]^K \times [0,1]^L$ and any $\alpha \in (0,1)$, $R(n) = o(n^\alpha)$.

Theorem 2. *For any problem $(\bar{u}, \bar{v}) \in [0,1]^K \times [0,1]^L$ with a unique best arm and any uniformly efficient algorithm $\mathcal{A}$ whose regret is $R(n)$,*

$$\begin{aligned}
\liminf_{n\to\infty}\frac{R(n)}{\log n} \geq & \sum_{i\in[K]\setminus\{i^*\}}\frac{\bar{u}(i^*)\bar{v}(j^*) - \bar{u}(i)\bar{v}(j^*)}{d(\bar{u}(i)\bar{v}(j^*), \bar{u}(i^*)\bar{v}(j^*))} + \\
& \sum_{j\in[L]\setminus\{j^*\}}\frac{\bar{u}(i^*)\bar{v}(j^*) - \bar{u}(i^*)\bar{v}(j)}{d(\bar{u}(i^*)\bar{v}(j), \bar{u}(i^*)\bar{v}(j^*))}\,,
\end{aligned}$$

where $d(p,q)$ is the Kullback-Leibler (KL) divergence *between Bernoulli random variables with means p and q.*

The lower bound involves two terms. The first term is the regret due to learning the optimal row i^*, while playing the optimal column j^*. The second term is the regret due to learning the optimal column j^*, while playing the optimal row i^*. We do not know whether this lower bound is tight. We discuss its tightness in Section 5.3.

Proof. The proof is based on the change-of-measure techniques from Kaufmann *et al.* [13] and Lagree *et al.* [21], who ultimately build on Graves and Lai [11]. Let

$$w^*(\bar{u}, \bar{v}) = \max_{(i,j)\in[K]\times[L]}\bar{u}(i)\bar{v}(j)$$

be the maximum reward in model $(\bar{u}, \bar{v})$. We consider the set of models where $\bar{u}(i^*)$ and $\bar{v}(j^*)$ remain the same, but the optimal arm changes,

$$\begin{aligned}
B(\bar{u}, \bar{v}) = \{(\bar{u}', \bar{v}') \in [0,1]^K \times [0,1]^L : & \bar{u}(i^*) = \bar{u}'(i^*), \\
& \bar{v}(j^*) = \bar{v}'(j^*),\ w^*(\bar{u}, \bar{v}) < w^*(\bar{u}', \bar{v}')\}\,.
\end{aligned}$$

By Theorem 17 of Kaufmann *et al.* [13],

$$\liminf_{n\to\infty}\frac{\sum_{i=1}^{K}\sum_{j=1}^{L}\mathbb{E}\left[\mathbf{T}_n(i,j)\right]d(\bar{u}(i)\bar{v}(j), \bar{u}'(i)\bar{v}'(j))}{\log n} \geq 1$$

for any $(\bar{u}', \bar{v}') \in B(\bar{u}, \bar{v})$, where $\mathbb{E}\left[\mathbf{T}_n(i,j)\right]$ is the expected number of times that arm (i,j) is chosen in n steps in problem $(\bar{u}, \bar{v})$. From this and the regret decomposition

$$R(n) = \textstyle\sum_{i=1}^{K}\sum_{j=1}^{L}\mathbb{E}\left[\mathbf{T}_n(i,j)\right](\bar{u}(i^*)\bar{v}(j^*) - \bar{u}(i)\bar{v}(j))\,,$$

we get that

$$\liminf_{n\to\infty}\frac{R(n)}{\log n} \geq f(\bar{u}, \bar{v})\,,$$

where

$$f(\bar{u}, \bar{v}) = \inf_{c \in \Theta} \sum_{i=1}^{K} \sum_{j=1}^{L} (\bar{u}(i^*)\bar{v}(j^*) - \bar{u}(i)\bar{v}(j)) c_{i,j}$$
$$\text{s.t.} \quad \forall (\bar{u}', \bar{v}') \in B(\bar{u}, \bar{v}) :$$
$$\sum_{i=1}^{K} \sum_{j=1}^{L} d(\bar{u}(i)\bar{v}(j), \bar{u}'(i)\bar{v}'(j)) c_{i,j} \geq 1$$

and $\Theta = [0, \infty)^{K \times L}$. To obtain our lower bound, we carefully relax the constraints of the above problem, so that we do not loose much in the bound. The details are presented in Appendix B. In the relaxed problem, only $K + L - 1$ entries in the optimal solution c^* are non-zero, as in Combes *et al.* [6], and they are

$$c^*_{i,j} = \begin{cases} 1/d(\bar{u}(i)\bar{v}(j^*), \bar{u}(i^*)\bar{v}(j^*)) \,, & j = j^*, i \neq i^* \,; \\ 1/d(\bar{u}(i^*)\bar{v}(j), \bar{u}(i^*)\bar{v}(j^*)) \,, & i = i^*, j \neq j^* \,; \\ 0 \,, & \text{otherwise.} \end{cases}$$

Now we substitute c^* into the objective of the above problem and get our lower bound. ■

5.3 Discussion

We derive a gap-dependent upper bound on the n-step regret of `Rank1Elim` in Theorem 1, which is

$$O((K + L)(1/\mu^2)(1/\Delta) \log n) \,,$$

where K denotes the number of rows, L denotes the number of columns, $\Delta = \min \{\Delta^{\text{U}}_{\text{min}}, \Delta^{\text{V}}_{\text{min}}\}$ is the minimum of the row and column gaps in (3), and μ is the minimum of the average of entries in $\bar{u}$ and $\bar{v}$, as defined in (4).

We argue that our upper bound is nearly tight on the following class of problems. The i-th entry of $\mathbf{u}_t$, $\mathbf{u}_t(i)$, is an independent Bernoulli variable with mean

$$\bar{u}(i) = p_{\text{U}} + \Delta_{\text{U}} \mathbb{1}\{i = 1\}$$

for some $p_{\text{U}} \in [0, 1]$ and row gap $\Delta_{\text{U}} \in (0, 1 - p_{\text{U}}]$. The j-th entry of $\mathbf{v}_t$, $\mathbf{v}_t(j)$, is an independent Bernoulli variable with mean

$$\bar{v}(j) = p_{\text{V}} + \Delta_{\text{V}} \mathbb{1}\{j = 1\}$$

for $p_{\text{V}} \in [0, 1]$ and column gap $\Delta_{\text{V}} \in (0, 1 - p_{\text{V}}]$. Note that the optimal arm is $(1, 1)$ and that the expected reward for choosing it is $(p_{\text{U}} + \Delta_{\text{U}})(p_{\text{V}} + \Delta_{\text{V}})$. We refer to the instance of this problem by $B_{\text{SPIKE}}(K, L, p_{\text{U}}, p_{\text{V}}, \Delta_{\text{U}}, \Delta_{\text{V}})$; and parameterize it by K, L, p_{U}, p_{V}, Δ_{U}, and Δ_{V}.

Let $p_{\text{U}} = 0.5 - \Delta_{\text{U}}$ for $\Delta_{\text{U}} \in [0, 0.25]$, and $p_{\text{V}} = 0.5 - \Delta_{\text{V}}$ for $\Delta_{\text{V}} \in [0, 0.25]$. Then the upper bound in Theorem 1 is

$$O([K(1/\Delta_{\text{U}}) + L(1/\Delta_{\text{V}})] \log n)$$

since $1/\mu^2 \leq 1/0.25^2 = 16$. On the other hand, the lower bound in Theorem 2 is

$$\Omega([K(1/\Delta_{\text{U}}) + L(1/\Delta_{\text{V}})] \log n)$$

since $d(p, q) \leq [q(1 - q)]^{-1}(p - q)^2$ and $q = 1 - q = 0.5$. Note that the bounds match in K, L, the gaps, and $\log n$.

We conclude with the observation that `Rank1Elim` is suboptimal in problems where μ in (4) is small. In particular, consider the above problem, and choose $\Delta_{\text{U}} = \Delta_{\text{V}} = 0.5$ and $K = L$. In this problem, the regret of `Rank1Elim` is $O(K^3 \log n)$; because `Rank1Elim` eliminates $O(K)$ rows and columns with $O(1/K)$ gaps, and the regret for choosing any suboptimal arm is $O(1)$. This is much higher than the regret of a naive solution by `UCB1` in Section 3, which would be $O(K^2 \log n)$. Note that the upper bound in Theorem 1 is also $O(K^3 \log n)$. Therefore, it is not loose, and a new algorithm is necessary to improve over `UCB1` in this particular problem.

6 Experiments

We conduct three experiments. In Section 6.1, we validate that the regret of `Rank1Elim` grows as suggested by Theorem 1. In Section 6.2, we compare `Rank1Elim` to three baselines. Finally, in Section 6.3, we evaluate `Rank1Elim` on a real-world problem where our modeling assumptions are violated.

6.1 Regret Bound

The first experiment shows that the regret of `Rank1Elim` scales as suggested by our upper bound in Theorem 1. We experiment with the class of synthetic problems from Section 5.3, $B_{\text{SPIKE}}(K, L, p_{\text{U}}, p_{\text{V}}, \Delta_{\text{U}}, \Delta_{\text{V}})$. We vary its parameters and report the n-step regret in 2 million (M) steps.

Table 1 shows the n-step regret of `Rank1Elim` for various choices of K, L, p_{U}, p_{V}, Δ_{U}, and Δ_{V}. In each table, we vary two parameters and keep the rest fixed. We observe that the regret increases as K and L increase, and Δ_{U} and Δ_{V} decrease; as suggested by Theorem 1. Specifically, the regret doubles when K and L are doubled, and when Δ_{U} and Δ_{V} are halved. We also observe that the regret is not quadratic in $1/\mu$, where $\mu \approx \min \{p_{\text{U}}, p_{\text{V}}\}$. This indicates that the upper bound in Theorem 1 is loose in μ when μ is bounded away from zero. We argue in Section 5.3 that this is not the case as $\mu \to 0$.

6.2 Comparison to Alternative Solutions

In the second experiment, we compare `Rank1Elim` to the three alternative methods in Section 3: `UCB1`, `LinUCB`, and `GLM-UCB`. The confidence radii of `LinUCB` and `GLM-UCB` are set as suggested by Abbasi-Yadkori *et al.* [1] and Filippi *et al.* [9], respectively. The maximum-likelihood estimates of

K	L	Regret
8	8	17491 ± 384
8	16	29628 ± 1499
8	32	50030 ± 1931
16	8	28862 ± 585
16	16	41823 ± 1689
16	32	62451 ± 2268
32	8	46156 ± 806
32	16	61992 ± 2339
32	32	85208 ± 3546

$p_{\mathrm{U}} = p_{\mathrm{V}} = 0.7,\ \Delta_{\mathrm{U}} = \Delta_{\mathrm{V}} = 0.2$

p_{U}	p_{V}	Regret
0.700	0.700	17744 ± 466
0.700	0.350	23983 ± 594
0.700	0.175	24776 ± 2333
0.350	0.700	22963 ± 205
0.350	0.350	38373 ± 71
0.350	0.175	57401 ± 68
0.175	0.700	27440 ± 2011
0.175	0.350	57492 ± 67
0.175	0.175	95586 ± 99

$K = L = 8,\ \Delta_{\mathrm{U}} = \Delta_{\mathrm{V}} = 0.2$

Δ_{U}	Δ_{V}	Regret
0.20	0.20	17653 ± 307
0.20	0.10	22891 ± 912
0.20	0.05	30954 ± 787
0.10	0.20	20958 ± 614
0.10	0.10	33642 ± 1089
0.10	0.05	45511 ± 3257
0.05	0.20	30688 ± 482
0.05	0.10	44390 ± 2542
0.05	0.05	68412 ± 2312

$K = L = 8,\ p_{\mathrm{U}} = p_{\mathrm{V}} = 0.7$

Table 1: The n-step regret of `Rank1Elim` in $n = 2$M steps as K and L increase (left), p_{U} and p_{V} decrease (middle), and Δ_{U} and Δ_{V} decrease (right). The results are averaged over 20 runs.

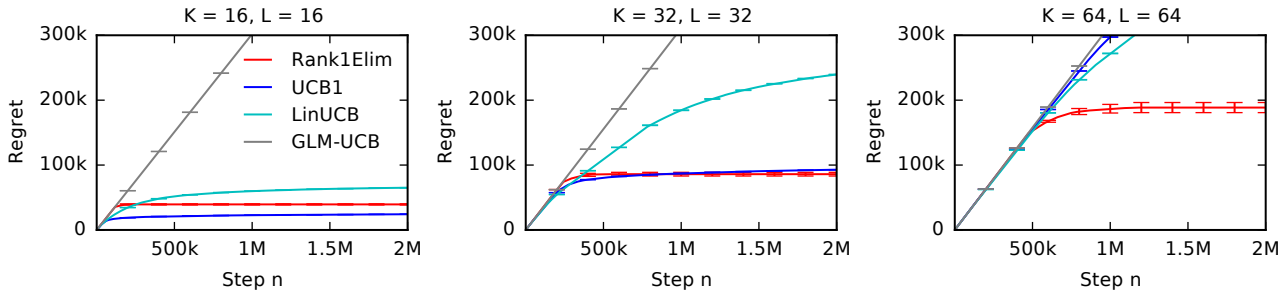


Figure 1: The n-step regret of `Rank1Elim`, `UCB1`, `LinUCB`, and `GLM-UCB` on three synthetic problems in up to $n = 2$M steps. The results are averaged over 20 runs.

$\bar{u}$ and $\bar{v}$ in `GLM-UCB` are computed using the online EM [4], which is observed to converge to $\bar{u}$ and $\bar{v}$ in our problems. We experiment with the problem from Section 6.1, where $p_{\mathrm{U}} = p_{\mathrm{V}} = 0.7$, $\Delta_{\mathrm{U}} = \Delta_{\mathrm{V}} = 0.2$, and $K = L$.

Our results are reported in Figure 1. We observe that the regret of `Rank1Elim` flattens in all three problems, which indicates that `Rank1Elim` learns the optimal arm. When $K = 16$, `UCB1` has a lower regret than `Rank1Elim`. However, because the regret of `UCB1` is $O(KL)$ and the regret of `Rank1Elim` is $O(K + L)$, `Rank1Elim` can outperform `UCB1` on larger problems. When $K = 32$, both algorithms already perform similarly; and when $K = 64$, `Rank1Elim` clearly outperforms `UCB1`. This shows that `Rank1Elim` can leverage the structure of our problem. Neither `LinUCB` nor `GLM-UCB` are competitive on any of our problems.

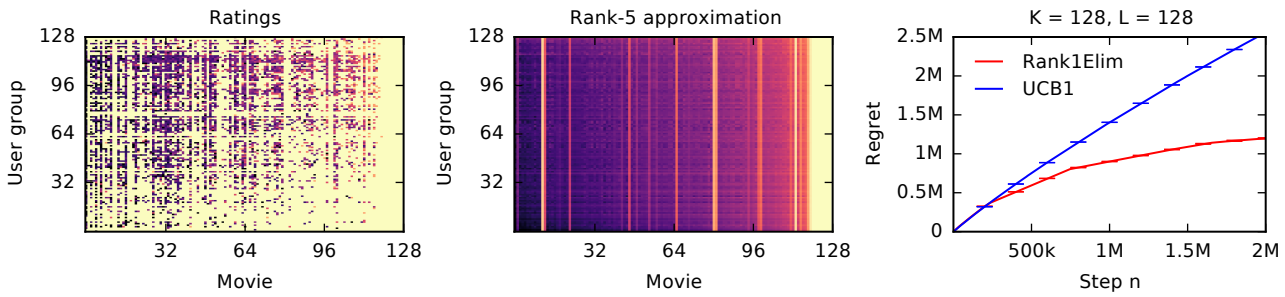
We investigated the poor performance of both `LinUCB` and `GLM-UCB`. When the confidence radii of `LinUCB` are multiplied by $1/3$, `LinUCB` becomes competitive on all problems. When the confidence radii of `GLM-UCB` are multiplied by $1/100$, `GLM-UCB` is still not competitive on any of our problems. We conclude that `LinUCB` and `GLM-UCB` perform poorly because their theory-suggested confidence intervals are too wide. In contrast, `Rank1Elim` is implemented with its theory-suggested intervals in all experiments.

6.3 MovieLens Experiment

In our last experiment, we evaluate `Rank1Elim` on a recommendation problem. The goal is to identify the pair of a user group and movie that has the highest expected rating. We experiment with the *MovieLens* dataset from February 2003 [22], where 6k users give 1M ratings to 4k movies.

Our learning problem is formulated as follows. We define a user group for every unique combination of gender, age group, and occupation in the MovieLens dataset. The total number of groups is 241. For each user group and movie, we average the ratings of all users in that group that rated that movie, and learn a low-rank approximation to the underlying rating matrix by a state-of-the-art algorithm [15]. The algorithm automatically detects the rank of the matrix to be 5. We randomly choose $K = 128$ user groups and $L = 128$ movies. We report the average ratings of these user groups and movies in Figure 2a, and the corresponding completed rating matrix in Figure 2b. The reward for choosing user group $i \in [K]$ and movie $j \in [L]$ is a categorical random variable over five-star ratings. We estimate its parameters based on the assumption that the ratings are normally distributed with a fixed variance, conditioned on the completed ratings. The expected rewards in this experiment are not rank 1. Therefore, our model is misspecified and `Rank1Elim` has no guarantees on its performance.

Our results are reported in Figure 2c. We observe that the



(a) (b) (c)

Figure 2: **a**. Ratings from the MovieLens dataset. The darker the color, the higher the rating. The rows and columns are ordered by their average ratings. The missing ratings are shown in yellow. **b**. Rank-5 approximation to the ratings. **c**. The n-step regret of `Rank1Elim` and `UCB1` in up to $n = 2$M steps.

regret of `Rank1Elim` is concave in the number of steps n, and flattens. This indicates that `Rank1Elim` learns a near-optimal solution. This is possible because of the structure of our rating matrix. Although it is rank 5, its first eigenvalue is an order of magnitude larger than the remaining four non-zero eigenvalues. This structure is not surprising because the ratings of items are often subject to significant *user and item biases* [17]. Therefore, our rating matrix is nearly rank 1, and `Rank1Elim` learns a good solution. Our theory cannot explain this result and we leave it for future work. Finally, we note that `UCB1` explores throughout because our problem has more than 10k arms.

7 Related Work

Zhao *et al.* [29] proposed a bandit algorithm for low-rank matrix completion, where the posterior of latent item factors is approximated by its point estimate. This algorithm is not analyzed. Kawale *et al.* [14] proposed a Thompson sampling (TS) algorithm for low-rank matrix completion, where the posterior of low-rank matrices is approximated by particle filtering. A computationally-inefficient variant of the algorithm has $O((1/\Delta^2) \log n)$ regret in rank-1 matrices. In contrast, note that `Rank1Elim` is computationally efficient and its n-step regret is $O((1/\Delta) \log n)$.

The problem of learning to recommended in the bandit setting was studied in several recent papers. Valko *et al.* [28] and Kocak *et al.* [16] proposed content-based recommendation algorithms, where the features of items are derived from a known similarity graph over the items. Gentile *et al.* [10] proposed an algorithm that clusters users based on their preferences, under the assumption that the features of items are known. Li *et al.* [23] extended this algorithm to the clustering of items. Maillard *et al.* [25] studied a multi-armed bandit problem where the arms are partitioned into latent groups. The problems in the last three papers are a special form of low-rank matrix completion, where some rows are identical. In this work, we do not make any such assumptions, but our results are limited to rank 1.

`Rank1Elim` is motivated by the structure of the position-based model [7]. Lagree *et al.* [21] proposed a bandit algorithm for this model under the assumption that the examination probabilities of all positions are known. Online learning to rank in click models was studied in several recent papers [18, 6, 19, 12, 24, 30]. In practice, the probability of clicking on an item depends on both the item and its position, and this work is a major step towards learning to rank from such heterogeneous effects.

8 Conclusions

In this work, we propose stochastic rank-1 bandits, a class of online learning problems where the goal is to learn the maximum entry of a rank-1 matrix. This problem is challenging because the reward is a product of latent random variables, which are not observed. We propose a practical algorithm for solving this problem, `Rank1Elim`, and prove a gap-dependent upper bound on its regret. We also prove a nearly matching gap-dependent lower bound. Finally, we evaluate `Rank1Elim` empirically. In particular, we validate the scaling of its regret, compare it to baselines, and show that it learns high-quality solutions even when our modeling assumptions are mildly violated.

We conclude that `Rank1Elim` is a practical algorithm for finding the maximum entry of a stochastic rank-1 matrix. It is surprisingly competitive with various baselines (Section 6.2) and can be applied to higher-rank matrices (Section 6.3). On the other hand, we show that `Rank1Elim` can be suboptimal on relatively simple problems (Section 5.3). We plan to address this issue in our future work. We note that our results can be generalized to other reward models, such as $\mathbf{u}_t(i)\mathbf{v}_t(j) \sim \mathcal{N}(\bar{u}(i)\bar{v}(j), \sigma)$ for $\sigma > 0$.

Acknowledgments

This work was partially supported by NSERC and by the Alberta Innovates Technology Futures through the Alberta Machine Intelligence Institute (AMII).

References

[1] Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24*, pages 2312–2320, 2011.

[2] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002.

[3] Peter Auer and Ronald Ortner. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

[4] Olivier Cappe and Eric Moulines. Online EM algorithm for latent data models. *Journal of the Royal Statistical Society Series B*, 71(3):593–613, 2009.

[5] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. *Click Models for Web Search*. Morgan & Claypool Publishers, 2015.

[6] Richard Combes, Stefan Magureanu, Alexandre Proutiere, and Cyrille Laroche. Learning to rank: Regret lower bounds and efficient algorithms. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, 2015.

[7] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 1st ACM International Conference on Web Search and Data Mining*, pages 87–94, 2008.

[8] Varsha Dani, Thomas Hayes, and Sham Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 355–366, 2008.

[9] Sarah Filippi, Olivier Cappe, Aurelien Garivier, and Csaba Szepesvari. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems 23*, pages 586–594, 2010.

[10] Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *Proceedings of the 31st International Conference on Machine Learning*, pages 757–765, 2014.

[11] Todd Graves and Tze Leung Lai. Asymptotically efficient adaptive choice of control laws in controlled Markov chains. *SIAM Journal on Control and Optimization*, 35(3):715–743, 1997.

[12] Sumeet Katariya, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. DCM bandits: Learning to rank with multiple clicks. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1215–1224, 2016.

[13] Emilie Kaufmann, Olivier Cappe, and Aurelien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17:1–42, 2016.

[14] Jaya Kawale, Hung Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. Efficient Thompson sampling for online matrix-factorization recommendation. In *Advances in Neural Information Processing Systems 28*, pages 1297–1305, 2015.

[15] Raghunandan Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11:2057–2078, 2010.

[16] Tomas Kocak, Michal Valko, Remi Munos, and Shipra Agrawal. Spectral Thompson sampling. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1911–1917, 2014.

[17] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[18] Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

[19] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Combinatorial cascading bandits. In *Advances in Neural Information Processing Systems 28*, pages 1450–1458, 2015.

[20] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 2015.

[21] Paul Lagree, Claire Vernade, and Olivier Cappe. Multiple-play bandits in the position-based model. In *Advances in Neural Information Processing Systems 29*, pages 1597–1605, 2016.

[22] Shyong Lam and Jon Herlocker. MovieLens Dataset. http://grouplens.org/datasets/movielens/, 2016.

[23] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *Proceedings of the 39th Annual International ACM SIGIR Conference*, 2016.

[24] Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. Contextual combinatorial cascading bandits. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1245–1253, 2016.

[25] Odalric-Ambrym Maillard and Shie Mannor. Latent bandits. In *Proceedings of the 31st International Conference on Machine Learning*, pages 136–144, 2014.

[26] Maxim Raginsky and Igal Sason. Concentration of measure inequalities in information theory, communications and coding. *CoRR*, abs/1212.4663, 2012.

[27] Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: Estimating the click-through rate for new ads. In *Proceedings of the 16th International Conference on World Wide Web*, pages 521–530, 2007.

[28] Michal Valko, Remi Munos, Branislav Kveton, and Tomas Kocak. Spectral bandits for smooth graph functions. In *Proceedings of the 31st International Conference on Machine Learning*, pages 46–54, 2014.

[29] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. Interactive collaborative filtering. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pages 1411–1420, 2013.

[30] Shi Zong, Hao Ni, Kenny Sung, Nan Rosemary Ke, Zheng Wen, and Branislav Kveton. Cascading bandits for large-scale recommendation problems. In *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence*, 2016.