Stochastic Low-Rank Latent Bandits

Adobe Advisor(s): Branislav Kveton, Anup Rao, Zheng Wen *Intern: Subhojyoti Mukherjee*

Abstract

To be written.

1 Introduction

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 $\bar{R} \in [0,1]^{K\times L}$, 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 $\bar{R}$ where user is denoted by i_t. The learner selects some items (or columns) from $\bar{R}$, where an item is denoted by $j_t \in [L]$. Then the learner receives one noisy feedback $r_t(i_t, j_t) \sim Ber(\bar{R}(i_t, j_t))$, where Ber is a Binomial distribution over the entries in $\bar{R}$ and $\mathbb{E}[r_t(i_t, j_t)] = \bar{R}(i_t, j_t)$. Then the goal of the learner is to minimize the cumulative regret by quickly identifying the best item j_t^* for each $i_t \in \bar{R}$ where $\bar{R}_{i_t, j_t^*} = \arg\max_{j\in[L]}\{\bar{R}_{i_t,j}\}$.

1.1 Notations, Problem Formulation and Assumptions

We define $[n] = \{1, 2, \dots, n\}$ and for any two sets A and B, A^B denotes the set of all vectors who take values from A and are indexed by B. Let, $R \in [0,1]^{K\times L}$ denote any matrix, then $R(I, :)$ denote any submatrix of k rows such that $I \in [K]^k$ and similarly $R(:, J)$ denote any submatrix of j columns such that $J \in [L]^j$.

Let $\bar{R}$ be reward matrix of dimension $K \times L$ where K is the number of user or rows and L is the number of arms or columns. Also, let us assume that this matrix $\bar{R}$ has a low rank structure of rank $d << \min\{L, K\}$. Let U and V denote the latent matrices for the users and items, which are not visible to the learner such that,

$$\bar{R} = UV^{\intercal} \qquad \textbf{s.t.} \qquad U \in [\mathbb{R}^{+}]^{K\times d}, V \in [0,1]^{L\times d}$$

Furthermore, we put a constraint on V such that, $\forall j \in [L]$, $\|V(j,:)\|_1 \leq 1$.

Assumption 1. *We assume that there exists d-column base factors, denoted by $V(J^*, :)$, such that all rows of V can be written as a convex combination of $V(J^*, :)$ and the zero vector and $J^* = [d]$. We denote the column factors by $V^* = V(J*, :)$. Therefore, for any $i \in [L]$, it can be represented by*

$$V(i, :) = a_i V(J^*, :),$$

where $\exists a_i \in [0,1]^d$ and $\|a_i\|_1 \leq 1$.

Assumption 2. *For each user i_t revealed by the nature at round t, the learner is allowed to select atmost d-items, where d is the rank of the matrix $\bar{R}$.*

The above assumption 2 can be conceptualized in this real-world scenario where the learner has to suggest movies to users and each movie belongs to a different genre (say thriller, romance, comedy, etc). So, the learner can suggest d movies belonging to different genres to each user, and the user can click one, or both, or none of the recommended movies.

The main goal of the learning agent is to minimize the cumulative regret until the end of horizon n. We define the cumulative regret, denoted by $\mathcal{R}_n$ as,

$$\mathcal{R}_n = \sum_{t=1}^{n} \left\{ \sum_{z=1}^{d} \left(r_t\left(i_t, j^*_{t,z}\right) - r_t\left(i_t, j_{t,z}\right) \right) \right\}$$

where, $j^*_{t,z} = \arg\max_{j\in[L]}\{\bar{R}(i_t, j)\}$ and $j_{t,z}$ be the suggestion of the learner for the i_t -th user for $z = 1, 2, \ldots, d$. Note that $r_t\left(i_t, j^*_{z,t}\right) \sim Ber(\bar{R}\left(i_t, j^*_{z,t}\right))$ and $r_t\left(i_t, j_{z,t}\right) \sim Ber(\bar{R}\left(i_t, j_{z,t}\right))$. Taking expectation over both sides, we can show that,

$$\mathbb{E}[\mathcal{R}_n] = \mathbb{E}\left[\sum_{t=1}^{n} \left\{ \sum_{z=1}^{d} \left(r_{z,t}\left(i_t, j^*_{z,t}\right) - r_{z,t}\left(i_t, j_{z,t}\right) \right) \right\}\right] = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{z=1}^{d} \left(N_{i_t, j_{z,t}} \right)\right] \Delta_{i_t, j_{z,t}}$$

where, $\Delta_{i_t, j_{z,t}} = \bar{R}(i_t, j^*_{z,t}) - \bar{R}(i_t, j_{z,t})$ and $N_{i_t, j_{z,t}}$ is the number of times the learner has observed the $j_{z,t}$-th item for the i_t-th user for $z = 1, 2, \ldots, d$. Let, $\Delta = \min_{i\in[K], j\in[L]}\{\Delta_{i,j}\}$ be the minimum gap over all the user, item pair in $\bar{R}$.

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 Contributions

To be written.

3 Proposed Algorithms

Let $\bar{R} = UV^\intercal$, 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

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

Algorithm 2 Low Rank Bandit Greedy (LRG)

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

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.

Algorithm 4 Meta Low Rank Bandit Greedy(MLRG)

Input: Time horizon T, $Rank(\bar{R}) = d$.
Explore Parameters: $\gamma \in (0, 1]$.
Initialization: Initialize all user experts $M_i\left([L]\right), \forall i \in [K]$, $\mathcal{B}_0 \leftarrow \emptyset$, and MetaEXP$_z(\gamma, 0, \mathcal{B}_0), \forall z \in [d]$.
Receive $\mathcal{B}_1$ from MetaEXP$_z(\gamma, 0, \mathcal{B}_0), \forall z \in [d]$ by sampling rule.
for $t = 1, .., T$ **do**
  Nature reveals i_t. ▷ Nature chooses user
  With ϵ probability **do**
    Select all arms $j \in \mathcal{B}_t$ once, where $|\mathcal{B}_t| = d$.
    Update MetaEXP$_z(\gamma, t), \forall z \in [d]$ by *update rule*. ▷ Update MetaEXP$_z(\gamma, t), \forall z \in [d]$
    Receive new $\mathcal{B}_{t+1}$ from MetaEXP$_z(\gamma, t, \mathcal{B}_t), \forall z \in [d]$ by *sampling rule*.
  Or With $1 - \epsilon$ probability **do** ▷ Run User Expert M_{i_t} on $\mathcal{B}_t$ columns for d times
    Select j_t suggested by greedy $M_{i_t}\left(\mathcal{B}_t\right)$ for d times, observe r_{i_t, j_t} and update $M_{i_t}\left(\mathcal{B}_t\right)$.

Algorithm 5 Meta Low Rank Bandit UCB(MLRUCB)

Input: Time horizon T, $Rank(\bar{R}) = d$.
Explore Parameters: $\gamma \in (0, 1]$.
Definition: $U(i,j) = \dfrac{2\log T}{N_{i,j}}$.
Initialization: Initialize all user experts $M_i\left([L]\right), \forall i \in [K]$, $\mathcal{B}_0 \leftarrow \emptyset$, and MetaEXP$_z(\gamma, 0, \mathcal{B}_0), \forall z \in [d]$.
Receive $\mathcal{B}_1$ from MetaEXP$_z(\gamma, 0, \mathcal{B}_0), \forall z \in [d]$ by sampling rule.
for $t = 1, .., T$ **do**
  Nature reveals i_t. ▷ Nature chooses user
  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**
    Select all arms $j \in \mathcal{B}_t$ once, where $|\mathcal{B}_t| = d$.
    Update MetaEXP$_z(\gamma, t), \forall z \in [d]$ by *update rule*. ▷ Update MetaEXP$_z(\gamma, t), \forall z \in [d]$
    Receive new $\mathcal{B}_{t+1}$ from MetaEXP$_z(\gamma, t, \mathcal{B}_t), \forall z \in [d]$ by *sampling rule*.
  else ▷ Run User Expert M_{i_t} on $\mathcal{B}_t$ columns for d times
    Select j_t suggested by $M_{i_t}\left(\mathcal{B}_t\right)$ for d times, observe r_{i_t, j_t} and update $M_{i_t}\left(\mathcal{B}_t\right)$.

Algorithm 6 MetaEXP$_z(\gamma, t, \mathcal{B}_t)$

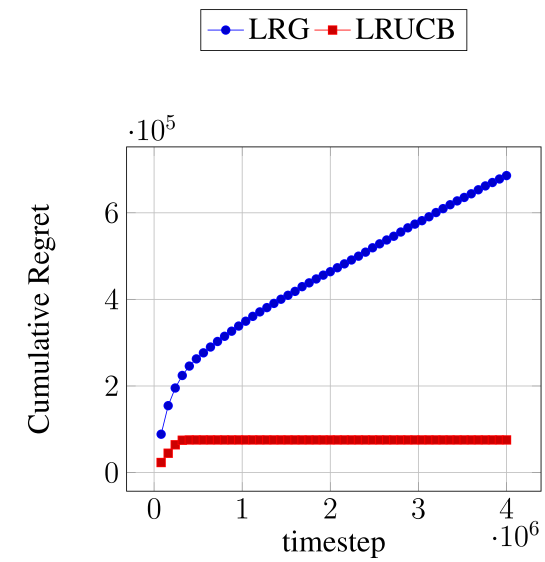
Initialization: Set $w_{z,j}(t=0) = 1, \forall j \in [L]$.
Sampling Rule:
for each $j \in [L]$ **do**

$$p_{z,j}(t) = (1-\gamma)\,\frac{w_{z,j}(t)}{\sum_{j'=1}^{L} w_{z,j'}(t)} + \frac{\gamma}{L}.$$

Suggest $j_{z,t} \notin \mathcal{B}_{t+1}$ by sampling according to the probabilities $p_{z,1}(t), p_{z,2}(t), \ldots, p_{z,L}(t)$.
Update Rule:
for each $j \in [L]$ **do**

$$\hat{r}_j(t) = \begin{cases} \dfrac{r_j(t)}{p_{z,j}(t)}, & \text{if } j = j_{z,t} \\ 0, & \text{otherwise} \end{cases}$$

$$w_{z,j}(t+1) = w_{z,j}(t) \exp\left(\gamma \hat{r}_j(t)\right)$$



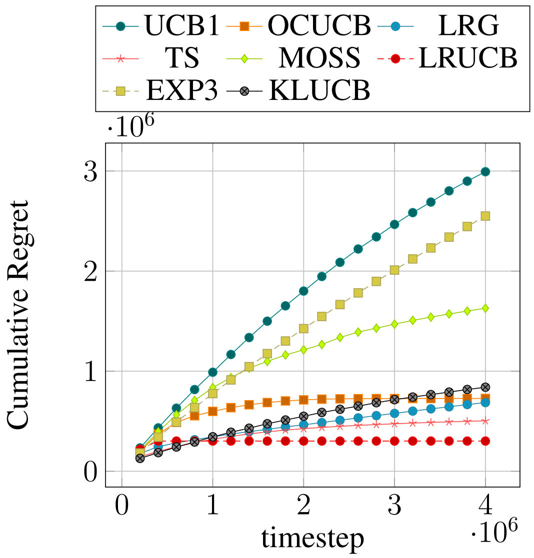
(a) Expt-1: 2048 Users, 64 Bernoulli-distributed arms, Round-Robin, Noisy Setting, Rank 2, equal sized clusters

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

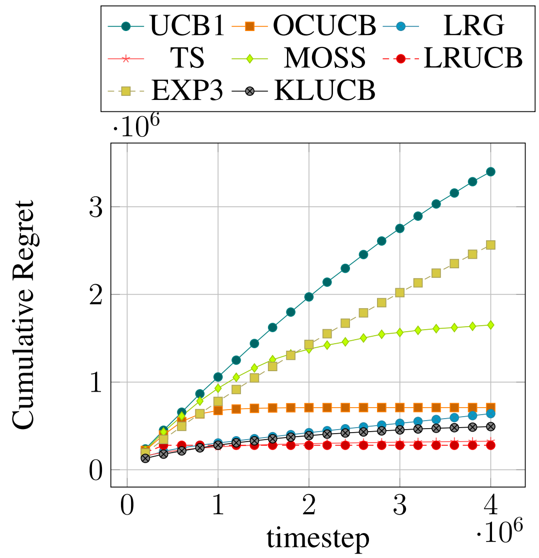
4 Experiments

5 Conclusions and Future Direction

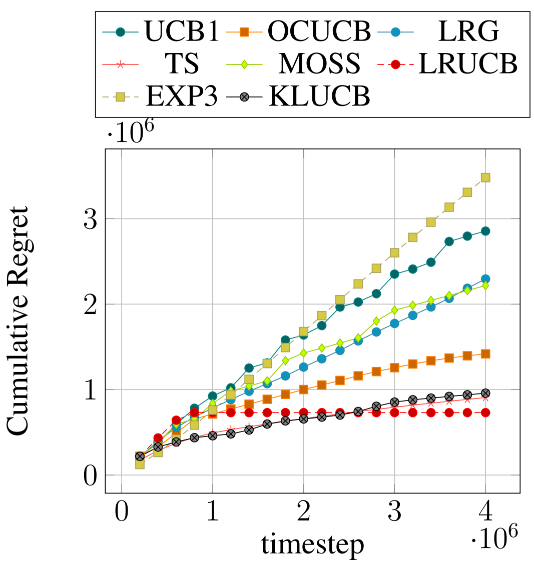
To be written.



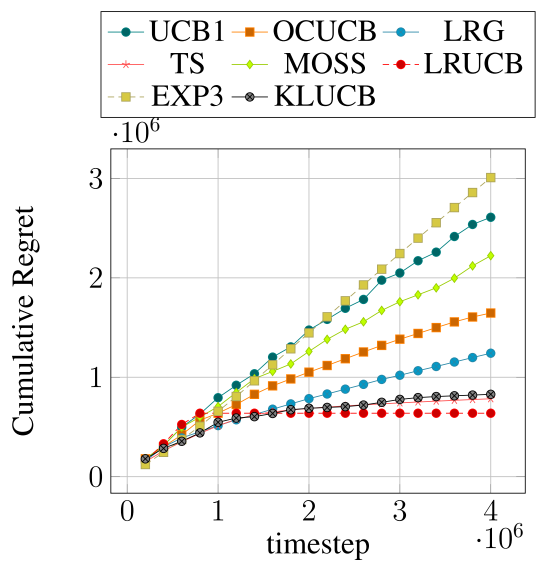
(a) Expt-1: 2048 Users, 64 Bernoulli-distributed arms, Round-Robin, Noisy Setting, Rank 2, equal sized clusters



(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.

References

Maillard, O.-A. and Mannor, S. (2014). Latent bandits.