Stochastic Low-Rank Latent Bandits

Anonymous Author(s)
Affiliation
Address
email

Abstract

To be written.

1 Introduction

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

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

xxxxxxxxxxxxxxxxxxxx

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

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

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

1.1 Notation and Learning Setting

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

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

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 Gentile et al. (2014). However, the rank of the CTR matrix is assumed to be low, that is $d << \min\{L, K\}$. 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. ■

Cla: 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 seqence of i.i.d centered random variables.

  Cla: fix your noise model here. Bernoulli ??

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

Cla: 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 Maillard and Mannor (2014) 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 $|C|$ clusters, indexed by $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_C \in \mathcal{C}$ 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 $|A| \times |B|$ where again the rows in $\mathcal{B}$ can be partitioned into $|C|$ 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.

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

2 `MetaBand`: learning efficiently on a rank-d matrix.

Let $\bar{M} = UV^{\intercal}$, where U is non-negative and V is hott topics. Let j_1^* and j_2^* be the indices of hott-topics vectors. Then

$$(j_1^*, j_2^*) = \arg\max_{j_1,j_2\in[L]} f(\{j_1, j_2\}),$$

where $f(S) = \dfrac{1}{K}\sum_{i\in[K]} \max_{j\in S} R(i,j)$

The key observation is that f is monotone and submodular in S. Therefore, the problem of learning j_1, j_2 online is an online submodular maximization problem.

So, when $d = 2$, $|\mathcal{B}_t| = 2$ and there are two EXP3 Column-Bandits.

After observing the reward r_1, r_2 for $j_1, j_2 \in \mathcal{B}_t$ we update,

EXP_1, $\hat{r}_{1,j_1} = r_1$.

EXP_2, $\hat{r}_{2,j_2} = \max\{r_1, r_2\} - r_1$.

Algorithm 1 Low Rank Bandit Strategy

1: **Input:** Time horizon n, $Rank(\bar{M}) = d$.
2: **for** $t = 1, .., n$ **do**
3: Nature reveals user i_t. ▷ Nature chooses user
4: Column-Bandits suggests $\mathcal{B}_t \subseteq [L]$ items. $|\mathcal{B}_t| = d$
5: **if** Exploration condition satisfied **then**
6: User Bandits suggests each item in $\mathcal{B}_t$, once to user i_t and receive feedback.
7: Update Column-Bandits and User Bandits on feedback received.
8: **else**
9: Suggest best item in $\mathcal{B}_t$ d times to user i_t and receive feedback.

Algorithm 2 Low Rank Bandit Greedy (LRG)

1: **Input:** Time horizon n, $Rank(\bar{R}) = d$.
2: **Explore Parameters:** $\epsilon \in (0, 1)$.
3: **for** $t = 1, .., n$ **do**
4: Nature reveals user i_t. ▷ Nature chooses user
5: Column-EXP3 suggests $\mathcal{B}_t \subseteq [L]$ items. $|\mathcal{B}_t| = d$
6: **With** ϵ probability **do** ▷ Exploration
7: User Bandit suggests each arm $j \in \mathcal{B}_t$ once to user i_t and receive feedback.
8: **Or With** $(1-\epsilon)$ probability **do** ▷ Exploitation
9: User Bandit suggests arm $j \in \arg\max_{j\in\mathcal{B}_t}\left\{\hat{R}(i_t, j)\right\}$, d times to user i_t and receive feedback.
10: Update Column-Bandits and User Bandit on feedback received.

3 Regret Bound

Let us assume that the users are coming in sequentially $i_1, ..., i_n \in [K]$. We use the notation $j^*(i)$ to denote the optimal arm for user i. When $J = (j_1, ..., j_k) \in [L]^k$ is a k-tuple, by $J[l]$ we will mean j_l, the l'th entry of J and $\max R(i, J) := \max_l R(i, J[l])$. 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 time step $t \in [n]$ is $R_t = U_t V_t^T$.

Algorithm 3 Low Rank Bandit UCB (LRUCB)

Input: Time horizon n, $Rank(\bar{R}) = d$.
Definition: $U(i,j) = \sqrt{\dfrac{2\log n}{N_{i,j}}}$.
for $t = 1, .., n$ **do**
  Nature reveals user i_t. ▷ Nature chooses user
  Column-EXP3 suggests $\mathcal{B}_t \subseteq [L]$ items. $|\mathcal{B}_t| = d$
  if $\left(\hat{R}(i_t, j) - U(i_t, j) \leq \hat{R}(i_t, j') + U(i_t, j')\right), \forall j, j' \in \mathcal{B}_t$ **then** ▷ Confidence interval overlap, Exploration
    User Bandit suggests each arm $j \in \mathcal{B}_t$ once to user i_t and receive feedback.
  else ▷ Exploitation
    User Bandit suggests arm $j \in \arg\max_{j \in \mathcal{B}_t} \left\{\hat{R}(i_t, j) + U(i_t, j)\right\}$, d times to user i_t and receive feedback.
  Update Column-Bandits and User Bandits on feedback received.

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 $\alpha_1^j, ..., \alpha_d^j \geq 0, \sum_k \alpha_k^j \leq 1$ and*

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

for every $t \in [n]$.

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

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

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

Todo: What happens with the Bernoulli rounding trick? The above holds when the input is non-stochastic, but I haven't thought about the stochastic case. Without loss of generality, we will also assume that for every $1 \leq k \leq d$, $\max_{J=(j_1,...,j_k) \in [L]^k} \sum_t \max R_t(i_t, J) = \sum_t \max R_t(i_t, J^*[1:k])$

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

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

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

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

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

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

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

and

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

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*

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

We now prove the bounds for $R_c(n)$ and $R_r(n)$ separately. Todo: Make it clear what the randomness is when using $\mathbb{E}$ throughout.

Bounding Column Regret

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

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

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 carrying tildes, we will denote $\tilde{R}_t$ by R_t from hereon.

Lemma 2. *For every $k \in [d]$, we will show that*

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

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

$$\begin{aligned} &\mathbb{E}\sum_t (\max R_t(i_t, J_t[1:k]) - R_t(i_t, J_t[1:k-1])) &(1)\\ &\geq \max_{j_k}\mathbb{E}\sum_t \left(\max R_t(i_t, (J_t[1:k-1], j_k)) - \max R_t(i_t, J_t[1:k-1])\right) - O\left(\sqrt{nL}\right) &(2)\\ &\geq \max_{j_k}\mathbb{E}\sum_t \left(\max R_t(i_t, (J^*[1:k-1], j_k)) - \max R_t(i_t, J_t[1:k-1])\right) - O\left(\sqrt{nL}\right) - O\left((k-1)\sqrt{nL}\right) &(3)\\ &= \mathbb{E}\sum_t \left(\max R_t(i_t, J^*[1:k]) - \max R_t(i_t, J_t[1:k-1])\right) - O\left(k\sqrt{nL}\right). &(4) \end{aligned}$$

The last equality follows from Theorem **??**. The first inequality is from the guarantees of kth col-EXP3. 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 $O\left((k-1)\sqrt{nL}\right)$ term. This follows from induction hypothesis and Lemma **??**. We note that from Equation **??**, we have

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

which concludes the proof. ∎

Lemma 3. *Suppose*

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

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

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

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

$$\begin{aligned}
\mathbb{E} \sum_t \max R_t(i_t, (J_t[1:k-1], j_k)) &= \mathbb{E} \sum_{t \in T_1} \max R_t(i_t, (J_t[1:k-1], j_k)) + \mathbb{E} \sum_{t \in T_2} \max R_t(i_t, (J_t[1:k-1], j_k)) \\
&\geq \sum_{t \in T_1} \max R_t(i_t, (J^*[1:k-1], j_k)) + \mathbb{E} \sum_{t \in T_2} \max R_t(i_t, (J_t[1:k-1], j_k)) \\
&\geq \sum_{t \in T_1} \max R_t(i_t, (J^*[1:k-1], j_k)) + \mathbb{E} \sum_{t \in T_2} \max R_t(i_t, J_t[1:k-1]) \\
&\geq \sum_{t \in T_1} \max R_t(i_t, (J^*[1:k-1], j_k)) + \sum_{t \in T_2} \max R_t(i_t, J^*[1:k-1]) - C_k \\
&= \sum_{t \in T_1} \max R_t(i_t, (j_1^*, j_2)) + \sum_{t \in T_2} \max R_t(i_t, (J^*[1:k-1], j_2)) - C_k \\
&= \sum_{t \in [n]} \max R_t(i_t, (J^*[1:k-1], j_2)) - C_k.
\end{aligned}$$

The first inequality is easy because $\max R_t(i_t, (J^*[1:k-1], j_2)) = R_t(i_t, j_2)$ for $t \in T_1$. Second inequality is trivial. Third inequality follows from the assumption. The next equality holds because of the definition of T_2. ■

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

Bounding Row Regret

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

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

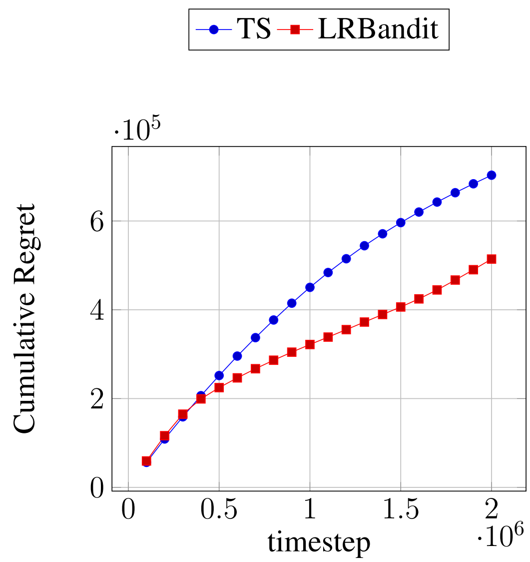
Since there is an EXP3 algorithm on $d+1$ arms for each user, we have

$$R_r(n) = O\left(\frac{Kd \log d}{\gamma} + \gamma n\right),$$

where $\gamma > 0$ is any positive number.

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

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.

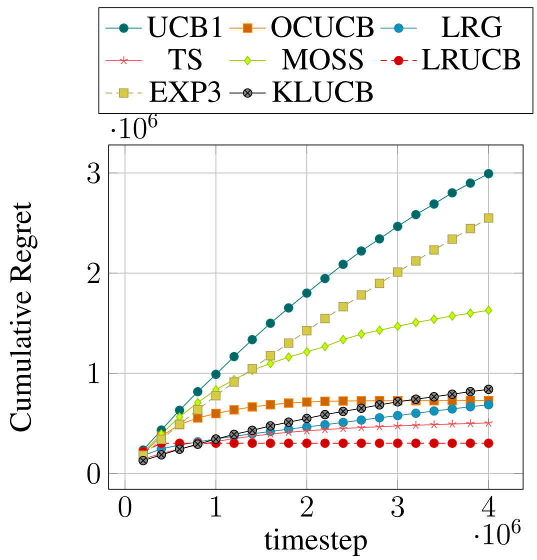
5 Conclusions and Future Direction

To be written.

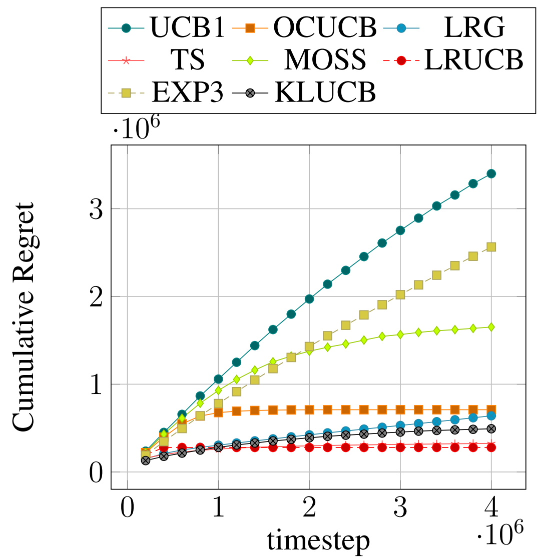
References

Gentile, C., Li, S., and Zappella, G. (2014). Online clustering of bandits. In *International Conference on Machine Learning*, pages 757–765.

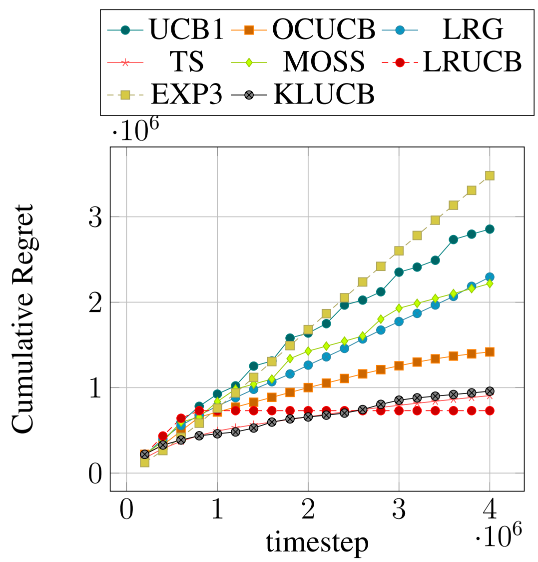
Maillard, O.-A. and Mannor, S. (2014). Latent bandits. In *International Conference on Machine Learning*, pages 136–144.



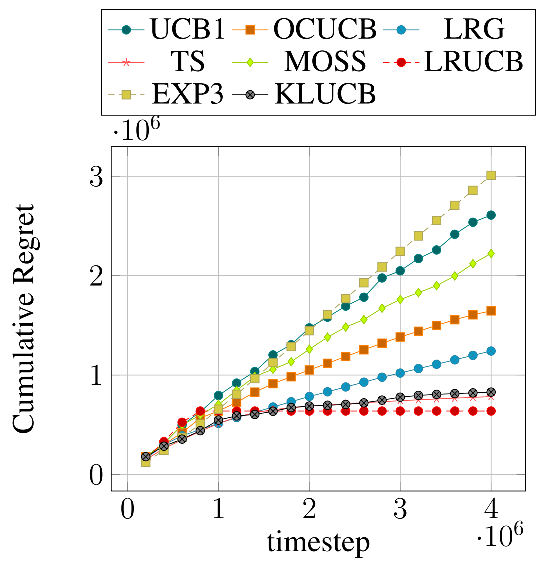
(a) Expt-1: 2048 Users, 64 Bernoulli-distributed arms, Round-Robin, Noisy Setting, Rank 2, equal sized clusters



(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.