Ranking In Generalized Linear Bandits

Amitis Shidani
George Deligiannidis, Arnaud Doucet
Department of Statistics
University of Oxford
shidani, deligian, doucet@stats.ox.ac.uk

Abstract

We study the ranking problem in generalized linear bandits. At each time, the learning agent selects an ordered list of items and observes stochastic outcomes. In recommendation systems, displaying an ordered list of the most attractive items is not always optimal as both position and item dependencies result in a complex reward function. A very naive example is the lack of diversity when all the most attractive items are from the same category. We model the position and item dependencies in the ordered list and design UCB and Thompson Sampling type algorithms for this problem. Our work generalizes existing studies in several directions, including position dependencies where position discount is a particular case, and connecting the ranking problem to graph theory.

1 Introduction

The *multi-armed bandit* (MAB) problem is a sequential decision-making problem in which there are K possible choices called arms, each with an unknown reward distribution. At each time step t, the decision-maker can choose one arm and see a reward sample drawn from its distribution. The goal is to minimize the regret, which in the simplest case is defined as the difference between the total expected reward when playing the optimal action over time horizon T and the total expected reward collected by the decision-maker (Gittins 1979; Lattimore and Szepesvári 2020). In the classic MAB problem, the arms are assumed independent. However, in the real world, arms are often dependent; pulling one arm gives information about the others. In cases like recommendation systems, the goal is to show an ordered list of items that best engage with the users and provide more rewards (i.e., clicks, watch time). To incorporate arm dependencies, (Lykouris, Tardos, and Wali 2020), (Singh et al. 2020), and (Buccapatnam, Eryilmaz, and Shroff 2014) use the graph-based feedback setting based on the work of (Mannor and Shamir 2011). In this work, when the learner selects arm a, they also observe the rewards of all adjacent arms. (Gupta et al. 2021) introduced another approach where rewards obtained by pulling different arms are correlated.

On the ranking problem, (Radlinski, Kleinberg, and Joachims 2008) proposes algorithms that learn a marginal utility for each document at each rank separately by either exploring and then committing to the best arms or running separate bandit algorithms for each position. (Slivkins, Radlinski, and Gollapudi 2013) introduced a contextual bandit algorithm, where, the context for each position is the event that previous items have not been clicked. This work was generalized in (Ermis et al. 2020; Lagrée, Vernade, and Cappe 2016) using contextual bandits for position-based models. (Lattimore and Szepesvári 2020) and (Gauthier, Gaudel, and Fromont 2022) introduce a more general approach for click models (Chuklin, Markov, and de Rijke 2015) where the objective is to identify the most attractive list. However, these works do not use any information on the items' similarities. In (Li et al. 2016), the authors proposed a more general cascading bandit model using the position discount and contextual information between arms.

1.1 Our Contribution

In previous works, the expected reward function is non-decreasing with respect to items' attractiveness, i.e. if the user finds item a more attractive than item a', any ordered list with item a' replaced by item a provides a higher expected reward. This assumption can be very restrictive since the expected reward function may not always be monotonic in real scenarios; a realistic example is when the attractiveness of an item depends on neighboring items.

Here, we generalize the previous works in two ways. First, the reward function we propose can be non-monotonic, addressing the abovementioned issue. Second, in recommendation systems, it is reasonable to assume that items receive different levels of attention from users in different positions, i.e. having different attractiveness in each position. Discount factor (Li et al. 2016) is one way to address this. However, item-position dependency may be more complicated. Therefore, we let items share different contextual information at each position. We propose a novel graph-based ranking solution. To the best of our knowledge, this is the first work addressing these issues.

2 Notation and Setting

Let $A = \{1, \dots, K\}$ be the finite set of arms, where for each arm i there exists a vector $v_i \in \mathbb{R}^d$. We have L slots available, for which we want to find the best-ordered list of L items, where $L \ll K$. At each round $t \in [T]$ the learner chooses an ordered list of L arms called an *action* $a_t = \{a_t^1, \dots, a_t^L\} \in \mathcal{A}$, where for each i, $a_t^i \in A$ and $\mathcal{A}$ denotes the set of all the possible actions. At the end of each round, the learner observes the sample reward $r^l_{a_t}$ for each position $1 \leq l \leq L$. The goal is to minimize the expected regret $\mathcal{R}_T$ over the time horizon T; i.e. we have:

$$\mathcal{R}_T = \mathbb{E}\left[\sum_{t=1}^{T} \max_{a \in \mathcal{A}} \sum_{l=1}^{L} \mathbb{E}\left[r^l_a\right] - \sum_{t=1}^{T}\sum_{l=1}^{L} r^l_{a_t}\right].$$

Choosing the optimal L-tuple of items is NP-hard since it is equivalent to the maximum coverage problem (Nemhauser, Wolsey, and Fisher 1978). The standard greedy algorithm for this problem translates to iteratively choosing the items with the highest reward, which is what (Radlinski, Kleinberg, and Joachims 2008) does. A simple scenario to address this is to formulate the problem with K^L arms and $d \times L$ dimension. This setting is reducible to the standard finite-arm generalized linear bandit with the total reward of L positions as $f(\langle\theta, v_{a_t}\rangle)$, which results in $\tilde{O}(L\sqrt{dT\log(K)})$ regret. However, we propose another way that allows for different function behaviors f^l for each position and cannot be reduced to the classical setting and has $\tilde{O}(L\sqrt{dT})$ regret. For each action a_t at round t and position l, we assume the reward function $r^l_{a_t}$ to be as follows:

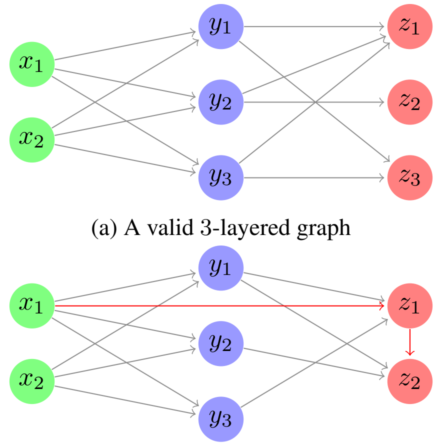
$$r^l_{a_t} = f^l(\langle\theta^l\ ,\ v_{a^l_t} + w_l v_{a_t{}^{l-1}}\rangle) + \eta^l_t, \qquad (1)$$

where $f^l : \mathbb{R} \mapsto \mathbb{R}$ is a continuous and differentiable function called the *link function*, $\langle\cdot\ ,\ \cdot\rangle : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the Euclidean inner product, $\theta^l \in \Theta$ is an unknown d-dimensional vector for position l from the convex compact set $\Theta \subseteq \mathbb{R}^d$, $w_l \in \mathbb{R}$ is a known parameter measuring the dependency of the reward function at position l to the item in the previous position, and $\{\eta^l_t\}_{t,l}$ is a family of centered, independent 1-subgaussian random variables. We denote the history *before* the learner chooses action at time t by $\mathcal{H}_t$.

Equation 1 assumes that the reward at position l depends on the attractiveness of both the items at positions l and $l-1$. The result can be generalized to a window of neighboring items instead (see Appendix D.1). Moreover, the parameter w_l can be negative allowing the reward function to be non-monotonic. Finally, θ^l allows the arms to share contextual information at each position. Different parameters and reward functions at each position allow us to model a more general case of discount factors, i.e., model different users' behavior for each position. The discount factor model is a special case of $f^l(x) = x$ and $\theta^{l+1} = d_l\theta^l$, where d_l denotes the discount parameter.

For the first position, as there is no previous item, one approach is to assume that $v_{a^0_t} = 0$ or $w_1 = 0$, for any action a_t. However, we could recommend a list based on the user's last action in a movie recommendation system. In this case, $v_{a^0_t}$ would be a vector embedding the user's last action, and w_1 would indicate its *importance*, i.e. the degree to which it affects the list. We denote $v_{a^0_t}$ by v_0 in the rest of this paper.

We can now reformulate the problem by defining a new set of arms, called "super-arms of set A", as pairs of arms, i.e. (i, j) where $i, j \in A$, denoting the items in the previous and present positions respectively. As there is no previous item for the first position, we denote the corresponding super-arms by $(0, i)$. We now propose a graph-based approach, which finds the best-ordered list using super-arms.



(a) A valid 3-layered graph

(b) An invalid 3-layered graph

Figure 1: An illustration of a Valid (1a) and an Invalid (1b) 3-Layered Graph. The graph in 1b is invalid due to the red edges that violate conditions (a) and (b) of Definition 1.

3 The Graph-Based Approach for Ranking

Let us start by defining the L-layered graph. All graphs we consider are weighted directed graphs.

Definition 1. *The directed graph $G = (V, E)$ is "L-layered" if and only if (a) $V = \bigcup_{j=1}^{L} V_j$, where $V_i \cap V_j = \emptyset$ for $i \neq j$, (b) all edges $e \in E$ have the form $e = (v, w)$ where $v \in V_l, w \in V_{l+1}$ for some $0 \leq l \leq L-1$, (c) there are no edges $e = (v, w)$ with $v \in V_0$, and (d) l-th layer, V_l, with $l \geq 2$, consists of the nodes with a depth of exactly $l-1$ from the nodes of the first layer V_1.*

An illustration of a valid and an invalid 3-layered graph is presented in Figure 1. Now, we want to build a L-layered graph G using the super-arms defined in the previous section. We add K vertices to the first layer and K^2 vertices for each layer $2 \leq l \leq L$. We denote the nodes at layer one as u^1_{0i} for $i \in [K]$, corresponding to the super-arm $(0, i)$; at layer l we write u^l_{ij}, where $1 \leq i, j \leq K$ for the vertex assigned to super-arm (i, j) at position l. We connect the vertex $u^l_{ij}, l \in [L-1]$, to all the vertices u^{l+1}_{jq}, where $q \in [K]$. It is not hard to see that G is L-layered. Additionally, note that $G = \bigcup_{i=1}^{K} G_i$, where G_i is the induced subgraph of G that includes all the paths of G containing u^1_{0i}.

Next, we define the weights of the edges for the weighted graph G and the vector $\theta = (\theta^1, \dots, \theta^L)$. If e is an edge between vertices u^l_{ij} and u^{l+1}_{jq}, then the weight of e denoted by c_e is defined as follows:

$$c_e = \begin{cases} \frac{1}{2}\big(2f^1(\langle \theta^1 \ , \ v_j + w_1 v_0 \rangle) \\ \qquad + f^2(\langle \theta^2 \ , \ v_q + w_2 v_j \rangle)\big) & \text{if } l = 1; \\ \frac{1}{2}\big(f^{L-1}(\langle \theta^{L-1} \ , \ v_j + w_{L-1} v_i \rangle) \\ \qquad + 2f^L(\langle \theta^L \ , \ v_q + w_L v_j \rangle)\big) & \text{if } l = L-1; \\ \frac{1}{2}\big(f^l(\langle \theta^l \ , \ v_j + w_l v_i \rangle) \\ \qquad + f^{l+1}(\langle \theta^{l+1} \ , \ v_q + w_{l+1} v_j \rangle)\big) & \text{otherwise.} \end{cases} \tag{2}$$

We call this process of building G as "L-layering" over super-arms of set A and vector θ. Now, consider a path p with L vertices. In the L-layered graph, a path starts with one of the first layer vertices and ends at a vertex from the L-th layer resulting in a sequence of the form $\{u^1_{0i_1}, u^2_{i_1 i_2}, \ldots, u^L_{i_{L-1} i_L}\}$. The sum of the weights of this path is equal to the expected reward of playing the action $(i_1, \ldots, i_L)$. The interesting thing about this graph is that every path with L vertices provides a valid ordered list for the main ranking problem. Moreover, the problem of finding the best-ordered list corresponds to finding the longest weighted path. Two problems arise, (1) the time complexity of finding the longest weighted path needs to be controlled, and (2) the vector θ (i.e. the reward functions r^l) is unknown.

Longest Weighted Path. Finding the longest weighted path of an arbitrary graph G is NP-hard (Sedgewick and Wayne 2011). However, if G is a directed acyclic graph, then no negative cycles can be created, and the longest path in G can be determined in linear time by finding the shortest path in $-G$ (replacing every weight with its opposite) (Cormen et al. 2022). If G is L-layered, it is also a directed acyclic graph, and we can find the shortest path which gives us the best-ordered list of items. Moreover, the L-layered property already gives a topological ordering for the graph G, which helps the shortest path algorithms. Also, note that we can reduce the time complexity of finding the shortest path of G by running the algorithm for each G_i in parallel and then comparing the shortest paths of sub-graphs. For instance, if we use Dijkstra's algorithm (Sniedovich 2006), the worst-case running time complexity would be $O\left(|E_G| + |V_G| \log(|V_G|)\right)$, where $|E_G|$ and $|V_G|$ represent the number of edges and the number of vertices of graph G. For the L-layered graph G with K arms, it would be $O(K^3)$.

Unknown Vector θ. To find the longest path, we need to know the weights of the graph, which is not possible if the vector θ is unknown. This situation is similar to the MAB problem where we cannot play the optimal action from the beginning. In the MAB setting, we estimate the expected reward for each action at each round and then play the action with the highest expected reward. We will take a similar approach here. For any algorithm that estimates the expected reward of each super-arm, we will be able to find the longest path in the graph with these estimated weights. At each round, we update the weights based on the reward history of the super-arms. When the algorithm converges to the actual values of the expected reward, the longest path would also converge to the best-ordered list.

We use the L-layered technique to find the best-ordered list in the next section by adapting famous algorithms, UCB (Auer, Cesa-Bianchi, and Fischer 2002). In Appendix C we apply Thompson Sampling (Russo et al. 2018).

4 Ranking UCB Algorithm

We first explain the main idea behind the algorithm. By Equation 1, we have:

$$\mathbb{E}\left[r^l_a | \mathcal{H}_t\right] = f^l({\theta^l}^{\mathrm{T}}(v_{a^l} + w_l v_{a^{l-1}})).$$

We estimate θ^l and the expected reward using the Maximum Likelihood Estimator (MLE) in the classical likelihood theory of generalized linear models (McCullagh and Nelder 1989) with samples $x^l_t = v_{a^l_t} + w_l v_{a^{l-1}_t}$ and labels $r^l_{a_t}$.

Then, we construct a confidence set $\mathcal{C}^l_t \subset \mathbb{R}^d$ that contains the unknown parameter θ^l with high probability. (Filippi et al. 2010) was the first to study generalized linear bandits using UCB methods. However, the bound is not optimal with respect to T. Here, we use an approach that provides the optimal bound. First, let us define the following variables:

$$g^l_t(\theta) = \lambda\theta + \textstyle\sum_{s=1}^t f^l(\langle \theta \ , \ x^l_s \rangle) x^l_s \tag{3}$$

$$L^l_t(\theta) = \|g^l_t(\theta) - \textstyle\sum_{s=1}^t r^l_s x^l_s\|_{{V^l_t}^{-1}} \tag{4}$$

where $V^l_0(\lambda) = \lambda I$, $V^l_t(\lambda) = V^l_0(\lambda) + \sum_{s=1}^t x^l_s {x^l_s}^{\mathrm{T}}$, and $x^l_s = v_{a^l_s} + w_l v_{a^{l-1}_s}$. Note that $V^l_t(\lambda) \in \mathbb{R}^{d \times d}$ is a symmetric strictly positive definite matrix, and for any strictly positive definite matrix V, a norm on $\mathbb{R}^d$ is given by $\|x\|_V = (x^{\mathrm{T}} V x)^{\frac{1}{2}}$. Now, we have the following lemma:

Lemma 1. *Let* $\delta \in (0, 1)$, *and* $\sqrt{\beta^l_t} = \sqrt{\lambda}\|\theta^l\|_2 + \sqrt{2\log\left(\frac{1}{\delta}\right) + \log\left(\frac{\det\left(V^l_t(\lambda)\right)}{\lambda^d}\right)}$. *Define* $\mathcal{C}^l_t$ *as follows:*

$$\mathcal{C}^l_t = \left\{\theta \in \Theta : L^l_t(\theta) \leq \sqrt{\beta^l_t}\right\} \tag{5}$$

Then, with probability at least $1 - \delta$, *it holds that for any time t,* $\theta^l \in \mathcal{C}^l_t$; *i.e.* $\mathbb{P}(\exists t : \theta^l \notin \mathcal{C}^l_t) \leq \delta$.

The proof can be found in Appendix A.1 which uses the super-martingale technique introduced in (Abbasi-Yadkori, Pál, and Szepesvári 2011). Now, we can define the optimistic estimated reward for any super-arm (i, j) and position l in the UCB algorithm as follows:

$$\mathrm{UCB}^l_t(i, j) = \max\nolimits_{\theta \in \mathcal{C}^l_t} f^l\left(\langle \theta \ , \ v_j + w_l v_i \rangle\right). \tag{6}$$

Then, we use Equation 2 to build the L-layering graph G over the super-arms and the estimated rewards. Namely, at each round t, Equation 6 allows us to replace the weight c_e for the edge $e = (u^l_{ij}, u^{l+1}_{jq})$ by the estimate $\hat{c}_e$:

$$\hat{c}_e = \begin{cases} \frac{1}{2}(2\mathrm{UCB}^1_t(0, j) + \mathrm{UCB}^2_t(j, q) & \text{if } l = 1; \\ \frac{1}{2}(\mathrm{UCB}^{L-1}_t(i, j) + 2\mathrm{UCB}^L_t(j, q)) & \text{if } l = L-1; \\ \frac{1}{2}(\mathrm{UCB}^l_t(i, j) + \mathrm{UCB}^{l+1}_t(j, q)) & \text{otherwise.} \end{cases} \tag{7}$$

Finding the longest path of G leads us to the best-ordered list for each round t using the UCB algorithm. The complete algorithm, RankUCB, is described in Algorithm 1. We will now provide a regret bound for the RankUCB algorithm under the following assumptions:

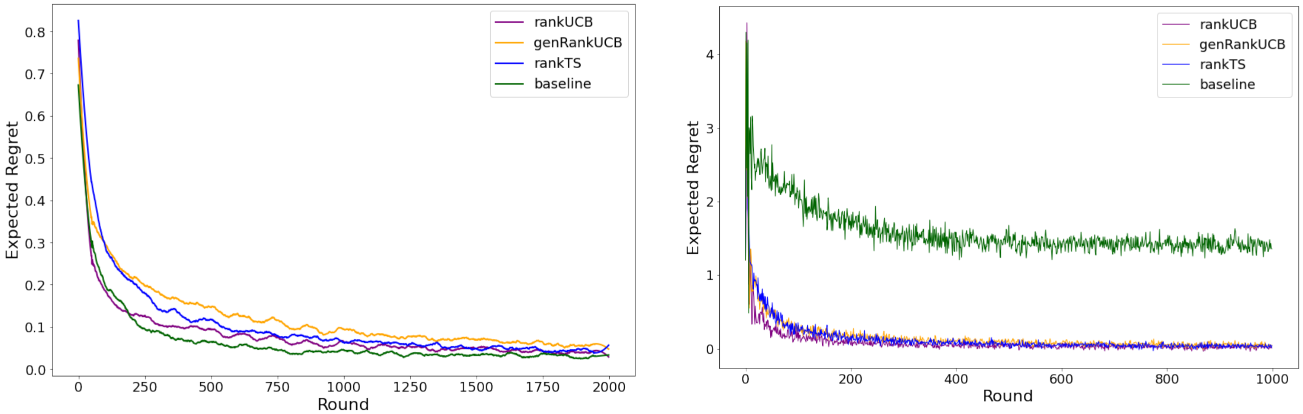


Figure 2: Expected regret for $K = 100$. **Left:** $w_l = 0 \;\; \forall l \in [L]$, **Right:** $\max_{l\in[L]} |w_l| = 10$.

Algorithm 1: RankUCB

```
Input: $\lambda > 0$, $\delta \in (0, 1)$, $L$, $\{w_l\}_{l \le L}$, $T$, arm set $A = \{1, \dots, K\}$, and vector $v_0$
Create $L$-layered graph $G = \bigcup_{i=1}^{K} G_i$ over super-arms of set $A$
Initialization: $\hat{\theta}_0^l = 0$, $V_0^l = \lambda I$ for $l \in [L]$, and for any edge $e$ of $G$, set $\hat{c}_e = 0$
for $t = 1, 2, \dots, T$ do
    Obtain $p_i \leftarrow \mathrm{ShortestPathAlgorithm}(-G_i)$ for all $i \in [K]$ simultaneously
    $p_\star \leftarrow \mathrm{argmin}_{p_i} \sum_{e \in p_i} \hat{c}_e$
    Choose action $a_t$ as the ordered vertices of path $p_\star$
    Play $a_t$ and observe $r_{a_t}^l$ for $l \in [L]$
    for $l = 1, \dots L$ do
        $V_t^l(\lambda) \leftarrow V_{t-1}^l + (v_{a_t^l} + w_l v_{a_t^{l-1}})(v_{a_t^l} + w_l v_{a_t^{l-1}})^{\mathrm{T}}$
        Create $\mathcal{C}_{t+1}^l$ based on Equation 5
        $\mathrm{UCB}_{t+1}^l(i, j) \leftarrow \max_{\theta \in \mathcal{C}_{t+1}^l} f^l \langle (\theta, v_j + w_l v_i) \rangle$ for all super-arms $(i, j)$
        Update $\hat{c}_e$, for any edge $e$, based on Equation 7
    end for
end for
```

Assumption 1. *For some $m_1, m_2 > 0$, the following hold: (a) for any arm $i \in A$, $\|v_i\|_2 \le m_1$, (b) for all $l \in L$, $\|\theta^l\|_2 \le m_2$, (c) $\sup_{l\in[L]} \sup_{a \in \mathbb{R}^d} |f^l(\langle \theta^l \ , \ a \rangle)| \le 1$, (d) There exist $\delta \in (0, 1)$ such that with probability at least $1 - \delta$, for all $t \in [T]$ and $l \in [L]$, $\theta^l \in \mathcal{C}_t^l$ where $\mathcal{C}_t^l$ satisfies the Equation 5.*

Assumption 2. *We denote the derivative of f^l by $\dot{f}^l$. There exist $c_1 > 0$ and $c_2 < \infty$ such that:*

$$c_1 = \min\{1, \min_{l\in[L]} \min_{a \in \mathbb{R}^d} \min_{\theta^l \in \Theta} \dot{f}^l(\langle \theta^l \ , \ a \rangle)\}$$

$$c_2 = \max\{1, \max_{l\in[L]} \max_{a \in \mathbb{R}^d} \max_{\theta^l \in \Theta} \dot{f}^l(\langle \theta^l \ , \ a \rangle)\}$$

Theorem 1. *Under Assumptions 1 and 2, with probability at least $1 - \delta$, the expected regret of the RankUCB algorithm satisfies:*

$$\mathcal{R}_T \le \frac{2\sqrt{2}c_2}{c_1} L \sqrt{dT\beta_T \log\left(1 + \frac{T\left((1+\max_{l\in[L]} |w_l|)m_1\right)^2}{d\lambda}\right)} \tag{8}$$

where

$$\sqrt{\beta_T} = \max_{l\in[L]} \sqrt{\lambda} m_2 + \sqrt{2\log\left(\frac{1}{\delta}\right) + \log\left(\frac{\det\left(V_T^l(\lambda)\right)}{\lambda^d}\right)}.$$

Theorem 1 (see Appendix A.2 for proof) provides an upper bound of $\tilde{O}(L\sqrt{dT})$ for the ranking MAB problem. The notation $\tilde{O}$ drops the logarithmic complexity. This upper bound increases linearly on L while capturing both item and position dependencies compared to previous works, where these dependencies were either ignored or simplified.

5 Experiments

In this section, we compare RankUCB, genRankUCB (Appendix B), and RankTS (Appendix C) to the baseline algorithm (Radlinski, Kleinberg, and Joachims 2008), where there is no assumption on the dependency between positions. To make the comparison fair to the baseline, we implemented the linear case. The experiments are contextual bandits with $d = 10$, $L = 4$ and $K \in \{10, 100\}$, and various values for weights w_l to be small, large, and zero. The case where $w_l = 0$ for all $l \in [L]$ would be similar to the setting discussed in (Radlinski, Kleinberg, and Joachims 2008). Regarding the parameters, we randomly choose $\theta_l' \in \mathbb{R}^{d-1}$ with $\|\theta_l'\|_2 = 1$ and let $\theta^l = (\frac{\theta_l'}{2} \ , \ \frac{1}{2})$. We let the vector associated to arm i be $v_i = (v_i' \ , \ 1)$, where $v_i' \in \mathbb{R}^{d-1}$ with $\|v_i'\|_2 = 1$. This process will guarantee that $\sup_{l\in[L]} \sup_{i \in A} |\langle \theta^l \ , \ v_i \rangle| \le 1$, which is required for assumptions. Next, we generate the weight w_l by a random sample from the uniform distribution.

The results are reported in Figure 2 and Appendix D. When $\max_{l\in[L]} |w_l|$ is very small or zero, Figure 2-left, all the algorithms perform well. As $\max_{l\in[L]} |w_l|$ increases, the baseline algorithm, which does not capture the position dependencies, does not converge to the optimal action. This leads to a non-zero regret over time T. In contrast, our algorithms perform well with all ranges of w_l.

6 Conclusion

We studied the ranking problem in multi-armed bandits with position-item dependency with generalized linear reward functions. We proposed two algorithms, RankUCB and RankTS, where the key idea is to formulate the optimal ordered list as the longest path in a graph. The experiments show the advantage of involving position dependency. We hope this work motivates the community to gather temporal data to improve the ranking in recommendation systems.

Acknowledgement

Arnaud Doucet is partly supported by the EPSRC grant EP/R034710/1. He also acknowledges the support of the UK Defence Science and Technology Laboratory (DSTL) and EPSRC under grant EP/R013616/1. This is part of the collaboration between US DOD, UK MOD, and UK EPSRC, under the Multidisciplinary University Research Initiative.

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems*, 24.

Abeille, M.; and Lazaric, A. 2017. Linear Thompson sampling revisited. In *Artificial Intelligence and Statistics*, 176–184. PMLR.

Agrawal, S.; and Goyal, N. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, 127–135. PMLR.

Andrieu, C.; De Freitas, N.; Doucet, A.; and Jordan, M. I. 2003. An introduction to MCMC for machine learning. *Machine Learning*, 50(1): 5–43.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2): 235–256.

Buccapatnam, S.; Eryilmaz, A.; and Shroff, N. B. 2014. Stochastic Bandits with Side Observations on Networks. In *The 2014 ACM international conference on Measurement and modeling of computer systems*, 289–300.

Chuklin, A.; Markov, I.; and de Rijke, M. 2015. Click models for web search. *Synthesis lectures on information concepts, retrieval, and services*, 7(3).

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; and Stein, C. 2022. *Introduction to Algorithms*. MIT press.

Ding, Q.; Hsieh, C.-J.; and Sharpnack, J. 2021. An efficient algorithm for generalized linear bandit: Online stochastic gradient descent and thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, 1585–1593. PMLR.

Ermis, B.; Ernst, P.; Stein, Y.; and Zappella, G. 2020. Learning to rank in the position based model with bandit feedback. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2405–2412.

Filippi, S.; Cappe, O.; Garivier, A.; and Szepesvári, C. 2010. Parametric Bandits: The Generalized Linear Case. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

Gauthier, C.-S.; Gaudel, R.; and Fromont, E. 2022. UniRank: Unimodal Bandit Algorithms for Online Ranking. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*. PMLR.

Gittins, J. C. 1979. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2): 148–164.

Gupta, S.; Chaudhari, S.; Joshi, G.; and Yagan, O. 2021. Multi-Armed Bandits With Correlated Arms. *IEEE Transactions on Information Theory*, 67(10).

Kim, W.; Lee, K.; and Paik, M. C. 2022. Double Doubly Robust Thompson Sampling for Generalized Linear Contextual Bandits. *arXiv preprint arXiv:2209.06983*.

Lagrée, P.; Vernade, C.; and Cappe, O. 2016. Multiple-play bandits in the position-based model. *Advances in Neural Information Processing Systems*, 29.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit Algorithms*. Cambridge University Press.

Li, S.; Wang, B.; Zhang, S.; and Chen, W. 2016. Contextual Combinatorial Cascading Bandits. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*. PMLR.

Liu, C.-Y.; and Li, L. 2016. On the Prior Sensitivity of Thompson Sampling. In *Algorithmic Learning Theory*, 321–336. Springer International Publishing.

Lykouris, T.; Tardos, E.; and Wali, D. 2020. Feedback graph regret bounds for Thompson Sampling and UCB. In *Algorithmic Learning Theory*, 592–614. PMLR.

Mannor, S.; and Shamir, O. 2011. From bandits to experts: On the value of side-observations. *Advances in Neural Information Processing Systems*, 24.

McCullagh, P.; and Nelder, J. 1989. *Generalized Linear Models, Second Edition*. Chapman and Hall/CRC Monographs on Statistics and Applied Probability Series. Chapman & Hall.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions–I. *Mathematical programming*, 14(1).

Radlinski, F.; Kleinberg, R.; and Joachims, T. 2008. Learning diverse rankings with multi-armed bandits. In *International Conference on Machine Learning*, 784–791.

Russo, D. J.; Van Roy, B.; Kazerouni, A.; Osband, I.; and Wen, Z. 2018. A Tutorial on Thompson Sampling. *Foundations and Trends in Machine Learning*, 11(1).

Sedgewick, R.; and Wayne, K. 2011. *Algorithms, 4th Edition.* Addison-Wesley.

Singh, R.; Liu, F.; Liu, X.; and Shroff, N. 2020. Contextual bandits with side-observations. *arXiv preprint arXiv:2006.03951*.

Slivkins, A.; Radlinski, F.; and Gollapudi, S. 2013. Ranked bandits in metric spaces: learning diverse rankings over large document collections. *Journal of Machine Learning Research*, 14(Feb): 399–436.

Sniedovich, M. 2006. Dijkstra's algorithm revisited: the dynamic programming connexion. *Control and Cybernetics*, 35(3): 599–620.

Thompson, W. R. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4): 285–294.

Wainwright, M. J.; and Jordan, M. I. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2): 1–305.

A RankUCB Proofs

A.1 Proof of Lemma 1

Here, we give the proof of Lemma 1.

Proof. It is enough to show that $L_t^l(\theta^l) \leq \sqrt(\beta_t^l)$. To do so, we use Equations 3 and 4, and rewrite $L_t^l(\theta)$ as follows:

$$\begin{aligned} L_t^l(\theta^l) &= \|\lambda\theta^l + \sum_{s=1}^{t} \Big[f(\langle \theta^l \ , \ v_{a_s^l} + w_l v_{a_s^{l-1}} \rangle) - f(\langle \theta^l \ , \ v_{a_s^l} + w_l v_{a_s^{l-1}} \rangle) - \eta_s^l \Big] (v_{a_s^l} + w_l v_{a_s^{l-1}}) \|_{V_t^{l-1}} \\ &= \|\lambda\theta^l - \sum_{s=1}^{t} \eta_s^l (v_{a_s^l} + w_l v_{a_s^{l-1}}) \|_{V_t^{l-1}} \\ &\leq \sqrt{\lambda} \|\theta^l\|_2 + \| \sum_{s=1}^{t} \eta_s^l (v_{a_s^l} + w_l v_{a_s^{l-1}}) \|_{V_t^{l-1}} \end{aligned}$$

Now, we can use the standard self-normalized bound for vector-valued martingales proposed in (Abbasi-Yadkori, Pál, and Szepesvári 2011), see Lemma 8 and 9 for details, to bound the second term in the right hand side. The proof follows the corresponding lemmas by replacing $X_s = v_{a_s^l} + w_l v_{a_s^{l-1}}$ for any time s. □

A.2 Proof of Theorem 1

In order to provide the proof for Theorem 1, we first need the following lemmas, which often the second one is called the elliptical potential lemma:

Lemma 2. *For any $\theta, \theta' \in \Theta$ and $l \in [L]$, we have $c_1 \|\theta - \theta'\|_{V_t^l} \leq \|g_t^l(\theta) - g_t^l(\theta')\|_{V_t^{l-1}}$.*

Proof. Since f is continuous and differentiable, $g_t^l(\theta)$ is also continuous and differentiable. By Mean Value Theorem, we have that there exist $\theta^\star$, such that:

$$g_t^l(\theta) - g_t^l(\theta') = \dot{g}_t^l(\theta^\star)(\theta - \theta')$$

Therefore, we have:

$$g_t^l(\theta) - g_t^l(\theta') = \underbrace{(\lambda I + \sum_{s=1}^{t} \dot{f}(\langle \theta^\star \ , \ v_{a_s^l} + w_l v_{a_s^{l-1}} \rangle)(v_{a_s^l} + w_l v_{a_s^{l-1}})(v_{a_s^l} + w_l v_{a_s^{l-1}})^{\mathrm{T}})}_{M_t^l}(\theta - \theta')$$

Since $M_t^l \succeq c_1 V_t^l$, then we have:

$$\|g_t^l(\theta) - g_t^l(\theta')\|_{V_t^{l-1}} = \|\theta - \theta'\|_{M_t^l V_t^{l-1} M_t^l} \geq c_1 \|\theta - \theta'\|_{V_t^l}$$

And the proof is complete. □

Lemma 3. *Let $V_0 \in \mathbb{R}^{d \times d}$ be a positive definite matrix and $b_1, \ldots, b_T \in \mathbb{R}^d$ be a sequence of vectors with $\|b_t\|_2 \leq M < \infty$. For all $t \in [T]$, define $V_t = V_0 + \sum_{s \leq t} b_s b_s^{\mathrm{T}}$. Then,*

$$\sum_{t=1}^{T} \min\{1 \ , \ \|b_t\|_{V_t^{-1}}^2\} \leq 2 \log\left(\frac{\det(V_T)}{\det(V_0)}\right) \leq 2d \log\left(\frac{\mathrm{tr}(V_0) + TM^2}{d \det(V_0)^{\frac{1}{d}}}\right).$$

Proof. If V is a symmetric positive definite matrix, then $V + U = V^{1/2}(I + V^{-1/2}UV^{-1/2})V^{1/2}$. Moreover, for each $t \in [T]$, we have that V_t is a symmetric positive definite matrix. Thus, for any $t \geq 1$, we can write:

$$V_t = V_{t-1} + b_t b_t^{\mathrm{T}} = V_{t-1}^{1/2} \left(I + V_{t-1}^{-1/2} b_t b_t^{\mathrm{T}} V_{t-1}^{-1/2} \right) V_{t-1}^{-1/2}$$

By noting that $\det(VU) = \det(V)\det(U)$, we have that:

$$\det(V_t) = \det(V_{t-1}) \det\left(I + V_{t-1}^{-1/2} b_t b_t^{\mathrm{T}} V_{t-1}^{-1/2} \right) = \det(V_{t-1}) \left(1 + \|b_t\|_{V_{t-1}^{-1}}^2 \right).$$

The last equality is due to the fact that the determinant of a matrix is the product of its eigenvalues, and matrix $I + xx^{\mathrm{T}}$ has eigenvalues $1 + \|x\|_2^2$ and 1. By repeatedly applying this equality, we have that:

$$\det(V_t) = \det(V_0) \prod_{s=1}^{t} \left(1 + \|b_s\|_{V_{s-1}^{-1}}^2 \right).$$

Therefore, we obtain

$$\frac{\det(V_t)}{\det(V_0)} = \prod_{s=1}^{t} \left(1 + \|b_s\|^2_{V_{s-1}^{-1}}\right). \tag{9}$$

Now, using Equation 9 and the fact that for any $x \geq 0$, $\min\{1, x\} \leq 2\log(1+x)$, we get the following:

$$\sum_{t=1}^{T} \min\left\{1 \ , \ \|b_t\|^2_{V_t^{-1}}\right\} \leq 2\sum_{t=1}^{T} \log\left(1 + \|b_t\|^2_{V_t^{-1}}\right) = 2\log\left(\frac{\det(V_t)}{\det(V_0)}\right).$$

This proves the first inequality in the lemma. For the second inequality, we use the inequality of arithmetic and geometric means. So, we have that:

$$\det(V_T) = \prod_{i=1}^{d} \lambda_i \leq \left(\frac{1}{d}\sum_{i=1}^{d} \lambda_i\right)^d = \left(\frac{1}{d}\mathrm{tr}(V_T)\right)^d \leq \left(\frac{\mathrm{tr}(V_0) + TM^2}{d}\right)^d,$$

where $\lambda_1, \ldots, \lambda_d$ denote the eigenvalues of V_T, and the proof is complete. □

Now, we give the proof of Theorem 1.

Proof. By Assumption 1, it suffices to prove the bound on the event that for all $l \in [L]$, $\theta^l \in \mathcal{C}^l_t$. Let $a_\star = \mathrm{argmax}_{a \in \mathcal{A}} \sum_{l=1}^{L} f^l\left(\langle \theta^l \ , \ v_{a^l} + w_l v_{a^{l-1}} \rangle\right)$, and R_t be the instantaneous total regret in round t. Then,

$$R_t = \sum_{l=1}^{L} f^l\left(\langle \theta^l \ , \ v_{a^l_\star} + w_l v_{a^{l-1}_\star} \rangle\right) - \sum_{l=1}^{L} f^l\left(\langle \theta^l \ , \ v_{a^l_t} + w_l v_{a^{l-1}_t} \rangle\right).$$

For each $l \in [L]$, let $\tilde{\theta}^l_t \in \mathcal{C}^l_t$ be the parameter for which $f^l\left(\langle \tilde{\theta}^l \ , \ v_{a^l_t} + w_l v_{a^{l-1}_t} \rangle\right) = \mathrm{UCB}^l_t(a^{l-1}_t, a^l_t)$. Now, the fact that $\theta^l \in \mathcal{C}^l_t$ and Equation 6 lead us to the following:

$$f^l\left(\langle \theta^l \ , \ v_{a^l_\star} + w_l v_{a^{l-1}_\star} \rangle\right) \leq \mathrm{UCB}^l_t(a^{l-1}_\star, a^l_\star)$$

$$\Longrightarrow \sum_{l=1}^{L} f^l\left(\langle \theta^l \ , \ v_{a^l_\star} + w_l v_{a^{l-1}_\star} \rangle\right) \leq \sum_{l=1}^{L} \mathrm{UCB}^l_t(a^{l-1}_\star, a^l_\star).$$

Note that $a_\star$ corresponds to a path in graph G of Algorithm 1, and since the longest path of graph G at round t has been a_t, we can write:

$$\sum_{l=1}^{L} \mathrm{UCB}^l_t(a^{l-1}_\star, a^l_\star) \leq \sum_{l=1}^{L} \mathrm{UCB}^l_t(a^{l-1}_t, a^l_t) = \sum_{l=1}^{L} f^l\left(\langle \tilde{\theta}^l \ , \ v_{a^l_t} + w_l v_{a^{l-1}_t} \rangle\right).$$

Therefore,

$$\begin{aligned}
R_t &= \sum_{l=1}^{L} f^l\left(\langle \theta^l \ , \ v_{a^l_\star} + w_l v_{a^{l-1}_\star} \rangle\right) - \sum_{l=1}^{L} f^l\left(\langle \theta^l \ , \ v_{a^l_t} + w_l v_{a^{l-1}_t} \rangle\right) \\
&\leq \sum_{l=1}^{L} f^l\left(\langle \tilde{\theta}^l_t \ , \ v_{a^l_t} + w_l v_{a^{l-1}_t} \rangle\right) - \sum_{l=1}^{L} f^l\left(\langle \theta^l \ , \ v_{a^l_t} + w_l v_{a^{l-1}_t} \rangle\right) \\
&\leq \sum_{l=1}^{L} c_2 \langle \tilde{\theta}^l_t - \theta^l \ , \ v_{a^l_t} + w_l v_{a^{l-1}_t} \rangle \\
&\leq \sum_{l=1}^{L} c_2 \|v_{a^l_t} + w_l v_{a^{l-1}_t}\|_{V^{l-1}_{t-1}} \|\tilde{\theta}^l_t - \theta^l\|_{V^l_{t-1}}.
\end{aligned}$$

The last line follows from the Cauchy-Schwartz inequality and the line before that is concluded by Assumption 2. Now, by using Lemma 2 and Equation 4, we have:

$$\begin{aligned} R_t &\le \sum_{l=1}^{L} c_2 \|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}} \|\tilde{\theta}_t^l - \theta^l\|_{V_{t-1}^l} \\ &\le \sum_{l=1}^{L} c_2 \|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}} \frac{\|g_{t-1}^l\left(\tilde{\theta}_t^l\right) - g_{t-1}^l\left(\theta^l\right)\|_{V_{t-1}^l}}{c_1} \\ &\le \sum_{l=1}^{L} \frac{c_2}{c_1} \|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}} \left(L_{t-1}^l(\tilde{\theta}^l) - L_{t-1}^l(\theta^l)\right) \\ &\le \sum_{l=1}^{L} \frac{2c_2}{c_1} \sqrt{\beta_{t-1}^l} \|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}}. \end{aligned}$$

The last inequality follows from Lemma 1. By Assumption 1, we can write that $R_t \le 2L$. Hence,

$$R_t \le \min\left\{2L\ ,\ \sum_{l=1}^{L} \frac{2c_2}{c_1} \sqrt{\beta_{t-1}^l} \|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}}\right\}$$

Using Lemma 1 and Assumption 1, we have that:

$$\sqrt{\beta_{t-1}^l} \le \sqrt{\lambda} m_2 + \sqrt{2\log\left(\frac{1}{\delta}\right) + \log\left(\frac{\det\left(V_{t-1}^l(\lambda)\right)}{\lambda^d}\right)}$$

Thus,

$$R_t \le \frac{2c_2}{c_1} \sum_{l=1}^{L} \left(\sqrt{\lambda} m_2 + \sqrt{2\log\left(\frac{1}{\delta}\right) + \log\left(\frac{\det\left(V_{t-1}^l(\lambda)\right)}{\lambda^d}\right)}\right) \min\left\{1\ ,\ \|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}}\right\} \tag{10}$$

Moreover, the expected regret can be written as $\mathcal{R}_T = \mathbb{E}\left[\sum_{t=1}^{T} R_t\right]$, which can be upper bounded by Equation 10. Also, note that $\left\{\det\left(V_t^l\right)\right\}_{t=0}^{T}$ is an increasing sequence.[1] Therefore, we can upper bound the regret as follows:

$$\begin{aligned} \mathcal{R}_T &= \mathbb{E}\left[\sum_{t=1}^{T} R_t\right] \\ &\le \frac{2c_2}{c_1} \sum_{t=1}^{T} \sum_{l=1}^{L} \sqrt{\beta_T} \min\left\{1\ ,\ \|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}}\right\} \\ &\le \frac{2c_2}{c_1} \sqrt{LT \sum_{l=1}^{L} \sum_{t=1}^{T} \beta_T \min\left\{1\ ,\ \|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}}^2\right\}}. \end{aligned}$$

The last inequality follows from Cauchy–Schwartz inequality. Now, we can use Lemma 3 to upper bound $\sum_{t=1}^{T} \min\left\{1\ ,\ \|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}}^2\right\}$. One can check that if we define variable $b_t = v_{a_t^l} + w_l v_{a_t^{l-1}}$, and variable $M = (1 + \max_{l\in[L]} |w_l|) m_1$, then we can write:

$$\mathcal{R}_T \le \frac{2c_2}{c_1} \sqrt{LT\beta_T \sum_{l=1}^{L} 2d \log\left(\frac{\mathrm{tr}(V_0^l) + T\left((1 + \max_{l\in[L]} |w_l|) m_1\right)^2}{d \det\left(V_0^l\right)^{\frac{1}{d}}}\right)}.$$

By replacing $\mathrm{tr}(V_0^l) = d\lambda$ and $\det(V_0^l) = \lambda^d$, we get the following bound:

$$\mathcal{R}_T \le 2\sqrt{2} \frac{c_2}{c_1} L \sqrt{dT\beta_T \log\left(1 + \frac{T\left((1 + \max_{l\in[L]} |w_l|) m_1\right)^2}{d\lambda}\right)}.$$

This completes the proof. □

[1] For more details, see the proof of Lemma 3.

B Generalization of RankUCB: Estimating Position Dependencies

In Section 4, we have assumed that the dependency parameters are known. However, in realistic scenarios, we need to estimate them. We now reformulate the problem in a way that allows us to jointly estimate $\theta = (\theta^1, \dots, \theta^l)$ and $w = (w_1, \dots, w_l)$. Then, we modify the RankUCB algorithm for this general case and provide the corresponding regret bound.

Let us rewrite the expected reward of action $a = (a^1, \dots, a^L)$ at position l as follows:

$$\mathbb{E}\left[r^l_a\right] = f^l\left(\langle \theta^l \ , \ v_{a^l} + w_l v_{a^{l-1}} \rangle\right) = f^l\left(\langle \phi^l \ , \ \tilde{x}^l_a \rangle\right),$$

where $\phi^l = \begin{pmatrix}\theta^l & w_l\theta^l\end{pmatrix}^{\mathrm{T}} \in \mathbb{R}^{2d}$ and $\tilde{x}^l_a = \begin{pmatrix}v_{a^l} & v_{a^{l-1}}\end{pmatrix}^{\mathrm{T}} \in \mathbb{R}^{2d}$. We can follow a similar procedure to that in Section 4. We define the modified required variables as follows:

$$g^l_t(\theta) = \lambda\theta + \sum_{s=1}^{t} f^l(\langle \theta \ , \ x^l_s \rangle) x^l_s,$$

$$L^l_t(\theta) = \|g^l_t(\theta) - \sum_{s=1}^{t} r^l_s x^l_s\|_{V_t^{l^{-1}}},$$

where $\tilde{V}^l_0(\lambda) = \lambda I \in \mathbb{R}^{2d\times 2d}$, and $\tilde{V}^l_t(\lambda) = \tilde{V}^l_0(\lambda) + \sum_{s=1}^{t} \tilde{x}^l_{a_s} \tilde{x}^{l^{\mathrm{T}}}_{a_s}$. Now, we can use Lemma 1 to create a confidence interval for ϕ^l denoted by $\tilde{\mathcal{C}}^l_t$. Thus, the estimated reward for super-arm (i, j) at position l, which is denoted by vector $\tilde{x}_{ji} = \begin{pmatrix}v_j & v_i\end{pmatrix}^{\mathrm{T}}$, would be

$$\mathrm{UCB}^l_t(i, j) = \max_{\phi \in \tilde{\mathcal{C}}^l_t} f^l\left(\langle \phi \ , \ \tilde{x}_{ji} \rangle\right). \tag{11}$$

Finally, to find the best-ordered list at each round, we can build the L-layered graph G as before and use the Equations 11 and 7 to update the weights of the edges. The generalized algorithm, genRankUCB, is provided in Algorithm 2. The next theorem upper bounds the regret of this algorithm. First, we need the following assumption:

Algorithm 2: genRankUCB

```
Input: $\lambda > 0$, $\delta \in (0, 1)$, $L$, $T$, arm set $A = \{1, \dots, K\}$, and vector $v_0$
Create $L$-layered graph $G = \bigcup_{i=1}^{K} G_i$ over super-arms of set $A$
Initialization: $\hat{\phi}^l_0 = 0$, $\tilde{V}^l_0 = \lambda I$ for $l \in [L]$, and for any edge $e$ of $G$, set $\hat{c}_e = 0$
for $t = 1, 2, \dots, T$ do
    Obtain $p_i \leftarrow \mathrm{ShortestPathAlgorithm}(-G_i)$ for all $i \in [K]$ simultaneously
    $p_\star \leftarrow \mathrm{argmin}_{p_i} \sum_{e \in p_i} \hat{c}_e$
    Choose action $a_t$ as the ordered vertices of path $p_\star$
    Play $a_t$ and observe $r^l_{a_t}$ for $l \in [L]$
    for $l = 1, \dots L$ do
        $\tilde{V}^l_t(\lambda) \leftarrow \tilde{V}^l_{t-1} + \tilde{x}^l_{a_s} \tilde{x}^{l^{\mathrm{T}}}_{a_s}$
        Create $\tilde{\mathcal{C}}^l_{t+1}$ based on Lemma 1
        $\mathrm{UCB}^l_{t+1}(i, j) \leftarrow \max_{\phi \in \tilde{\mathcal{C}}^l_{t+1}} f^l\left(\langle \phi \ , \ \tilde{x}_{ji} \rangle\right)$ for all super-arms $(i, j)$
        Update $\hat{c}_e$, for any edge $e$, based on Equation 7
    end for
end for
```

Assumption 3. *For some $m_1, m_2, m_3 > 0$, the following hold: (a) for any arm $i \in A$, $\|v_i\|_2 \leq m_1$, (b) for all $l \in L$, $\|\theta^l\|_2 \leq m_2$, (c) for all $l \in L$, $|w_l| \leq m_3$, (d) $\sup_{l \in [L]} \sup_{a \in \mathbb{R}^d} \left|f^l\left(\langle \theta^l \ , \ a \rangle\right)\right| \leq 1$, (e) There exist $\delta \in (0, 1)$ such that with probability at least $1 - \delta$, for all $t \in [T]$ and $l \in [L]$, $\phi^l \in \tilde{\mathcal{C}}^l_t$ where $\tilde{\mathcal{C}}^l_t$ satisfies the Equation 5 for ϕ^l.*

Theorem 2. *Under the conditions of Assumptions 3 and 2, with probability at least $1-\delta$, the expected regret of the genRankUCB algorithm satisfies:*

$$\mathcal{R}_T \leq 4\frac{c_2}{c_1} L \sqrt{dT\beta_T \log\left(1 + \frac{2Tm_2^2}{d\lambda}\right)} \tag{12}$$

where $\sqrt{\beta_T} = \max_{l \in [L]} \sqrt{\lambda}\left(m_2\sqrt{1 + m_3^2}\right) + \sqrt{2\log\left(\frac{1}{\delta}\right) + \log\left(\frac{\det\left(\tilde{V}^l_T(\lambda)\right)}{\lambda^{2d}}\right)}$.

The upper bound provided in Theorem 2 has a larger coefficient factor and is looser than the bound reported in Theorem 1, which was predictable since there are more unknown parameters.

Proof. The proof is similar to the proof of Theorem 1 in Appendix A.2. By Assumption 3, it is suffices to prove the bound on the event that for all $l \in [L]$, $\phi^l \in \tilde{\mathcal{C}}_t^l$. Let $a_\star = \operatorname{argmax}_{a \in \mathcal{A}} \sum_{l=1}^{L} f^l\left(\langle \phi^l \ , \ x_a^l \rangle\right)$, where $x_a^l = (v_{a^l} \quad v_{a^{l-1}})^{\mathrm{T}}$, and R_t be the instantaneous total regret in round t. Then,

$$R_t = \sum_{l=1}^{L} f^l\left(\langle \phi^l \ , \ x_{a_\star}^l \rangle\right) - \sum_{l=1}^{L} f^l\left(\langle \phi^l \ , \ x_{a_t}^l \rangle\right).$$

For each $l \in [L]$, let $\tilde{\phi}_t^l \in \tilde{\mathcal{C}}_t^l$ be the parameter for which $f^l\left(\langle \tilde{\phi}^l \ , \ x_{a_t}^l \rangle\right) = \mathrm{UCB}_t^l(a_t^{l-1}, a_t^l)$. Now, the fact that $\phi^l \in \tilde{\mathcal{C}}_t^l$ and Equation 11 lead us to the following:

$$f^l\left(\langle \phi^l \ , \ x_{a_\star}^l \rangle\right) \leq \mathrm{UCB}_t^l(a_\star^{l-1}, a_\star^l). \tag{13}$$

Using Equation 13 and the facts that $a_\star$ corresponds to a path in graph G of Algorithm 2, and the longest path of graph G at round t has been a_t, we can write:

$$\sum_{l=1}^{L} f^l\left(\langle \phi^l \ , \ x_{a_\star}^l \rangle\right) \leq \sum_{l=1}^{L} \mathrm{UCB}_t^l(a_\star^{l-1}, a_\star^l) \leq \sum_{l=1}^{L} \mathrm{UCB}_t^l(a_t^{l-1}, a_t^l) = \sum_{l=1}^{L} f^l\left(\langle \tilde{\phi}^l \ , \ x_{a_t}^l \rangle\right).$$

Therefore,

$$\begin{aligned}
R_t &= \sum_{l=1}^{L} f^l\left(\langle \phi^l \ , \ x_{a_\star}^l \rangle\right) - \sum_{l=1}^{L} f^l\left(\langle \phi^l \ , \ x_{a_t}^l \rangle\right) \\
&\leq \sum_{l=1}^{L} f^l\left(\langle \tilde{\phi}_t^l \ , \ v_{a_t}^l \rangle\right) - \sum_{l=1}^{L} f^l\left(\langle \phi^l \ , \ x_{a_t}^l \rangle\right) \\
&\leq \sum_{l=1}^{L} c_2 \langle \tilde{\phi}_t^l - \phi^l \ , \ x_{a_t}^l \rangle \\
&\leq \sum_{l=1}^{L} c_2 \|x_{a_t}^l\|_{\tilde{V}_{t-1}^{l-1}} \|\tilde{\phi}_t^l - \phi^l\|_{\tilde{V}_{t-1}^l} \\
&\leq \sum_{l=1}^{L} c_2 \|x_{a_t}^l\|_{\tilde{V}_{t-1}^{l-1}} \frac{\|g_{t-1}^l\left(\tilde{\phi}_t^l\right) - g_{t-1}^l\left(\phi^l\right)\|_{V_{t-1}^l}}{c_1} \\
&\leq \sum_{l=1}^{L} \frac{c_2}{c_1} \|x_{a_t}^l\|_{V_{t-1}^{l-1}} \left(L_{t-1}^l(\tilde{\phi}^l) - L_{t-1}^l(\phi^l)\right) \\
&\leq \sum_{l=1}^{L} \frac{2c_2}{c_1} \sqrt{\beta_{t-1}^l} \|x_{a_t}^l\|_{V_{t-1}^{l-1}} \\
&\leq \min\left\{2L \ , \ \sum_{l=1}^{L} \frac{2c_2}{c_1} \sqrt{\beta_{t-1}^l} \|x_{a_t}^l\|_{V_{t-1}^{l-1}}\right\}.
\end{aligned}$$

The lines are followed by the Cauchy-Schwartz inequality, Lemmas 1 and 2, and the last line is due to Assumption 3, which bounds $R_t \leq 2L$. Now, using Lemma 1 and Assumption 3, we have that:

$$\sqrt{\beta_{t-1}^l} \leq \sqrt{\lambda}\left(m_2\sqrt{1+m_3^2}\right) + \sqrt{2\log\left(\frac{1}{\delta}\right) + \log\left(\frac{\det\left(\tilde{V}_{t-1}^l(\lambda)\right)}{\lambda^{2d}}\right)}.$$

Thus,

$$R_t \leq 2\frac{c_2}{c_1} \sum_{l=1}^{L} \left(\sqrt{\lambda}\left(m_2\sqrt{1+m_3^2}\right) + \sqrt{2\log\left(\frac{1}{\delta}\right) + \log\left(\frac{\det\left(\tilde{V}_{t-1}^l(\lambda)\right)}{\lambda^{2d}}\right)}\right) \min\left\{1 \ , \ \|x_{a_t}^l\|_{\tilde{V}_{t-1}^{l-1}}\right\}. \tag{14}$$

Now, we can upper bound the expected regret $\mathcal{R}_T = \mathbb{E}\left[\sum_{t=1}^{T} R_t\right]$ by Equation 14. Noting that $\left\{\det\left(\tilde{V}_t^l\right)\right\}_{t=1}^{T}$ is an increasing sequence, we can write:

$$\begin{aligned}\mathcal{R}_T &= \mathbb{E}\left[\sum_{t=1}^{T} R_t\right] \\ &\leq 2\frac{c_2}{c_1}\sum_{t=1}^{T}\sum_{l=1}^{L}\sqrt{\beta_T}\min\left\{1\ ,\ \|x^l_{a_t}\|_{\tilde{V}^{l-1}_{t-1}}\right\} \\ &\leq 2\frac{c_2}{c_1}\sqrt{LT\sum_{l=1}^{L}\sum_{t=1}^{T}\beta_T\min\left\{1\ ,\ \|x^l_{a_t}\|^2_{\tilde{V}^{l-1}_{t-1}}\right\}}.\end{aligned}$$

The last claim follows from Cauchy–Schwartz inequality. Using Lemma 3 and defining $b_t = x^l_{a_t} = (v_{a^l_t} \quad v_{a^{l-1}_t})^{\mathrm{T}}$, and $M = 2m_2$, we can write:

$$\mathcal{R}_T \leq 2\frac{c_2}{c_1}\sqrt{LT\beta_T\sum_{l=1}^{L}4d\log\left(\frac{\mathrm{tr}(\tilde{V}^l_0) + 4Tm_2^2}{2d\det\left(\tilde{V}^l_0\right)^{\frac{1}{2d}}}\right)}.$$

By replacing $\mathrm{tr}(\tilde{V}^l_0) = 2d\lambda$ and $\det(\tilde{V}^l_0) = \lambda^{2d}$, we get the following bound:

$$\mathcal{R}_T \leq 4\frac{c_2}{c_1}L\sqrt{dT\beta_T\log\left(1 + \frac{2Tm_2^2}{d\lambda}\right)}.$$

This completes the proof. □

C Ranking Thompson Sampling Algorithm

Thompson Sampling (TS) (Thompson 1933) assumes there exists a prior distribution $\mathcal{Q}$ on the parameter $\theta \in \Theta$ of the conditional reward distribution $\mathcal{P}(\cdot|\theta)$. At each round t, the algorithm draws a sample from the posterior distribution $\hat{\theta}_t \sim \mathcal{Q}(\cdot|\mathcal{H}_t)$, selects the best action according to the sample, and updates the distribution based on the observed reward. However, the computation of the posterior becomes complicated when the conjugacy condition does not apply to these distributions, namely when the reward distribution is not conjugate to the distribution over θ. Recent papers (Ding, Hsieh, and Sharpnack 2021; Kim, Lee, and Paik 2022) have attempted to address this issue using different techniques. Nevertheless, the posterior distribution might be difficult or expensive to sample, even in the conjugate scenario.

This section will present an overview of the influence of the L-layered graph on the linear case to avoid the computation complexity caused by conjugacy. We assume that each θ^l is sampled independently from a prior distribution $\mathcal{Q}^l$, and we will update their posterior distributions separately. The prior distribution $\mathcal{Q}^l$ for different l can be different, i.e., the samples are not necessarily identically distributed. Also, note that for finding the best action according to the samples $\hat{\theta}^l_t$ for $l \in [L]$, we use the L-layering graph technique. In other words, we use the samples of the vector $\hat{\theta} = (\hat{\theta}^1, \ldots, \hat{\theta}^L)$ to estimate the weights of each edge e in the L-layered graph G over super-arms of set A, and find the longest path in the graph as the best action for round t. Thus, the estimated weight of $\hat{c}_e$, where e is the edge from u^l_{ij} to u^{l+1}_{jq} would be defined as follows:

$$\hat{c}_e = \begin{cases} \begin{aligned}&\tfrac{1}{2}(2\langle\hat{\theta}^1\ ,\ v_j + w_1 v_0\rangle \\ &\quad + \langle\hat{\theta}^2\ ,\ v_q + w_2 v_j\rangle)\end{aligned} & \text{if } l = 1; \\ \begin{aligned}&\tfrac{1}{2}(\langle\hat{\theta}^{L-1}\ ,\ v_j + w_{L-1} v_i\rangle \\ &\quad + 2\langle\hat{\theta}^L\ ,\ v_q + w_L v_j\rangle)\end{aligned} & \text{if } l = L-1; \\ \begin{aligned}&\tfrac{1}{2}(\langle\hat{\theta}^l\ ,\ v_j + w_l v_i\rangle \\ &\quad + \langle\hat{\theta}^{l+1}\ ,\ v_q + w_{l+1} v_j\rangle)\end{aligned} & \text{otherwise.} \end{cases} \tag{15}$$

The final adaptation of TS algorithm, RankTS, is described in Algorithm 3.

The first result providing an upper bound for TS with linear reward functions was obtained in (Agrawal and Goyal 2013). Then, (Abeille and Lazaric 2017) presented a new proof, which can also be applied to generalized or regularized linear models. Our upper bound for RankTS borrows the techniques from these two papers. We first need the following assumption to state the main theorem:

Assumption 4. *For some $m_1, m_2 > 0$, the following hold: (a) for any arm $i \in A$, $\|v_i\|_2 \leq m_1$, (b) for all $l \in L$, $\|\theta^l\|_2 \leq m_2$ with $\mathcal{Q}^l$-probability one, (c) $\sup_{l\in[L]}\sup_{a\in\mathbb{R}^d}|\langle\theta^l\ ,\ a\rangle| \leq 1$.*

Algorithm 3: RankTS

```
**Input:** $L$, prior distributions $\{\mathcal{Q}^l\}_{l=1}^L$, $\{w_l\}_{l\leq L}$, $T$, arm set $A = \{1, \dots, K\}$, and vector $v_0$
Create $L$-layered graph $G = \bigcup_{i=1}^K G_i$ over super-arms of set $A$
Initialization: For any edge $e$ of $G$, set $\hat{c}_e = 0$
**for** $t = 1, 2, \dots, T$ **do**
    $(\hat{\theta}^1, \dots, \hat{\theta}^L) \sim \mathcal{Q}^1(\cdot|\mathcal{H}_t) \otimes \dots \otimes \mathcal{Q}^L(\cdot|\mathcal{H}_t)$
    Update $\hat{c}_e$, for any edge $e$, based on Equation 15
    Obtain $p_i \leftarrow \mathrm{ShortestPathAlgorithm}(-G_i)$ for all $i \in [K]$ simultaneously
    $p_\star \leftarrow \mathrm{argmin}_{p_i} \sum_{e\in p_i} \hat{c}_e$
    Choose action $a_t$ as the ordered vertices of path $p_\star$
    Play $a_t$ and observe $r^l_{a_t}$ for $l \in [L]$
    $\mathcal{H}_{t+1} \leftarrow \mathcal{H}_t \cup \{a_t, (r^1_{a_t}, \dots, r^L_{a_t})\}$
    Update $\mathcal{Q}^l(\cdot|\mathcal{H}_{t+1})$ for $l \in L$
**end for**
```

Now, we have the following theorem:

Theorem 3. *Under Assumption 4, the expected regret of the RankTS algorithm is bounded by:*

$$\mathcal{R}_T \leq 2L\big(1 + \sqrt{2Td\beta^2 \log\left(1 + \tfrac{T\left((1+\max_{l\in[L]}|w_l|)m_1\right)^2}{d\lambda}\right)}\big) \tag{16}$$

where

$$\beta = 1 + \sqrt{4\log(T) + d\log\left(1 + \tfrac{T\left((1+\max_{l\in[L]}|w_l|)m_1\right)^2}{d\lambda}\right)}.$$

We need the following corollary of Lemma 3 to prove Theorem 3.

Corollary 1. *Let $V_0 = \lambda I \in \mathbb{R}^{d\times d}$, and $b_1, \dots, b_T \in \mathbb{R}^d$ be a sequence of vectors with $\|b_t\|_2 \leq M < \infty$. For all $t \in [T]$, define $V_t = V_0 + \sum_{s\leq t} b_s b_s^{\mathrm{T}}$. Then,*

$$\frac{\det\left(V_t(\lambda)\right)}{\lambda^d} \leq \left(\mathrm{tr}\left(\frac{V_t(\lambda)}{\lambda d}\right)^d\right) \leq \left(1 + \frac{TM^2}{\lambda d}\right)^2.$$

We can now give the proof of Theorem 3.

Proof. Let us denote the set of the super-arms of set A by $\mathcal{S}(A)$. We start by defining upper confidence bound functions $U^l_t : \mathcal{S}(A) \mapsto \mathbb{R}$ for all $l \in [L]$ as follows:

$$U^l_t(i,j) = \langle \hat{\theta}^l_{t-1}\ ,\ v_j + w_l v_i\rangle + \beta\|v_j + w_l v_i\|_{V^{l-1}_{t-1}}$$

where $V^l_t = \frac{1}{m_2^2}I + \sum_{s=1}^t (v_{a^l_s} + w_l v_{a^{l-1}_s})(v_{a^l_s} + w_l v_{a^{l-1}_s})^{\mathrm{T}}$. By Lemma 1 and Lemma 3, and setting $\lambda = \frac{1}{m_2^2}$ and $\delta = \frac{1}{T^2}$, we have that $\mathbb{P}(\exists t \in [T] : \|\hat{\theta}^l_{t-1} - \theta^l\|_{V^l_{t-1}} > \beta) \leq \frac{1}{T^2}$. Let E^l_t be the event that $\|\hat{\theta}^l_{t-1} - \theta^l\|_{V^l_{t-1}} \leq \beta$, and define $E^l = \bigcap_{t=1}^T E_t$, $E = \bigcap_{l=1}^L E^l$, and $a_\star = \mathrm{argmax}_{a\in\mathcal{A}} \sum_{l=1}^L \langle\theta^l\ ,\ a^l + w_l a^{l-1}\rangle$. Since $\{\theta^l\}_{l=1}^L$ are random, $a_\star$ is a random variable. Now, we can write the regret as follows:

$$\begin{aligned}\mathcal{R}_T &= \mathbb{E}\left[\sum_{t=1}^T\sum_{l=1}^L \langle\theta^l\ ,\ a^l_\star - a^l_t + w_l(a^{l-1}_\star - a^{l-1}_t)\rangle\right] \\ &= \mathbb{E}\left[\mathbf{1}_E \sum_{t=1}^T\sum_{l=1}^L \langle\theta^l\ ,\ a^l_\star - a^l_t + w_l(a^{l-1}_\star - a^{l-1}_t)\rangle\right] \\ &\quad + \mathbb{E}\left[\mathbf{1}_{E^c} \sum_{t=1}^T\sum_{l=1}^L \langle\theta^l\ ,\ a^l_\star - a^l_t + w_l(a^{l-1}_\star - a^{l-1}_t)\rangle\right].\end{aligned} \tag{17}$$

Here $\mathbf{1}_E$ is the indicator function of event E. Now, for the second term which is on the event E^c, we can bound the term inside the expectation based on Assumption 4:

$$
\begin{aligned}
&\mathbb{E}\left[\mathbf{1}_{E^c}\sum_{t=1}^{T}\sum_{l=1}^{L}\langle\theta^l\ ,\ a_\star^l - a_t^l + w_l(a_\star^{l-1} - a_t^{l-1})\rangle\right]\\
&= \sum_{l=1}^{L}\mathbb{E}\left[\mathbf{1}_{E^{l^c}}\sum_{t=1}^{T}\langle\theta^l\ ,\ a_\star^l - a_t^l + w_l(a_\star^{l-1} - a_t^{l-1})\rangle\right]\\
&\le 2T(1+\max_{l\in[l]}|w_l|)\sum_{l=1}^{L}\mathbb{P}(E^{l^c}).
\end{aligned}
$$

The first line is due to the fact that events E^l for any $l \in [L]$ are independent because in Algorithm 3 we have that $(\hat{\theta}^1, \ldots, \hat{\theta}^L) \sim \mathcal{Q}^1(\cdot|\mathcal{H}_t) \otimes \ldots \otimes \mathcal{Q}^L(\cdot|\mathcal{H}_t)$. Now, for $\mathbb{P}(E^{l^c})$ we have that:

$$
\mathbb{P}(E^{l^c}) = \mathbb{P}(\bigcup_{t=1}^{T} E_t^{l^c}) \le \sum_{t=1}^{T}\mathbb{P}(E_t^{l^c}) \le T\frac{1}{T^2} = \frac{1}{T}.
$$

Therefore, the second term of Equation 17 is bounded by $2L(1+\max_{l\in[l]}|w_l|)$. Now, for the first term, we can write:

$$
\begin{aligned}
&\mathbb{E}\left[\mathbf{1}_{E}\sum_{t=1}^{T}\sum_{l=1}^{L}\langle\theta^l\ ,\ a_\star^l - a_t^l + w_l(a_\star^{l-1} - a_t^{l-1})\rangle\right]\\
&\le \mathbb{E}\left[\sum_{t=1}^{T}\sum_{l=1}^{L}\mathbf{1}_{E_t^l}\langle\theta^l\ ,\ a_\star^l - a_t^l + w_l(a_\star^{l-1} - a_t^{l-1})\rangle\right]\\
&= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{l=1}^{L}\mathbb{E}\left[\mathbf{1}_{E_t^l}\langle\theta^l\ ,\ a_\star^l - a_t^l + w_l(a_\star^{l-1} - a_t^{l-1})\rangle|\mathcal{H}_t\right]\right].
\end{aligned}
\tag{18}
$$

To bound this, note that for any $l \in [L]$ both θ^l and $\hat{\theta}_t^l$ are drawn from the same prior, which basically means that $\mathbb{P}(\theta^l \in \cdot|\mathcal{H}_t) = \mathbb{P}(\hat{\theta}_t^l \in \cdot|\mathcal{H}_t)$. Hence, we can conclude that $\mathbb{P}(a_\star = \cdot|\mathcal{H}_t) = \mathbb{P}(a_t = \cdot|\mathcal{H}_t)$ and $\mathbb{E}\left[U_t^l(a_\star^{l-1}, a_\star^l)|\mathcal{H}_t\right] = \mathbb{E}\left[U_t^l(a_t^{l-1}, a_t^l)|\mathcal{H}_t\right]$. Thus,

$$
\begin{aligned}
\mathbb{E}\left[\mathbf{1}_{E_t^l}\langle\theta^l\ ,\ a_\star^l - a_t^l + w_l(a_\star^{l-1} - a_t^{l-1})\rangle|\mathcal{H}_t\right] &= \mathbf{1}_{E_t^l}\mathbb{E}\left[\langle\theta^l\ ,\ v_{a_\star^l} + w_l v_{a_\star^{l-1}}\rangle - U_t^l(a_\star^{l-1}, a_\star^l)\right]\\
&\quad + \mathbf{1}_{E_t^l}\mathbb{E}\left[U_t^l(a_t^{l-1}, a_t^l) - \langle\theta^l\ ,\ v_{a_t^l} + w_l v_{a_t^{l-1}}\rangle\right]\\
&\le \mathbf{1}_{E_t^l}\mathbb{E}\left[U_t^l(a_t^{l-1}, a_t^l) - \langle\theta^l\ ,\ v_{a_t^l} + w_l v_{a_t^{l-1}}\rangle\right]\\
&\le \mathbf{1}_{E_t^l}\mathbb{E}\left[\langle\hat{\theta}_{t-1}^l - \theta^l\ ,\ v_{a_t^l} + w_l v_{a_t^{l-1}}\rangle\right]\\
&\quad + \beta\|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}}\\
&\le \mathbf{1}_{E_t^l}\mathbb{E}\left[\|\hat{\theta}_{t-1}^l - \theta^l\|_{V_{t-1}^l}\|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}}\right]\\
&\quad + \beta\|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}}\\
&\le 2\beta\|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}}.
\end{aligned}
$$

The second line is due to the fact that, by the definition of U_t^l functions, the first term of the first line is negative or zero. Now, we can bound the Equation 18 by noting that according to Assumption 4, $\mathbf{1}_{E_t^l}\langle\theta^l\ ,\ a_\star^l - a_t^l + w_l(a_\star^{l-1} - a_t^{l-1})\rangle \le 2$. Therefore, we have:

$$
\mathbb{E}\left[\mathbf{1}_{E}\sum_{t=1}^{T}\sum_{l=1}^{L}\langle\theta^l\ ,\ a_\star^l - a_t^l + w_l(a_\star^{l-1} - a_t^{l-1})\rangle\right] \le 2\beta\mathbb{E}\left[\sum_{t=1}^{T}\sum_{l=1}^{L}\min\left\{1, \|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}}\right\}\right].
$$

Algorithm	K	ART (ms)
baseline	10	8.43
baseline	100	730.88
RankUCB	10	8.85
RankUCB	100	780.02
RankTS	10	8.02
RankTS	100	729.57
genRankUCB	10	9.30
genRankUCB	100	801.89

Table 1: Average response-time (ART) for $d = 10$, $L = 4$, $T = 1e4$ and 100 runs

Using Cauchy–Schwartz inequality and Lemma 3, we will have:

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{l=1}^{L}\min\left\{1, \|v_{a_t^l} + w_l v_{a_t^{l-1}}\|_{V_{t-1}^{l-1}}\right\}\right] &\leq \sqrt{LT\mathbb{E}\left[\sum_{t=1}^{T}\sum_{l=1}^{L}\min\left\{1, \|v_{a_t^l} + w_l v_{a_t^{l-1}}\|^2_{V_{t-1}^{l-1}}\right\}\right]} \\ &\leq \sqrt{LT\sum_{l=1}^{L} 2d\log\left(1 + \frac{T\left((1+\max_{l\in[L]}|w_l|)m_1\right)^2}{d\lambda}\right)} \\ &= L\sqrt{2Td\log\left(1 + \frac{T\left((1+\max_{l\in[L]}|w_l|)m_1\right)^2}{d\lambda}\right)}.\end{aligned}$$

By substituting all the above bounds to Equation 17, we get the following bound and the proof is complete.

$$\mathcal{R}_T \leq 2L\left(1 + \beta\sqrt{2Td\log\left(1 + \frac{T\left((1+\max_{l\in[L]}|w_l|)m_1\right)^2}{d\lambda}\right)}\right).$$

□

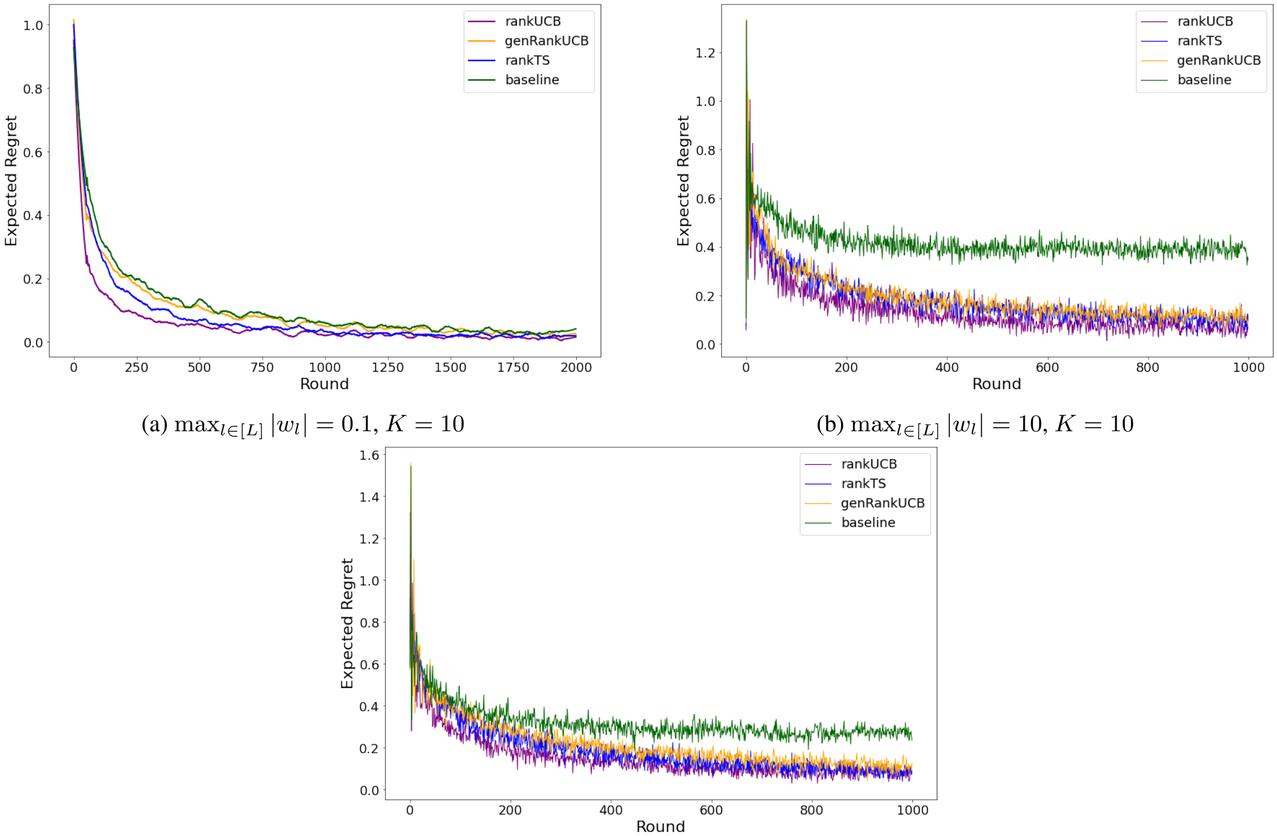
The upper bound obtained for RankTS matches the upper bound obtained by RankUCB, which is consistent with previous results on TS and UCB. As explained earlier, implementation of RankTS needs to sample from the posterior, which is not straightforward for some priors and might need numerical methods such as Markov chain Monte Carlo (Andrieu et al. 2003) or variational inference (Wainwright and Jordan 2008). Having sampled θ^l, finding the best action requires solving a linear optimization problem. By comparison, RankUCB needs to solve 6, which can be intractable for large or continuous action sets.

D More Details On Experiments

We conduct additional experiments to compare the algorithms. In Figure 3, the expected regret for all four algorithms under different initial parameters is shown. As it was discussed in Section 5, all algorithms perform well when $\max_{l\in[L]}|w_l|$ is close to zero. This can be seen in Figures 2 (Left figure) and 3a. However, the baseline algorithm cannot capture the true behavior of the optimal action when $|w_l|$ becomes larger. Even in relatively small $\max_{l\in[L]}|w_l|$ like Figure 3c, the baseline algorithm converges to a non-optimal action. In contrast, the other three algorithms proposed in this paper follow the optimal regret. Note that the regret at each time step t is averaged over 100 runs.

The Figures 2 and 3 are using the multivariate normal distribution as prior in RankTS, i.e. sampling $\hat{\theta}_t^l \sim \mathcal{N}(\mu_{t-1}^l, \Sigma_{t-1}^l)$ for each $\hat{\theta}^l$ separately. We assume the noise is Gaussian as well; therefore, the parameters of the normal distribution for the posterior can easily be updated by the following equations:

$$\begin{aligned}\mu_{t-1}^l &= \Sigma_{t-1}^{l^{-1}}\left[\sum_{s=1}^{t} r_{a_s}^l(v_{a_s}^l + w_l v_{a_s}^{l-1})\right], \\ \Sigma_t^l &= \Sigma_{t-1}^l + (v_{a_t^l} + w_l v_{a_t^{l-1}})(v_{a_t^l} + w_l v_{a_t^{l-1}})^{\mathrm{T}}.\end{aligned}$$



(a) $\max_{l\in[L]} |w_l| = 0.1$, $K = 10$

(b) $\max_{l\in[L]} |w_l| = 10$, $K = 10$

(c) $\max_{l\in[L]} |w_l| = 0.5$, $K = 100$

Figure 3: Expected Regret for $d = 10$, and $L = 4$.

It is noteworthy to mention that Thompson Sampling heavily relies on the prior distribution; a poor prior may prevent an arm from being played enough times, leading to linear regret. A detailed study of prior sensitivity is out of scope of this work[2]. In addition, it can be challenging to find a practical example of this sensitivity.

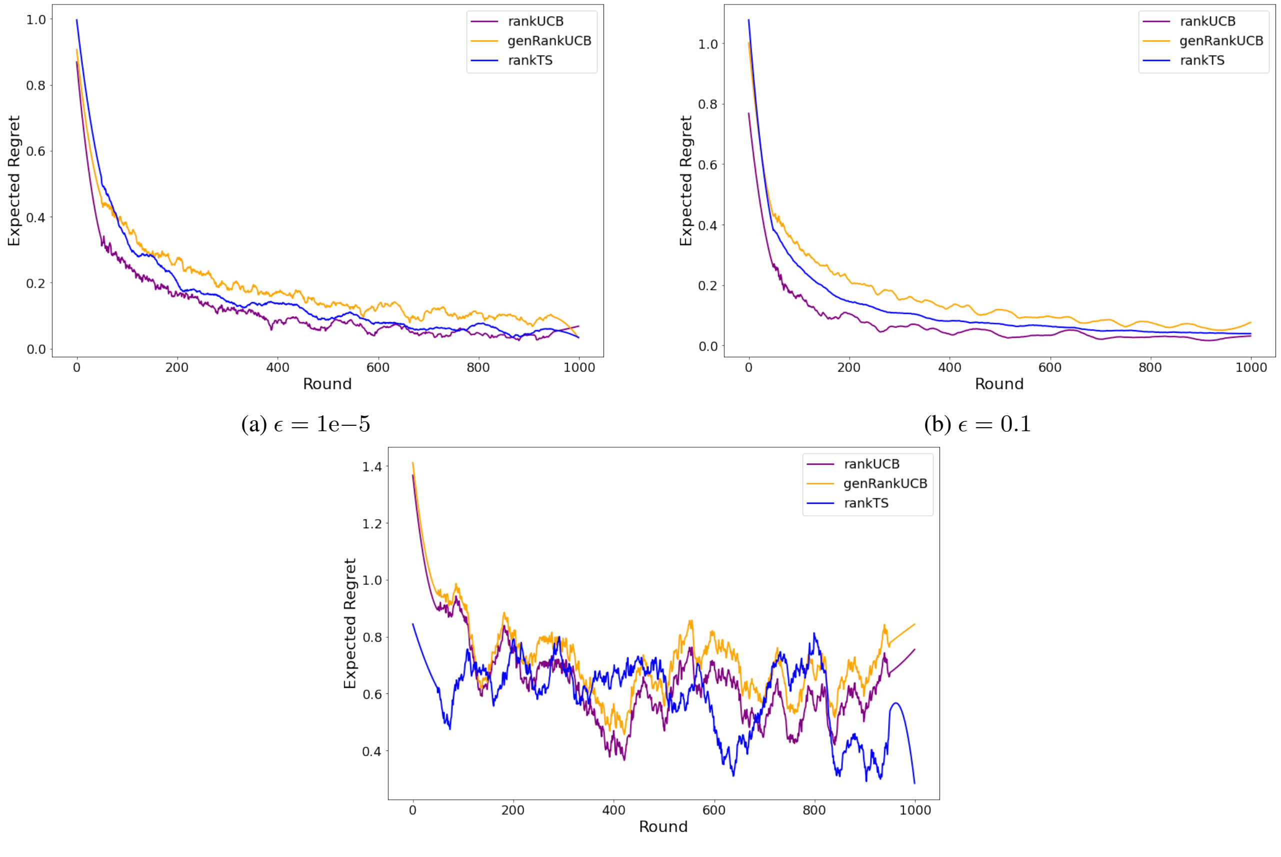
The algorithms are straightforward to implement. Since the majority of computations in UCB and TS are matrix multiplications, they are quite fast. However, when K is high, the shortest path algorithm becomes very slow. This is because shortest path algorithms are not optimized for a specific graph structure, which in our case is the L-layered graph. Changing to shortest path algorithms for sparse graphs, however, may speed up the process. An analysis of the run times of algorithms can be found in Table 1. Moreover, multiprocessing can be used to improve the run time of finding the shortest path to each induced subgraph of G_i, as defined in Section 3. The specifications of the system that generated the data for Table 1 are AMD Ryzen 5 5600x @ 3.7GHz. The importance of contextual bandit algorithms for practical applications such as recommendation systems and online advertising services makes the theoretical and practical investigation of the shortest path optimization problem essential. Applying algorithms with the shortest possible run time can mitigate negative societal impacts in the systems mentioned earlier.

Furthermore, it would be interesting to explore the robustness of these algorithms. Considering a small deviation from the main assumptions, such as non-subgaussianity of the noise, how well the algorithms would perform in finding the optimal action. For this case, we assume that the noise is sampled from a Laplace distribution. In this case, we have the following changes:

$$\begin{aligned} &r_a^l = \langle \theta^l \,,\ v_{a^l} + w_l v_{a^{l-1}} \rangle + \eta^l + \epsilon\tilde{\eta}^l \\ &\text{where } \tilde{\eta}^l \sim \text{Laplace}(0,1) \end{aligned}$$

Here, η^l is a subgaussian noise that matches the main assumptions, and $\tilde{\eta}^l$ is a sample from a zero-mean Laplace distribution with scale 1. Figure 4 shows the results. All algorithms can come close to the optimal action when the scale of perturbation is relatively small. However, a large perturbation scale might result in a potentially non-optimal action, resulting in linear regrets. The interesting point is that adding some perturbations seems to help the algorithms to converge faster, like Figure 4b.

[2] See for instance (Liu and Li 2016).



(a) $\epsilon = 1\mathrm{e}{-5}$

(b) $\epsilon = 0.1$

(c) $\epsilon = 3$

Figure 4: Robustness of Algorithms in Presence of Non-Subgaussian Noise. $\max_{l\in[L]}|w_l| = 1$, and $K = 10$

The codes to reproduce the result are available at https://github.com/shidani/rankingcontextualbandits.

D.1 Dependency on a Window of Previous Items

In this work, we assumed that the reward at position l depends on the attractiveness of both the items at positions l and $l-1$. The result can be generalized to a window of size $S-1$ of the previous items. In other words, we have:

$$\mathbb{E}\left[r_a^l|\mathcal{H}_t\right] = f^l\left(\theta^{l^{\mathrm{T}}}(v_{a^l} + \sum_{i=1}^{S-1} w_{l,i}v_{a^{l-i}})\right).$$

Here, $w_{l,i}$ denotes the dependency of position l to the item shown in position $l-i$. In this case, we need to modify the definition of super-arm to be an S-tuple of items instead of a pair of items. This generalization would change the size of the L-layered graph over super-arms while the structure remains very similar. In more detail, to build a L-layered graph G over the super-arms of S-tuple items, we add K^i vertices to the i-th layer, $1 \le i \le S$, and K^S vertices for the l-th layer, $S < l \le L$. The edges would connect two nodes at layer l to layer $l+1$ if and only if the vertex at layer $l+1$ is left shifted of the vertex at layer l.

Now, we can generalize the previous results. For instance, we will have the following algorithm and theorem based on UCB algorithm:

Theorem 4. *Under Assumption 1, with probability at least $1-\delta$, the expected regret of the winRankUCB algorithm satisfies:*

$$\mathcal{R}_T \le 2\sqrt{2}\frac{c_2}{c_1}L\sqrt{dT\beta_T\log\left(1+\frac{T\left((1+\max_{l\in[L]}\sum_{i=1}^{S-1}|w_{l,i}|)m_1\right)^2}{d\lambda}\right)} \tag{19}$$

where $\sqrt{\beta_T} = \max_{l\in[L]}\sqrt{\lambda}m_2 + \sqrt{2\log\left(\frac{1}{\delta}\right) + \log\left(\frac{\det\left(V_T^l(\lambda)\right)}{\lambda^d}\right)}$.

Algorithm 4: winRankUCB

```
**Input:** $\lambda > 0$, $\delta \in (0,1)$, $L$, $S \geq 1$, $\{w_{l,i}\}_{l \leq L\ ,\ i \leq S-1}$, $T$, arm set $A = \{1, \dots, K\}$, and vector $v_0$
Create $L$-layered graph $G = \bigcup_{i=1}^{K} G_i$ over $S$-tuple super-arms of set $A$
Initialization: $\hat{\theta}_0^l = 0$, $V_0^l = \lambda I$ for $l \in [L]$, and for any edge $e$ of $G$, set $\hat{c}_e = 0$
**for** $t = 1, 2, \dots, T$ **do**
    Obtain $p_i \leftarrow \mathrm{ShortestPathAlgorithm}(-G_i)$ for all $i \in [K]$ simultaneously
    $p_\star \leftarrow \mathrm{argmin}_{p_i} \sum_{e \in p_i} \hat{c_e}$
    Choose action $a_t$ as the ordered vertices of path $p_\star$
    Play $a_t$ and observe $r_{a_t}^l$ for $l \in [L]$
    **for** $l = 1, \dots L$ **do**
        $V_t^l(\lambda) \leftarrow V_{t-1}^l + (v_{a_t^l} + \sum_{i=1}^{S-1} w_{l,i} v_{a_t^{l-i}})(v_{a_t^l} + \sum_{i=1}^{S-1} w_{l,i} v_{a_t^{l-i}})^{\mathrm{T}}$
        Create $\mathcal{C}_{t+1}^l$ based on Equation 5
        $\mathrm{UCB}_{t+1}^l(i_1, \dots, i_{S-1}, j) \leftarrow \max_{\theta \in \mathcal{C}_{t+1}^l} f^l \left( \langle \theta\ ,\ v_j + \sum_{k=1}^{S-1} w_{l,k} v_{i_k} \rangle \right)$ for all super-arms
        Update $\hat{c}_e$, for any edge $e$, based on Equation 7 modified for $S$-tuple nodes of $G$
    **end for**
**end for**
```

The proof is exactly the same as Theorem 1 achieved by replacing the pair of actions with their S-tuple counterparts. However, the size of the L-layered graph would be $O(\frac{K^2}{K-1}(K^S - 1) + (L - S)K^{S+1})$, which directly affects the run-time of the shortest path algorithm. Depending on the computational power, we might be able to solve the shortest path problem for some large value of S.