

Stochastic Low-Rank Latent Bandits

Adobe Advisor(s): Branislav Kveton, Anup Rao, Zheng Wen

Intern: Subhojyoti Mukherjee

Disclaimer: These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this communication only with the permission of the Advisor(s).

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, 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 one item (or column) from \bar{R} , where the item is denoted by j_t . Then the learner receives one noisy feedback $r_t(i_t, j_t) \sim \mathcal{N}(\bar{R}(i_t, j_t), \sigma^2)$, where \mathcal{N} is a distribution over the entries in \bar{R} , σ^2 is variance 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 $\bar{R}_{i_t, j_t^*} = \arg \max_{j_t \in [L]} \{\bar{R}_{i_t, j_t}\}$.

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 $d \ll \min\{L, K\}$. Let U and V denote the latent matrices for the users and items, which are not visible to the learner such that,

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

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 $\|a_i\|_1 \leq 1$.

In this paper, in addition to the noisy setting explained in section 1 we first analyze the proposed algorithm in the easier noise free setting. In the noise free setting, the nature reveals the row i_t , and when the learner selects the column j_t , it observes the mean of the distribution $\bar{R}(i_t, j_t)$.

Assumption 2. *We assume that nature is revealing the user i in $\bar{R}(i, :)$, $\forall i \in [K]$ in a Round-Robin fashion such that at timestep t , nature reveals $i_t = (t \bmod K) + 1$.*

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

$$\mathcal{R}_T = \sum_{t=1}^T \{r_t(i_t, j_t^*) - r_t(i_t, j_t)\}$$

where, $j_t^* = \arg \max_{j \in [L]} \{\bar{R}(i_t, j)\}$ and j_t be the suggestion of the learner for the i_t -th user. Taking expectation over both sides, we can show that,

$$\mathbb{E}[\mathcal{R}_T] = \mathbb{E} \left[\sum_{t=1}^T \{r_t(i_t, j_t^*) - r_t(i_t, j_t)\} \right] = \mathbb{E} \left[\sum_{t=1}^T n_{i_t, j_t} \right] \Delta_{i_t, j_t}$$

where, $\Delta_{i_t, j_t} = \bar{R}(i_t, j_t^*) - \bar{R}(i_t, j_t)$ and n_{i_t, j_t} is the number of times the learner has observed the j_t -th item for the i_t -th user. 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 $|\mathcal{C}|$ 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 $X_{a, b, t}$ 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.

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

We propose two algorithms, one each for noise free and noisy setting. These algorithms are based on UCB-Improved, which is an arm elimination algorithm from Auer and Ortner (2010) and is suitable for the stochastic bandit setting. Both these algorithms are phase based column (arm) elimination algorithms where in each phase we select all the surviving columns equal number of times and then eliminate some sub-optimal columns based on some elimination criteria.

The algorithms are initialized with the estimate $\hat{R}_{i,j} = 0, \forall i \in [K], j \in [L]$. In the m -th phase, we denote the set of surviving columns as \mathcal{B}_m , the set of explored columns as \mathcal{Z}_m . The rows (users) are divided into equivalence classes which are contained in \mathcal{C} . We denote each equivalence class as \mathcal{G}_b , where b is indexed from $1, 2, \dots, \frac{K}{\gamma}$ and these class are contained in \mathcal{C} . N_m denotes the phase length for the m -th phase and each phase length consist of $\gamma|\mathcal{B}_m|n_m$, where $\gamma = \lceil \sqrt{K} \rceil$ is the exploration parameter and n_m is the number of times we select each surviving columns. In the column elimination sub-module we eliminate a sub-optimal column by making sure that it is not one of the d -best columns. In the reset parameters sub-module, we increase the exploration bonus for the next phase so that more exploration is conducted for the surviving columns. Note, that for these two sub-modules, the noise free and noisy setting has two different approaches which will be explained in subsection 3.1 and 3.2 respectively. Finally, if the algorithm has eliminated $L - d$ columns, it fully explores the d best columns (in the noise free setting) and for all the users it always selects the column j_t^* , where $j_t^* \leftarrow \arg \max_{j \in [\mathcal{B}_m]} \hat{R}(i_t, j)$. Whereas, in the noisy setting, the GLB-UCB runs the UCB1 algorithm for the remaining d columns.

3.1 Noise Free Setting

In the noise free setting, since at every pull of a column j for a user i , the algorithm observes the expected reward $\hat{R}_{i,j} = \bar{R}_{i,j}$, so $n_0 = n_m = 1, \forall m$. Furthermore, we do not require any confidence interval in the noise free setting for column elimination sub-module. The pseudo-code of this is shown in Algorithm 1.

3.2 Noisy setting

In the noisy setting, at every pull the algorithm observes a random reward which is drawn from the distribution $\mathcal{N}(\bar{R}_{i,j}, \sigma^2)$, so the algorithm samples each column more number of times (n_m) in each phase and increases the exploration from phase to phase. Moreover, in the column elimination sub-module, the confidence interval $U_m(\epsilon_m, n_m)$ helps in eliminating a sub-optimal column with high probability in the noisy setting. The pseudo-code of this is shown in Algorithm 2.

4 Main Results

Lemma 1. For any arbitrary row $i \in [K]$,

$$\arg \max_{j \in [L]} U(i, :) V(j, :)^T \leq \arg \max_{j \in [d]} U(i, :) V(j, :)^T.$$

Algorithm 1 Noise-Free GLB

```

1: Input: Time horizon  $T$ ,  $\text{Rank}(\bar{R}) = d$ .
2: Explore Parameter:  $\gamma = \lceil \sqrt{K} \rceil$ .
3: Initialization:  $\hat{R}(i, j) \leftarrow 0, \forall i \in [K], j \in [L], m = 0, \mathcal{B}_0 \leftarrow \mathcal{A}, \mathcal{Z}_0 \leftarrow \emptyset, \mathcal{C} \leftarrow \emptyset, i_0 = 1, j_0 = 1, n_0 = 1,$ 
    $Nx_0 = Kn_0d$  and  $N_0 = Nx_0 + \gamma|\mathcal{B}_0|n_0$ .
4: for each  $b \in \left[0, \frac{K}{\gamma} - 1\right]$  do (Create equivalence class  $\mathcal{C}$ )
5:    $\mathcal{G}_{b+1} \leftarrow \left\{i \in \left[\frac{bK}{\gamma} + 1, \frac{(b+1)K}{\gamma}\right]\right\}$  and  $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{G}_{b+1}$ .
6: for  $t = 1, \dots, T$  do
7:   Nature reveals  $i_t$  such that  $i_t \leftarrow (t \bmod K) + 1$  (Round-Robin).
8:   if  $|\mathcal{B}_m| > d$  then
9:     if  $t \leq N_m$  and  $t \leq Nx_m$  then (Phase Exploit)
10:      Choose  $j_0 \leftarrow \arg \max_{j \in \mathcal{B}_m} \left\{ \hat{R}(i_t, j_0) \right\}$ , observe  $\bar{R}(i_t, j_0)$  and  $\hat{R}(i_t, j_0) \leftarrow \bar{R}(i_t, j_0)$ .
11:     else if  $t \leq N_m$  and  $t > Nx_m$  then (Phase Explore)
12:       if  $i_0 \leq \gamma$  then
13:         Choose  $j_0$ , observe  $\bar{R}(i_0, j_0)$  and  $\hat{R}(i_0, j_0) \leftarrow \bar{R}(i_0, j_0)$ .
14:          $i_0 \leftarrow i_0 + 1$ .
15:       else
16:          $\mathcal{Z}_m \leftarrow \mathcal{Z}_m \cup \{j_0\}$ 
17:         Choose  $j_0 \in \mathcal{B}_m \setminus \mathcal{Z}_m$  uniform randomly and  $i_0 \leftarrow 1$ .
18:     else
19:       Column Elimination
20:
21:       for each  $\mathcal{G}_b \in \mathcal{C}$  do
22:
23:         while  $\exists j \in \mathcal{B}_m$  such that  $\forall i \in \mathcal{G}_b : \hat{R}(i, j) < \max_{j' \in \mathcal{B}_m \setminus j} \{ \hat{R}(i, j') \}$  do
24:            $\mathcal{B}_m \leftarrow \mathcal{B}_m \setminus \{j\}$ .
25:
26:       Reset Parameters
27:        $Nx_{m+1} \leftarrow t + Kn_0d$ 
28:        $N_{m+1} \leftarrow Nx_{m+1} + \gamma|\mathcal{B}_m|n_0$  and  $m \leftarrow m + 1$ .
29:
30:   else (Full Exploit)
31:     Select column  $j_t^*$ , observe  $\bar{R}(i_t, j_t^*)$  where  $j_t^* \leftarrow \arg \max_{j \in [\mathcal{B}_m]} \hat{R}(i_t, j)$  and  $\hat{R}(i_t, j_t^*) \leftarrow \bar{R}(i_t, j_t^*)$ .

```

Algorithm 2 Noisy GLB-UCB

```

1: Input: Time horizon  $T$ ,  $\text{Rank}(\bar{R}) = d$ .
2: Explore Parameter:  $\gamma = \lceil \sqrt{K} \rceil$ ,  $\psi = T$ .
3: Definition:  $U_m(\epsilon_m, n_m) = \sqrt{\frac{\psi \log(T\epsilon_m^2)}{2n_m}}$ 
4: Initialization:  $\forall i \in [K], j \in [L], \hat{R}(i, j) \leftarrow 0, m = 0, \mathcal{B}_0 \leftarrow \mathcal{A}, \mathcal{Z}_0 \leftarrow \emptyset, \mathcal{C} \leftarrow \emptyset, i_0 = 1, j_0 = 1, \epsilon_0 = 1,$ 
 $M = \frac{1}{2} \log_2\left(\frac{T}{\epsilon}\right), n_0 = \frac{2 \log(\psi T \epsilon_0^2)}{\epsilon_0}, Nx_0 = Kn_0d$  and  $N_0 = Nx_0 + \gamma|\mathcal{B}_0|n_0$ .
5: for each  $b \in \left[0, \frac{K}{\gamma} - 1\right]$  do (Create equivalence class  $\mathcal{C}$ )
6:    $\mathcal{G}_{b+1} \leftarrow \left\{i \in \left[\frac{bK}{\gamma} + 1, \frac{(b+1)K}{\gamma}\right]\right\}$  and  $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{G}_{b+1}$ .
7: for  $t = 1, \dots, T$  do
8:   Nature reveals  $i_t$  such that  $i_t \leftarrow (t \bmod K) + 1$  (Round-Robin).
9:   if  $|\mathcal{B}_m| > d$  then
10:    if  $t \leq N_m$  and  $t \leq Nx_m$  then (Phase Exploit)
11:      Choose  $j_0 \leftarrow \arg \max_{j \in \mathcal{B}_m} \left\{ \hat{R}(i_t, j) + \sqrt{\frac{2 \log T}{n_{i_t, j}}} \right\}$ , observe  $r_t(i_t, j_0) \sim \mathcal{N}(\bar{R}(i_t, j_0), \sigma^2)$  and
 $\hat{R}(i_t, j_0) \leftarrow \frac{\sum_{s=1}^t r_s(i_t, j_0) [\mathbb{I}_{s, i_t} = j_0]}{n_{i_t, j_0}}$ .
12:    else if  $t \leq N_m$  and  $t > Nx_m$  then (Phase Explore)
13:      if  $i_0 \leq \gamma$  then
14:        Choose  $j_0$ , observe  $r_t(i_0, j_0) \sim \mathcal{N}(\bar{R}(i_0, j_0), \sigma^2)$  and  $\hat{R}(i_0, j_0) \leftarrow \frac{\sum_{s=1}^t r_s(i_0, j_0) [\mathbb{I}_{s, i_0} = j_0]}{n_{i_0, j_0}}$ .
15:         $i_0 \leftarrow i_0 + 1$ .
16:      else
17:         $\mathcal{Z}_m \leftarrow \mathcal{Z}_m \cup \{j_0\}$ 
18:        Choose  $j_0 \in \mathcal{B}_m \setminus \mathcal{Z}_m$  uniform randomly and  $i_0 \leftarrow 1$ .
19:      else
20:        Column Elimination
21:
22:        for each  $\mathcal{G}_b \in \mathcal{C}$  do
23:
24:          while  $\exists j \in \mathcal{B}_m$  such that  $\forall i \in \mathcal{G}_b : \hat{R}(i, j) + U_m(\epsilon_m, n_m) < \max_{j' \in \mathcal{B}_m \setminus j} \{\hat{R}(i, j') - U_m(\epsilon_m, n_m)\}$ 
do
25:             $\mathcal{B}_m \leftarrow \mathcal{B}_m \setminus \{j\}$ .
26:
27:        Reset Parameters
28:         $\epsilon_{m+1} \leftarrow \frac{\epsilon_m}{2}$  and  $n_{m+1} \leftarrow \frac{2 \log(\psi T \epsilon_{m+1}^2)}{\epsilon_{m+1}}$ 
29:         $Nx_{m+1} \leftarrow t + Kn_{m+1}d$ 
30:         $N_{m+1} \leftarrow Nx_{m+1} + \gamma|\mathcal{B}_m|n_{m+1}$  and  $m \leftarrow m + 1$ .
31:
32:      else (Full Exploit)
33:        Select column  $j_t^*$ , observe  $r_t(i_t, j_t^*) \sim \mathcal{N}(\bar{R}(i_t, j_t^*), \sigma^2)$  where  $j_t^* \leftarrow$ 
 $\arg \max_{j \in [\mathcal{B}_m]} \left\{ \hat{R}(i_t, j) + \sqrt{\frac{2 \log T}{n_{i_t, j}}} \right\}$  and  $\hat{R}(i_t, j_t^*) \leftarrow \frac{\sum_{s=1}^t r_s(i_t, j_t^*) [\mathbb{I}_{s, i_t} = j_t^*]}{n_{i_t, j_t^*}}$ .

```

5 Proofs

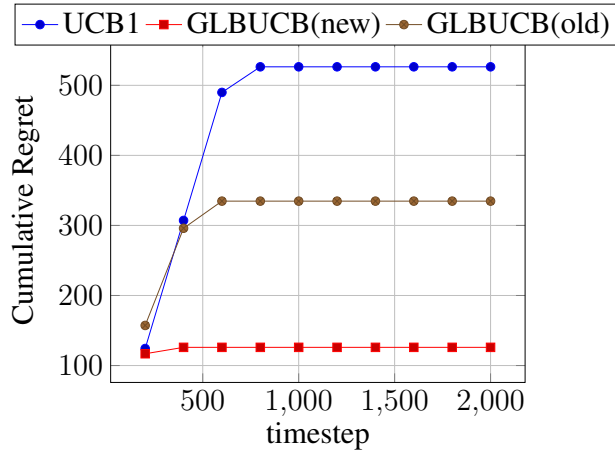
Proof. Considering any arbitrary row $i \in [K]$, we can show that,

$$\begin{aligned}
 \arg \max_{j \in [L]} U(i, :) V(j, :)^{\top} &= U(i, :) V(j^*(i), :)^{\top} \\
 &\stackrel{(a)}{=} U(i, :) (a_{j^*(i)} V(J^*, :))^{\top} \\
 &= \sum_{k=1}^d a_{j^*(i)}(k) U(i, :) V(j^*(i), :)^{\top} \\
 &\leq \arg \max_{k \in [d]} a_{j^*(i)}(k) U(i, :) V(k, :)^{\top} \\
 &\leq \arg \max_{k \in [d]} U(i, :) V(k, :)^{\top},
 \end{aligned}$$

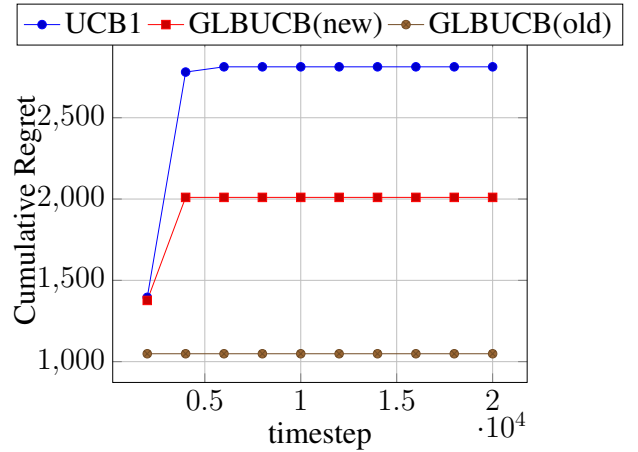
where (a) is from Assumption 1. □

Proof. Step 1: We denote the number of times □

6 Experiments



(a) Expt-1: 64 Users, 10 Bernoulli-distributed arms, Round-Robin, Noise-Free Setting, Well-Structured Environment



(b) Expt-2: 64 Users, 64 Bernoulli-distributed arms, Round-Robin, Noise-Free Setting, Deformed Environment

Figure 1: A comparison of the cumulative regret incurred by the various bandit algorithms.

7 Conclusions and Future Direction

To be written.

References

- Auer, P. and Ortner, R. (2010). Ucb revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65.
- Maillard, O.-A. and Mannor, S. (2014). Latent bandits.