
Stochastic Low-Rank Latent Bandits

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 To be written.

2 1 Introduction

3 STORY: We address a recommendation problem in the hard setting where no feature is available to
4 the learner. Blah blah: recommendation and bandits, major problem, blah blah.

5 We rely on the assumption that the underlying click-through rate matrix has a latent sructure that we
6 cannot directly observe but that we propose to leverage nonetheless. We formulate a rank- d bandit
7 problem that generalizes previous works on rank-1 and on latent bandits (quote, quote). We propose
8 a meta algorithm that uses two layers of bandit algorithms in order to learn 1/the best set of items
9 overall and 2/ the individual preferences. This is a novel and efficient bandit startegy for the latent
10 bandits and an elegant generalization of the rank-1 setting. We show a regret bound for our algorithm
11 and run experiments on simulated and real data.

12 XXXXXXXXXXXXXXXXXXXXXXX

13 Cla: I haven't changed this section yet, wanted to make sure the story is right before.

14 In this paper, we study the problem of recommending the best items to users who are coming
15 sequentially. The learner has access to very less prior information about the users and it has to adapt
16 quickly to the user preferences and suggest the best item to each user. Furthermore, we consider the
17 setting where users are grouped into clusters and within each cluster the users have the same choice
18 of the best item, even though their quality of preference may be different for the best item. These
19 clusters along with the choice of the best item for each user are unknown to the learner. Also, we
20 assume that each user has a single best item preference.

21 This complex problem can be conceptualized as a low rank stochastic bandit problem where there
22 are K users and L items. The reward matrix, denoted by $\bar{M} \in [0, 1]^{K \times L}$, generating the rewards
23 for user, item pair has a low rank structure. The online learning game proceeds as follows, at every
24 timestep t , nature reveals one user (or row) from \bar{M} where user is denoted by i_t . The learner selects
25 some items (or columns) from \bar{M} , where an item is denoted by $j_t \in [L]$. Then the learner receives
26 one noisy feedback $r_t(i_t, j_t) \sim \mathcal{D}(\bar{M}(i_t, j_t))$, where \mathcal{D} is a distribution over the entries in \bar{M} and
27 $\mathbb{E}[r_t(i_t, j_t)] = \bar{M}(i_t, j_t)$. Then the goal of the learner is to minimize the cumulative regret by quickly
28 identifying the best item j^* for each $i \in [K]$ where $\bar{M}(i, j^*) = \arg \max_{j \in [L]} \{\bar{M}(i, j)\}$.

29 1.1 Notation and Learning Setting

30 Throughout the paper, we denote $[n] = \{1, 2, \dots, n\}$. An instance of the *Low-Rank Bandit* problem is
31 a matrix $R \in [0, 1]^{K \times L}$ representing the expected click-through rates (CTRs) for each user $k \in [K]$
32 on each item $l \in [L]$. If, $J \subset [L]$ is a subset of columns, we denote $R(:, J) \in [0, 1]^{K \times |J|}$ the
33 corresponding submatrix containing the $|J|$ columns of R .

We assume that there exists a latent structure, i.e that $R = UV^T$ where the rows of U and V contain the hidden users' and item's features. It is important to notice that none of those features are observable, meaning that we cannot build on a linear bandit model, and in particular our problem cannot be seen as a *clustering of bandits* problem ?. However, the rank of the CTR matrix is assumed to be low, that is $d \ll \min\{L, K\}$. This is the key assumption of our model. It implies, by definition, the following property.

Observation 1. Let $M \in \mathbb{R}^{K \times L}$ be a rank- d matrix. Then,

- There exists a basis J^* of d column such that all the L columns' latent features are linear combinations of the vectors in J^* ;
- There exists a basis I^* of d users such that all the K users' latent features are linear combinations of the vectors in I^* .

Without loss of generality, the above mentioned bases can be chosen of maximal volume such that the corresponding transformation matrix is the least singular possible.

Proof. The existence of the basis on both dimensions comes directly by definition of the low rank assumption. The choice of the spanning vectors is arbitrary and maximising the volume means choosing vectors with larger norm and hence potentially larger payoff. ■

Cl: Here state the result on the existence of a best set of d items, I'm not sure how to state it. It is not an "assumption" though, it is a Lemma or a Fact but not an assumption. It is a consequence of the low rank assumption :)

The interaction at round $t \geq 1$ of the learner with the online recommender system characterized by R goes as follows:

- a user $i_t \in [K]$ shows up – it corresponds to the index of a row of the matrix. It can be seen as an unobserved context generated by the environment;
- the learner chooses a set $J_t \subset [L]$ such that $|J_t| = d$ to be sequentially presented to the user;
- the user browses those d options and send an individual feedback for each of them (semi-bandit setting): $\forall j \in J_t$, the learner observes $Y_{t,j} = R(i_t, j) + \eta_{t,j}$ where $(\eta_{t,j})_{t,j \geq 0}$ is a sequence of i.i.d centered random variables.

Cl: fix your noise model here. Bernoulli ??

For each user $i \in [K]$, there exists one unique best item $j^*(i) \in [L]$

Cl: Define the best item, define the expected regret

The objective of the learning agent is to minimize the expected cumulative regret up to horizon n . We define the cumulative regret, denoted by \mathcal{R}_n as,

1.2 Related Works

In ? the authors propose the Latent Bandit model where there are two sets: 1) set of arms denoted by \mathcal{A} and 2) set of types denoted by \mathcal{B} which contains the latent information regarding the arms. The latent information for the arms are modeled such that the set \mathcal{B} is assumed to be partitioned into $|\mathcal{C}|$ clusters, indexed by $\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_C \in \mathcal{C}$ such that the distribution $v_{a,b}$, $a \in \mathcal{A}, b \in \mathcal{B}_c$ across each cluster is same. Note, that the identity of the cluster is unknown to the learner. At every timestep t , nature selects a type $b_t \in \mathcal{B}_c$ and then the learner selects an arm $a_t \in \mathcal{A}$ and observes a reward $r_t(a, b)$ from the distribution $v_{a,b}$.

Another way to look at this problem is to imagine a matrix of dimension $|\mathcal{A}| \times |\mathcal{B}|$ where again the rows in \mathcal{B} can be partitioned into $|\mathcal{C}|$ clusters, such that the distribution across each of this clusters are same. Now, at every timestep t one of this row is revealed to the learner and it chooses one column such that the $v_{a,b}$ is one of the $\{v_{a,c}\}_{c \in \mathcal{C}}$ and the reward for that arm and the user is revealed to the learner.

77 This is actually a much simpler approach than the setting we considered because note that the
 78 distributions across each of the clusters $\{v_{a,c}\}_{c \in \mathcal{C}}$ are identical and estimating one cluster distribution
 79 will reveal all the information of the users in each cluster.

80 2 Algorithm

```

1: for  $t = 1, \dots, n$  do
2:   User  $i_t$  comes to the system
3:
4:   // Choose  $d$  arms in  $d$  column bandits
5:   Let  $p_{c,t}(k, j)$  be the probability of playing arm  $j \in [L]$  in c-bandit  $k \in [d]$  at time  $t$ 
6:   for  $k = 1, \dots, d$  do
7:     Sample  $J_t[k] \sim \text{Cat}(p_{c,t}(k, 1), \dots, p_{c,t}(k, L))$ 
8:
9:   // Choose an arm in row bandit  $i_t$ 
10:  Let  $p_{r,t}(i_t, k)$  be the probability of playing arm  $k \in [d+1]$  in r-bandit  $i_t \in [K]$  at time  $t$ 
11:  Sample  $k_t \sim \text{Cat}(p_{r,t}(i_t, 1), \dots, p_{r,t}(i_t, d+1))$ 
12:
13:  // Update row bandit  $i_t$ 
14:  for  $k = 1, \dots, d$  do
15:    if  $k_t \leq d$  then
16:       $j_{t,k} \leftarrow J_t[k_t]$ 
17:    else
18:       $j_{t,k} \leftarrow J_t[k]$ 
19:   $s_{r,t} \leftarrow s_{r,t-1}$ 
20:   $s_{r,t}(i_t, k_t) \leftarrow s_{r,t}(i_t, k_t) + \sum_{k=1}^d \frac{R_t(i_t, j_{t,k})}{p_{r,t}(i_t, k_t)}$ 
21:
22:  // Update  $d$  column bandits
23:   $s_{c,t} \leftarrow s_{c,t-1}$ 
24:  if  $k_t > d$  then
25:    for  $k = 1, \dots, d$  do
26:       $s_{c,t}(k, J_t[k]) \leftarrow s_{c,t}(k, J_t[k]) + \frac{\max R_t(i_t, J_t[:k]) - \max R_t(i_t, J_t[:k-1])}{p_{c,t}(k, J_t[k]) p_{r,t}(i_t, d+1)}$ 

```

81 3 Analysis

82 We assume that users come sequentially $i_1, \dots, i_n \in [K]$. We denote by $j^*(i)$ the optimal arm of
 83 user i . When $J = (j_1, \dots, j_k) \in [L]^k$ is a k -tuple, by $J[l]$ we will mean j_l , the l 'th entry of J and
 84 $\max R(i, J) := \max_{l \in [k]} R(i, J[l])$.

85 Bra: "I" is a horrible letter because it looks like many other symbols. What about ℓ instead?

86 Let $U_t \in \mathbb{R}_{\geq 0}^{K \times d}$ and $V_t \in \mathbb{R}_{\geq 0}^{L \times d}$ to be time varying latent user and item factors. The reward matrix at
 87 time step $t \in [n]$ is $R_t = U_t V_t^T$.

Assumption 1 (Hott Topics). We will assume that there is a d -tuple $J^* \in [L]^d$ such that for every
 $j \in [L]$, there exists $\alpha_1^j, \dots, \alpha_d^j \geq 0$, $\sum_k \alpha_k^j \leq 1$ and

$$V_t[j, :] = \sum_{k \in J^*} \alpha_k^j V_t[k, :],$$

88 for every $t \in [n]$.

89 An important thing to note is that α_k^j 's are independent of time t . With the above assumption, we
 90 have the following theorem.

Lemma 1. For any set of columns J_t , we have

$$\max_{j \in [L]} \sum_{t \in [n]} \max R_t(i_t, (J_t, j)) = \max_{l \in [d]} \sum_{t \in [n]} \max R_t(i_t, (J_t, J^*[l])).$$

91 **Todo:** What happens with the Bernoulli rounding trick? The above holds when the input is non-
 92 stochastic, but I haven't thought about the stochastic case. Without loss of generality, we will also
 93 assume that for every $1 \leq k \leq d$, $\max_{J \in [L]^k} \sum_t \max R_t(i_t, J) = \sum_t \max R_t(i_t, J^*[1 : k])$

94 Anup: Need to say why this is possible: follows easily from hott topics

. Let $J_t = (\tilde{j}_{t,1}, \tilde{j}_{t,2}, \dots, \tilde{j}_{t,d})$ be the tuple of d columns chosen by column-bandits at time t and $(j_{t,1}, \dots, j_{t,d})$ be the d tuple of columns chosen by i_t th row EXP3 at time t . We want to bound the expected regret $R(n)$

$$R(n) = \mathbb{E} \left(d \sum_t R(i_t, j_t^*(i_t)) - \sum_t \sum_k R(i_t, j_{t,k}) \right).$$

95 The row algorithm plays either one arm in J_t d times or plays every arm one time. We will use an
 96 indicator function $\mathbb{1}(j_{t,1} \neq j_{t,2})$ which takes value one only if the row algorithm

97 Anup: row algorithm? We need a better way to refer to the various EXP3s

98 plays every arm in J_t one time. Let $p_t = P(j_{t,1} \neq j_{t,2})$. We can write the expected regret as
 99 $R(n) = R_c(n) + R_r(n)$, where

$$R_c(n) = \mathbb{E} \left(d \sum_{t=1}^n R(i_t, j_t^*(i_t)) - d \sum_t \frac{\max R(i_t, J_t)}{p_t} \mathbb{1}(j_{t,1} \neq j_{t,2}) \right)$$

100 and

$$R_r(n) = \mathbb{E} \left(d \sum_t \frac{\max R(i_t, J_t)}{p_t} \mathbb{1}(j_{t,1} \neq j_{t,2}) - \sum_t \sum_k R(i_t, j_{t,k}) \right).$$

101 We will show that for every $\gamma > 0$, $R_c(n) = O\left(\frac{d^2}{\gamma} \sqrt{nL \log n}\right)$ and $R_r(n) = O\left(\frac{Kd \log d}{\gamma} + \gamma n\right)$.

Theorem 1. By choosing γ appropriately, for all large enough n , we have

$$R(n) = O\left(dL^{1/4}n^{3/4} \log^{1/4} n\right).$$

102 We now prove the bounds for $R_c(n)$ and $R_r(n)$ separately.

103 **Todo:** Make it clear what the randomness is when using \mathbb{E} throughout.

104 Bra: Given the limited time, let's go with the current setting. This is most natural in the non-stochastic community and nobody will question it. Then the only randomness is with respect to random actions of the algorithm.

105 3.1 Bounding Column Regret

106 To bound $R_c(n)$, we first rewrite it as

$$\begin{aligned} R_c(n) &= d \mathbb{E} \left(\sum_{t=1}^n \frac{R(i_t, j_t^*(i_t))}{p_t} \mathbb{1}(j_{t,1} \neq j_{t,2}) - \sum_t \frac{\max R(i_t, J_t)}{p_t} \mathbb{1}(j_{t,1} \neq j_{t,2}) \right) \\ &= d \mathbb{E} \left(\sum_{t=1}^n \frac{\tilde{R}(i_t, j_t^*(i_t))}{p_t} - \sum_t \frac{\max \tilde{R}(i_t, J_t)}{p_t} \right) \\ &\leq \frac{d}{\min_t p_t} \mathbb{E} \left(\sum_{t=1}^n \max \tilde{R}(i_t, J^*) - \sum_t \max \tilde{R}(i_t, J_t) \right). \end{aligned}$$

107 Here, we define $\tilde{R}_t(i, j) = R_t(i, j) \mathbb{1}(j_{t,1} \neq j_{t,2})$. We are now ready to bound the regret. To avoid
 108 carrying tildes, we denote \tilde{R}_t by R_t in the rest of the proof.

Lemma 2. For any $k \in [d]$,

$$\sum_t \mathbb{E} \max R_t(i_t, J_t[1 : k]) \geq \mathbb{E} \sum_t \max R_t(i_t, J^*[1 : k]) - O(k\sqrt{nL}).$$

109 *Proof.* We will show this by induction. Note that there are d column EXP3s in this case. The base
 110 case when $k = 1$ follows because of the guarantees of the first col-EXP3. Let $J^* = (j_1^*, j_2^*, \dots, j_d^*)$.
 111 We will now assume that the result is true for $k - 1$ for some $k > 1$. We have

$$\mathbb{E} \sum_t \max R_t(i_t, J_t[1 : k]) \tag{1}$$

$$\geq \max_{j_k} \mathbb{E} \sum_t \max R_t(i_t, (J_t[1 : k - 1], j_k)) - O(\sqrt{nL}) \tag{2}$$

$$\geq \max_{j_k} \mathbb{E} \sum_t \max R_t(i_t, (J^*[1 : k - 1], j_k)) - O(\sqrt{nL}) - O((k - 1)\sqrt{nL}) \tag{3}$$

$$= \mathbb{E} \sum_t R_t(i_t, J^*[1 : k]) - O(k\sqrt{nL}). \tag{4}$$

112 The last equality follows from Lemma 4. The first inequality is from the guarantees of k th col-EXP3.
 113

114 Bra: State the regret bound of Exp3 in a lemma.

The crucial step is the second inequality. It says that we can replace $J_t[1 : k - 1]$ with $J^*[1 : k - 1]$ by just losing another additive $O((k - 1)\sqrt{nL})$ term. This follows from induction hypothesis and Lemma 3. We note that from Equation 4, we have

$$\max R_t(i_t, J_t[1 : k]) \geq \mathbb{E} \sum_t \max R_t(i_t, J^*[1 : k]) - O(k\sqrt{nL}),$$

115 which concludes the proof. ■

Lemma 3. Suppose

$$\mathbb{E} \sum_t (\max R_t(i_t, (J_t[1 : k - 1])) \geq \mathbb{E} \sum_t (\max R_t(i_t, (J^*[1 : k - 1])) - C$$

and let $j_k \in [L]$. Then,

$$\mathbb{E} \sum_t (\max R_t(i_t, (J_t[1 : k - 1], j_k)) \geq \mathbb{E} \sum_t (\max R_t(i_t, (J^*[1 : k - 1], j_k)) - O((k - 1)\sqrt{nL})).$$

116 *Proof.* Let $T_1 = \{t \mid \max R_t(i_t, J^*[1 : k - 1]) < R_t(i_t, j_k)\}$ and $T_2 = [n] \setminus T_1$. We then have

$$\begin{aligned} & \mathbb{E} \sum_t \max R_t(i_t, (J_t[1 : k - 1], j_k)) \\ &= \mathbb{E} \sum_{t \in T_1} \max R_t(i_t, (J_t[1 : k - 1], j_k)) + \mathbb{E} \sum_{t \in T_2} \max R_t(i_t, (J_t[1 : k - 1], j_k)) \\ &\geq \sum_{t \in T_1} \max R_t(i_t, (J^*[1 : k - 1], j_k)) + \mathbb{E} \sum_{t \in T_2} \max R_t(i_t, (J_t[1 : k - 1], j_k)) \\ &\geq \sum_{t \in T_1} \max R_t(i_t, (J^*[1 : k - 1], j_k)) + \mathbb{E} \sum_{t \in T_2} \max R_t(i_t, J_t[1 : k - 1]) \\ &\geq \sum_{t \in T_1} \max R_t(i_t, (J^*[1 : k - 1], j_k)) + \sum_{t \in T_2} \max R_t(i_t, J^*[1 : k - 1]) - C \\ &= \sum_{t \in T_1} \max R_t(i_t, (J^*[1 : k - 1], j_k)) + \sum_{t \in T_2} \max R_t(i_t, (J^*[1 : k - 1], j_2)) - C \\ &= \sum_{t \in [n]} \max R_t(i_t, (J^*[1 : k - 1], j_k)) - C. \end{aligned}$$

117 The first inequality is easy because $\max R_t(i_t, (J^*[1 : k - 1], j_k)) = R_t(i_t, j_k)$ for $t \in T_1$. Second
 118 inequality is trivial. Third inequality follows from the assumption. The next equality holds because
 119 of the definition of T_2 . ■

120 Anup: Define a new slicing operator and a more compressed ' \cdot ' operator so that the above expressions look a bit nicer?

121 3.2 Bounding Row Regret

To bound $R_r(n)$, we first note that

$$R_r(n) = E \left(\sum_t d \cdot \max R(i_t, J_t) - \sum_t \sum_k R(i_t, j_{t,k}) \right).$$

We will decompose the regret as a sum of regret of row-EXP3s. There are K row-EXP3s and each one corresponds to a user. Let n_i be the number of times user i appears in the sequence i_1, \dots, i_n . We then have

$$R_{r,i}(n) = \sum_{i \in [K]} R_{r,i}(n)$$

where $R_{r,i}(n) = E (\sum_t d \cdot \max R(i, J_t) - \sum_t \sum_k R(i, j_{t,k}))$. Since each user has a row-EXP3 is over $d + 1$ arms, the regret is bounded by

$$R_{r,i}(n) = (e - 1) \frac{(d + 1) \log(d + 1)}{\gamma} + \gamma n_i,$$

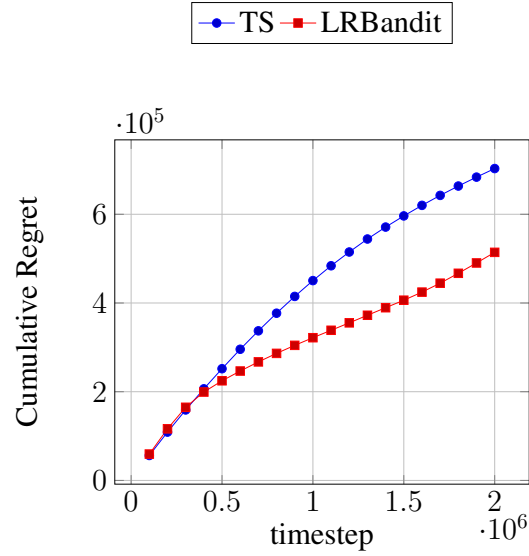
where $\gamma > 0$ is any positive number. Summing this over K users, we get

$$R(n) = (e - 1) \frac{(d + 1) \log(d + 1) K}{\gamma} + \gamma n.$$

122 Anup: This proof needs to have a bit more details. Also, γ should appear in the algorithm and we should refer to that.

123 Bra: Please add more details. This needs to be done over all users.

124 4 Experiments

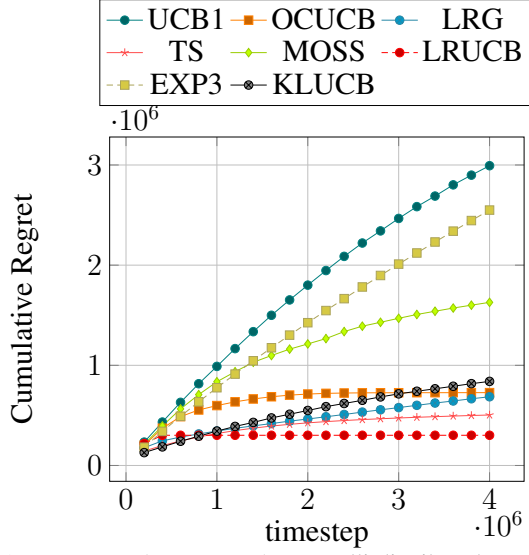


(a) Expt-1: 1024 Users, 128 arms, Round-Robin, Noisy Setting, Rank 2, equal sized clusters

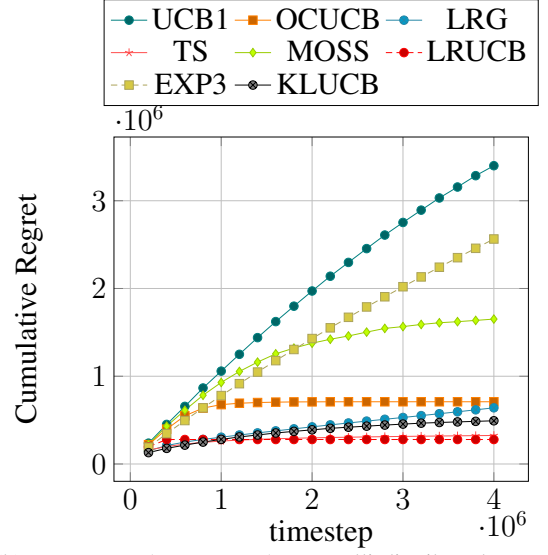
Figure 1: A comparison of the cumulative regret by MRLG and MRLUCB.

125 5 Conclusions and Future Direction

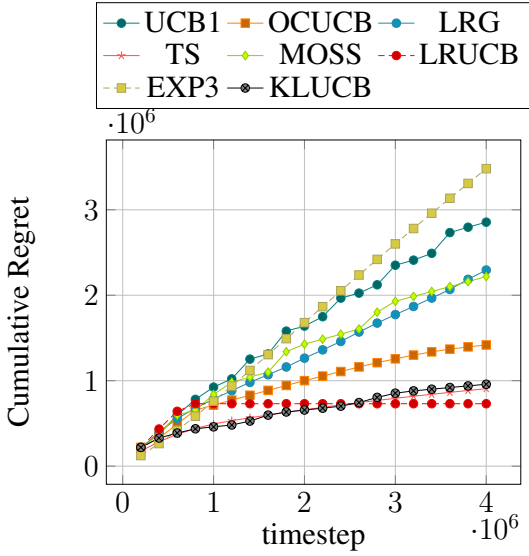
126 To be written.



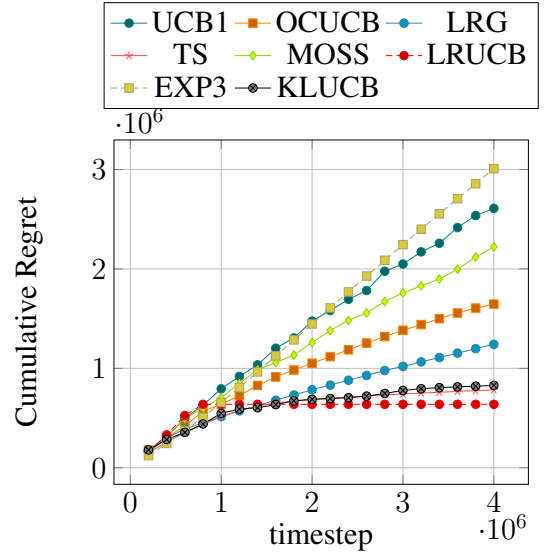
(a) Expt-1: 2048 Users, 64 Bernoulli-distributed arms, Round-Robin, Noisy Setting, Rank 2, equal sized clusters



(b) Expt-2: 2048 Users, 64 Bernoulli-distributed arms, Round-Robin, Noisy Setting, Rank 2, un-equal sized clusters, 70:30 split



(c) Expt-3: 4096 Users, 128 Bernoulli-distributed arms, Round-Robin, Noisy Setting, Rank 2, equal sized clusters



(d) Expt-4: 4096 Users, 128 Bernoulli-distributed arms, Round-Robin, Noisy Setting, Rank 2, un-equal sized clusters, 80:20 split

Figure 2: A comparison of the cumulative regret incurred by the various bandit algorithms.

127 A Claire's point of view

128 We assume that users come sequentially: the indices $i_1, \dots, i_n \in [K]$ are *i.i.d.* random variables
 129 chosen by an unknown (and uncontrolled) distribution \mathcal{D}_u .

130 When $J = (j_1, \dots, j_k) \in [L]^k$ is a k -tuple, by $J[\ell]$ we will mean j_ℓ , the ℓ 'th entry of J and
 131 $\max R(i, J) := \max_{l \in [k]} R(i, J[\ell])$. Let $U_t \in \mathbb{R}_{\geq 0}^{K \times d}$ and $V_t \in \mathbb{R}_{\geq 0}^{L \times d}$ be time varying latent user
 132 and item factors.

133 **Assumption 2** (Hot Topics). We will assume that there is a d -tuple $J^* \in [L]^d$ such that any user
 134 latent vector is a convex combination of the vectors in J^* . In other words, for every $j \in [L]$, there

135 exists $\alpha_1^j, \dots, \alpha_d^j \in [0, 1]^d$, $\sum_k \alpha_k^j \leq 1$ and

$$V[j, :] = \sum_{k \in J^*} \alpha_k^j V[k, :].$$

136 Moreover, for any user $i \in [K]$, we assume that there exists one unique best item that we denote $B(i)$.
 137 The mapping B is deterministic and defined by

$$B(i) = \arg \max_{j \in [L]} u_i^T v_j$$

138 With the above assumption, we have the following cornerstone Lemma.

139 **Lemma 4.** The mapping B has its image included in J^* . This means that for any user $i \in [K]$,
 140 $B(i) \in J^*$.

141 *Proof.* By definition,

$$B(i) = \arg \max_{j \in [L]} u_i^T v_j.$$

142 The function $v \mapsto u_i^T v$ is linear so its maximum on a convex is reached at one of its summits.

143 Cla: could be nice to write this down more properly.

144 ■

145 This means that no matter which user i_t shows up at round t , the best item recommendation is one of
 146 the d elements of J^* .

147 At each round, the learner must choose d arms possibly with repetitions. Given a user i_t , the
 148 optimal action is $A^*(i_t) = (B(i_t), \dots, B(i_t))$. The instantaneous regret incurred by taking action
 149 $A_t = (j_{t,1}, \dots, j_{t,k})$ is

$$r_t = dR(i_t, B(i_t)) - \sum_{k \in A_t} R(i_t, k).$$

150 The goal of the learner is to minimize the expected regret

$$\begin{aligned} R(T) &= \mathbb{E}_{D_u} \sum_{t=1}^T r_t \\ &= \mathbb{E}_{D_u} \sum_{t=1}^T dR(i_t, B(i_t)) - \sum_{k \in A_t} R(i_t, k) \\ &= \sum_{t=1}^T \sum_{k \in A_t} \mathbb{E}_{D_u} [R(i_t, B(i_t)) - R(i_t, k)] \\ &= \sum_{t=1}^T \sum_{j \in [L]} \mathbb{1}\{j \in A_t\} \mathbb{E}_{D_u} [R(i_t, B(i_t)) - R(i_t, j)] \\ &= \sum_{j \in [L]} [N_j(T)] \bar{\Delta}_j \end{aligned}$$

151 where

$$N_j(t) := \sum_{t=1}^T \mathbb{1}\{j \in A_t\}; \quad \bar{\Delta}_j := \mathbb{E}_{D_u} [R(i_t, B(i_t)) - R(i_t, j)].$$

152 Cla: This decomposition is correct but it also hides a little bit too much information about the users. I'm not sure it will actually help bounding the regret of our algorithm but I wanted to write it down.

A.1 Lower bound discussion

Now that the problem is defined, we discuss its complexity through a problem-dependent lower bound on the expected regret. It appears that as soon as the probability of each user to show up is positive, each row bandit problem will be allocated a linear number of request. Moreover, under the assumption that the reward matrix R is rank d , the users' latent vectors are d -dimensional and they span the space (otherwise, the matrix rank would be lower). This implies in particular that each summit of the convex set defined by the master columns in J^* is the best arm for a linear number of rounds. Thus, intuitively, a *uniformly efficient* strategy should pull each arm in J^* a linear number of times. However, all the suboptimal arms $j \notin J^*$ are never $B(i_t)$ for any i_t so they should be pulled only for the sake of exploration.

In order to frame this exploration-exploitation problem in the usual finitely-armed stochastic bandit setting, we will rewrite the parameter of each arm $j \in [L]$. We introduce the parameters $p_i = \mathbb{P}(i_t = i)$ for each user i and we write

$$\theta_j = \mathbb{E}_{i \sim D_u}[R(i, j)] = \left(\sum_{i=1}^K p_i u_i \right)^T v_j,$$

which is the expected reward that the learner receives when he chooses action j in his set. On average over the columns, some rewards have a higher expectation than other due to the uneven representation of the users preferring them. We will denote j_d the d -th best arm. We prove the following theorem

Theorem 2. *The distribution of the rewards associated with each arm $j \in [L]$ is a mixture of K distributions depending on the user. We denote \mathcal{P}_j the probability distribution of the rewards when pulling arm j . For any $\theta^* \in [0, 1]$,*

$$\mathcal{K}_{\inf}(j; \theta^*) = \inf_{\mathcal{P}} \{KL(\mathcal{P}_j, \mathcal{P}) | \mathbb{E}_{\mathcal{P}}[\text{reward}] \geq \theta^*\}. \quad (5)$$

172

173 *Clā: notation is needed here. It's a mess for now.*

174 *The expected regret of any uniformly efficient strategy is bounded from below by*

$$\liminf_{T \rightarrow \infty} \frac{R(T)}{\log(T)} \geq \sum_{j \notin J^*} \frac{\bar{\Delta}_j}{\mathcal{K}_{\inf}(j; \theta_{j_d})}.$$

175 Note that this lower bound takes into account the unknown probability distribution of the users both
176 in the numerator (the gaps are defined in expectation wrt this distribution) and in the denominator
177 (the information quantity \mathcal{K}_{\inf} also depends on it).

178 *Proof.* This proof relies on changes of measure (blah blah). Basically, $\mathcal{K}_{\inf}(j; \theta_{j_d})$ is the expected
179 log-likelihood ratio of the observations under two models : the original one and the one where arm j
180 has a modified parameter v_j that gives it a higher expected reward than j_d . This is a quite standard
181 result but the expectation over the users must be handled gently

182 *Clā: I still need to think about it and fix the notations to make it right but I believe the result is true.*

183 ■

A.2 About the algorithm

185 **Simple Multiple-Plays bandits as a baseline.** One first idea is run a MPB algorithm that builds a
186 list of d items that look better. Unfortunately, the expected regret of such strategy is linear ! Indeed,
187 the optimal action of this kind of method is $A_{\text{MP}}^* = J^*$ that incurs a regret $\sum_{j \in J^*} \bar{\Delta}_j > 0$. This is
188 because the bandit algorithm learns one fixed best action for all arms while the optimal strategy for
189 our problem is to learn the best arm of each user *among the d best arms*.

190 **Hierarchical bandits.** To overcome this additional difficulty, we suggest the following general
191 idea:

- 192 • We maintain a column bandit that takes the averaged rewards over the rows and learns the
193 best d columns in expectation over the users;
- 194 • At each round, a user i_t pops up and the corresponding *independent* row bandit will make
195 his own recommendation decision after calling the column bandit for advice.
- 196 • The column bandit will send a possible set of d *different* arms S_t and the row bandit will
197 choose the final action A_t by constructing a set out of these d suggestions. To simplify
198 exposition, we will assume that two types of set can be constructed: either an *exploratory*
199 one that simply pulls all the suggested arms and a *exploitation* action that decides which is
200 the most promising item among the suggested ones and simply fills the whole list with d
201 identical arms.

202 Cla: Note that it is really important to be consistent when using the terms action for the list and arms for each individual item of the list, otherwise it's a mess. I'm doing my best...

203 In order to decompose the regret, we need to split the rounds when the column bandit recommended
204 the best set J^* .

205 Fix a row $i \in [K]$ and consider the filtration $\mathcal{F}_i = \{t \leq T : i_t = i\}$. The regret of the corresponding
206 bandit is equal to

$$R_i(T) = \mathbb{E} \left[\sum_{t \in \mathcal{F}_i} r_t \mathbb{S} = J^* \right] + \mathbb{E} \left[\sum_{t \in \mathcal{F}_i} \mathbb{S} \neq J^* \right]$$

207 The idea is now to say

- 208 • the first term is bounded by $O(d \log(p_i T))$ because the chosen row bandit algorithm is
209 designed for that,
- 210 • the second term is bounded by $O(L \log(T))$ because the column bandit is designed for that.

211 Even if the intuition seems to go through, it is not completely immediate to prove. The recom-
212 mendations of the column bandits will be based on the actions and observations gathered by the
213 row bandits and simply averaged over the rows. This means for instance that if the column bandit
214 is TS, its posterior for arm j at round T is a gaussian (assuming Gaussian noise...) with mean
215 $\sum_i S_{i,j}(T) / \sum_i N_{i,j}(T)$. In order to make sure that the column bandit does learn the best action J^* ,
216 we must make sure that the arms in J^* are pulled enough such that the expectation of the optimal
217 action is not badly underestimated. Given that the column bandit cannot directly control the actions,
218 it seems hard.