

---

# Contextual Combinatorial Cascading Bandits

---

## Abstract

We propose the *contextual combinatorial cascading bandits*, an online partial monitoring game where at each time step a learning agent is given a set of contextual information, then selects a list of items, and observes stochastic outcomes of a prefix in the selected items by some stopping criterion. In online recommendation, the stopping criterion might be the first item a user selects; in network routing, the stopping criterion might be the first edge blocked in a path. We consider position discounts in the list order, so that the agent's reward is discounted depending on the position where the stopping criterion is met. We design a UCB-type algorithm,  $C^3$ -UCB, for this problem, prove an  $n$ -step regret bound  $\tilde{O}(\sqrt{n})$  in the general setting, and give finer analysis for two special cases. Our work generalizes existing studies in several directions, including contextual information, position discounts, and a more general cascading bandit model. Experiments on synthetic and real datasets demonstrate the advantage of involving contextual information and position discounts.

## 1. Introduction

Multi-armed bandit (MAB) has been extensively studied in statistics and machine learning. The problem is usually formulated as a system of  $K$  base arms whose rewards are random samples from unknown distributions with unknown means. The learning agent pulls one arm every time and tries to minimize the cumulative regret, which is the difference in cumulative rewards between always pulling the best arm in expectation and playing according to the agent's strategy. The problem of MAB has to deal with the trade-off between exploitation (pulling the best empirical arm) and exploration (trying other arms which are not sufficiently pulled).

Recently, stochastic combinatorial bandit started to draw much attention (Gai et al., 2012; Chen et al., 2013; Lin

et al., 2014; Gopalan et al., 2014; Chen et al., 2014; Kveton et al., 2014; 2015b;a;c; Lin et al., 2015; Combes et al., 2015b). At every time step, a learning agent chooses a subset of ground items (super arm) under certain combinatorial constraints. There are several different kinds of feedback: (1) bandit feedback, where the learning agent can only obtain the reward of the chosen super arm; (2) semi-bandit feedback, where the learning agent can also obtain the stochastic outcomes of the all base arms constituting the chosen super arm; (3) cascading feedback, where the learning agent can obtain the reward of the chosen super arm and the weights of some base arms in the chosen super arm, according to some problem-specific stopping criterion for observation.

Cascading feedback model fits into many real application scenarios. For example, in online or mobile recommendation, it is a typical practice that an ordered list (instead of a single item) is recommended to a user, who usually goes over the list based on the recommended order and selects one of her interest to click through. Thus it is reasonable to assume that items before the clicked item is of no interest to the user while user's interest to items after the clicked item is unclear. Another example is in networking routing, where the agent needs to select a routing path that is least likely of being blocked. To determine whether a path is blocked, the agent checks from the source node until meeting a blocked edge. The edges before the blocked one are unblocked and the edges after the blocked one are unobserved. In both of the above applications, we observe the feedbacks of a prefix of items in the chosen ordered list. The bandit problem with such cascading feedback has been studied in recent papers (Kveton et al., 2015a;c). In this paper, we generalize their work in several directions to cover more realistic scenarios.

First, we incorporate contextual information into cascading bandits. In online recommendation, contextual information includes various user and item information, and user behaviors in different contexts are different. Therefore utilizing contextual information is crucial for personalized recommendation. In network routing, temporal and spacial contextual information may also be useful to determine better routing paths. Therefore, we incorporate contextual information into the cascading bandit formulation. Second, cascading bandits studied in (Kveton et al., 2015a;c) treats all positions in the cascading list equally, but in applications

different positions may bring different rewards. For example, in online recommendation, we usually prefer users to find their interested item in the list as early as possible to increase user satisfaction. In network routing, we may prefer to hit a blocked edge as late as possible. To model these preferences, we introduce position discounts to the cascading bandit, and we show through experiments that incorporating position discounts may significantly improve the learning result. Finally, we generalize the reward functions of (Kveton et al., 2015a;c), which are based on disjunctive and conjunctive binary random variables, to more general non-linear reward functions satisfying monotonicity and Lipschitz continuity conditions. This generalization allows us to cover new realistic scenarios not covered in (Kveton et al., 2015a;c), as exhibited in Section 4.3.

The organization of this paper is as follows: Section 2 compares our work to previous work; Section 3 presents the formulation of *contextual combinatorial cascading bandits* with general reward functions and position discounts; Section 4 gives our algorithm and main results both on general reward functions and two special reward functions; Section 5 demonstrates the experimental results; and Section 6 gives a final conclusion.

## 2. Related Work

We first discuss several studies most relevant to our work. Table 1 summarizes the different settings of these work, which we will explain in details next. A comparison of regret bounds by different methods is deferred to Section 4.3, after we provide full results of our regret bounds.

The work of (Kveton et al., 2015a) introduced the cascading model to the multi-armed bandit framework with disjunctive objective, where the set of feasible actions form a uniform matroid, and the reward of an action is 1 if there is at least one “good” item in the list. The combinatorial cascading bandits of (Kveton et al., 2015c) (referred to as ‘CombCascade’ in Table 1) generalized the framework, allowing each feasible action to be a subset of ground items under combinatorial constraints. It studied a problem of combinatorial bandits with conjunctive objective, where each feasible action is a chain of items, and reward is 1 if all items are “good”. Our work generalizes these models and involves both contextual information and position discounts.

The experiments of (Kveton et al., 2015a) found that recommending items in increasing order of their UCBs (meaning the items with lower preference come early) has a better performance than in decreasing order of their UCBs. This unnatural result is perhaps because the order of items does not affect the reward and thus putting low preference items first and high preference items in the back may result in

	context	cascading	position discount	general reward
CUCB	no	yes	no	yes
C <sup>2</sup> UCB	yes	no	no	yes
CombCascade	no	yes	no	no
C <sup>3</sup> -UCB(ours)	yes	yes	yes	yes

Table 1. Comparisons of our setting with previous ones

more feedbacks in the cascade. The position discounts introduced in this paper makes the recommended set in the decreasing order of UCBs, which is more realistic. The work of (Combes et al., 2015a) considered a similar cascading model to that of (Kveton et al., 2015a) with a particular case of contextual information, where a user is recommended with  $K$  items selected from a user-related group. Under this particular setting, they considered position discounts. In our setting, users and items are represented by general feature vectors.

Our work considers contextual combinatorial bandits with cascading feedback and nonlinear reward. The paper (Li et al., 2010) studied multi-armed bandit with contextual information, which recommends one item a time. A recent work (Qin et al., 2014) (referred to as ‘C<sup>2</sup>UCB’ in Table 1) introduced contextual combinatorial bandit with semi-bandit feedback and nonlinear reward. Their feedback is more informative than ours. We relax the feedback from semi-bandit to cascading, yet our algorithm achieves the same order of regret bound.

There have been several pieces of work on partial monitoring MAB, but the results are not applicable to our setting. The papers of (Agrawal et al., 1989; Bartók et al., 2012) studied partial monitoring problems but their algorithms are inefficient in our setting. The work (Lin et al., 2014) studied combinatorial partial monitoring problem and their feedback is a fixed (with respect to the chosen action) combination of weights. In our setting, if the chosen action is fixed, the number of observed items varies with the user’s behavior, thus the combination constituting feedback is not fixed.

Chen et al. (2015) considered general reward functions and probabilistic triggering of base arms in the stochastic combinatorial semi-bandit model (referred to as ‘CUCB’ in Table 1). Cascading bandits can be viewed as a particular type of probabilistic triggering, where only the head of the list is deterministically triggered and all the remaining items are probabilistically triggered. However, their work does not consider contextual information, and it is also difficult to incorporate position discounts in a general triggering model.

### 3. Problem Formulation

We formulate the problem of *contextual combinatorial cascading bandits* as follows. Suppose we have a finite set  $E = \{1, \dots, L\}$  of  $L$  ground items, also referred to as *base arms*. Let  $\Pi^k = \{(a_1, \dots, a_k) : a_1, \dots, a_k \in E, a_i \neq a_j \text{ for any } i \neq j\}$  be the set of all  $k$ -tuples of distinct items from  $E$ ; we call each of such tuples an *action* of length  $k$ . We will use  $|A|$  to denote the length of an action  $A$ . Let  $\Pi^{\leq K} = \cup_{k=1}^K \Pi^k$  denote the set of all actions with length at most  $K$ , and let  $\mathcal{S} \subseteq \Pi^{\leq K}$  be the set of *feasible actions* with length at most  $K$ .

As a convention, we always use boldface symbols to represent random variables. We also denote  $[m] = \{1, \dots, m\}$ .

At time  $t$ , feature vectors  $x_{t,a} \in \mathbb{R}^{d \times 1}$  with  $\|x_{t,a}\|_2 \leq 1$  for every base arm  $a \in E$  are revealed to the learning agent. Each feature vector combines the information of the user and the corresponding base arm. For example, suppose the user at time  $t$  is characterized by a feature vector  $u_t$  and the base arm  $a$  has a feature vector  $v_a$ , then we can use  $x_{t,a} = u_t v_a^\top$ , the outer-product of  $u_t$  and  $v_a$ , as the combined feature vector of base arm  $a$  at time  $t$ . (See Section 5.2 for an application.) Denote  $x_t = (x_{t,a})_{a \in E}$  as all contextual information at time  $t$ . Then the learning agent recommends a feasible action  $\mathbf{A}_t = (a_1^t, \dots, a_{|\mathbf{A}_t|}^t) \in \mathcal{S}$  to the user. In the cascading feedback models, the user checks from the first item of recommended action and stops at the  $\mathbf{O}_t$ -th item under some stopping criterion. For example, in recommender systems, the stopping criterion might be that the user finds the first attractive item. Then the learning agent observes the weights of first  $\mathbf{O}_t$  base arms in  $\mathbf{A}_t$ , denoted by  $\mathbf{w}_t(a_k^t), k \leq \mathbf{O}_t$ .

Let  $\mathcal{H}_t$  denote the history before the learning agent chooses action at time  $t$ . Thus  $\mathcal{H}_t$  contains feedback information at all time  $s < t$ , as well as contextual information at time  $t$ . Each arm  $a$  has a weight  $\mathbf{w}_t(a)$ , representing the “quality” to the user at time  $t$ . Given history  $\mathcal{H}_t$ , we assume that  $\mathbf{w}_t(a)$ ’s are mutually independent random variables with expectation

$$\mathbb{E}[\mathbf{w}_t(a) | \mathcal{H}_t] = \theta_*^\top x_{t,a}, \quad (1)$$

where  $\theta_*$  is an unknown  $d$ -dimensional vector with the assumption that  $\|\theta_*\|_2 \leq 1$  and  $0 \leq \theta_*^\top x_{t,a} \leq 1$  for all  $t, a$ . Denote by

$$w_{t,a} = \theta_*^\top x_{t,a}$$

the expected weight of base arm  $a$  at time  $t$ . We assume that each  $\mathbf{w}_t(a)$  is a random variable with  $R$ -sub-Gaussian tail, which means that for all  $b \in \mathbb{R}$ ,

$$\mathbb{E}[\exp(b(\mathbf{w}_t(a) - \theta_*^\top x_{t,a})) | \mathcal{H}_t] \leq \exp(b^2 R^2 / 2).$$

Recall that the random variable  $\mathbf{O}_t$  is the number of observed base arms in  $\mathbf{A}_t$  and at the end of every time  $t$ , the

agent observes the first  $\mathbf{O}_t$  items of  $\mathbf{A}_t$ . We say that item  $a$  is *observed* if  $a = a_k^t$  for some  $k \leq \mathbf{O}_t$ . Thus,  $\mathcal{H}_t$  consists of  $\{x_s, \mathbf{A}_s = (a_1^s, \dots, a_{|\mathbf{A}_s|}^s), \mathbf{O}_s, \mathbf{w}_s(a_k^s)\}_{k \leq \mathbf{O}_s, s < t}$  and  $\{x_t\}$ .

We introduce position discounts  $\gamma_k \in [0, 1], k \leq K$ . For example in website recommendations by a search engine, if the user selects the first website, the learning agent will receive reward 1; and if the user selects the  $k$ -th website, the learning agent will receive a discounted reward  $\gamma_k$ .

The reward function on round  $t$  is an application dependent function. We assume the expected reward of action  $A$  is a function  $f(A, w)$  of *expected weights*  $w = (w(a))_{a \in E} \in [0, 1]^E$  (at time  $t$ ,  $w = (w_{t,a})_{a \in E} = (\theta_*^\top x_{t,a})_{a \in E}$ ) and satisfies the following two assumptions.

**Monotonicity** The expected reward function  $f(A, w)$  is a non-decreasing with respect to  $w$ : for any  $w, w' \in [0, 1]^E$ , if  $w(a) \leq w'(a)$ , we have  $f(A, w) \leq f(A, w')$ .

**Lipschitz continuity** The expected reward function  $f(A, w)$  is  $B$ -Lipschitz continuous with respect to  $w$  together with position discount parameters  $\gamma_k, k \leq K$ . More specifically, for any  $w, w' \in [0, 1]^E$ , we have

$$|f(A, w) - f(A, w')| \leq B \sum_{k=1}^{|A|} \gamma_k |w(a_k) - w'(a_k)|,$$

where  $A = (a_1, \dots, a_{|A|})$ .

The assumption of Lipschitz continuity gives an estimate of changes in the reward function. To obtain a good estimation of the reward function, the position  $k$  with large  $\gamma_k$  needs a good estimation of  $w(a_k)$ . This means the positions with large  $\gamma_k$  are more important to the reward function. For example in website recommendations, the learning agent will receive the highest reward if the user selects the first item, so the first position is the most important.

Notice that we do not require the knowledge how  $f(A, w)$  is defined. We only assume that the agent has access to an oracle  $\mathcal{O}_S(w)$  that outputs recommended action  $A$ . More precisely, an oracle  $\mathcal{O}_S(w)$  is called an  $\alpha$ -approximation oracle for some  $\alpha \leq 1$ , if on given input  $w$ , the oracle returns an action  $A = \mathcal{O}_S(w) \in \mathcal{S}$  satisfying  $f(A, w) \geq \alpha f(A^*, w)$  where  $A^* = \arg\max_{A \in \mathcal{S}} f(A, w)$ . In applications, the oracle  $\mathcal{O}_S(w)$  is often what the best offline algorithm can achieve. For this reason, in the regret defined next, we compare our online algorithm with what the best offline algorithm can produce, i.e.  $\alpha f_t^*$ .

The  $\alpha$ -regret of action  $A$  on time  $t$  is

$$R^\alpha(t, A) = \alpha f_t^* - f(A, w_t),$$

where  $f_t^* = f(A_t^*, w_t)$ ,  $A_t^* = \arg\max_{A \in \mathcal{S}} f(A, w_t)$  and

$w_t = (\theta_*^\top x_{t,a})_{a \in E}$ . Our goal is to minimize the  $\alpha$ -regret

$$R^\alpha(n) = \mathbb{E}[\sum_{t=1}^n R^\alpha(t, \mathbf{A}_t)].$$

## 4. Algorithms and Results

### 4.1. Algorithm

Before presenting the algorithm, we explain the main ideas in the design. By Eq.(1), at time  $s$ , we have

$$\mathbb{E}[(\gamma_k \mathbf{w}_{s, \mathbf{a}_k^s}) | \mathcal{H}_s] = \theta_*^\top (\gamma_k x_{s, \mathbf{a}_k^s}).$$

To estimate the expected reward, we first estimate  $\theta_*$ , which can be viewed as a ridge regression problem on samples  $x$  and labels  $\mathbf{w}$ . In details, using the ridge regression on data

$$\{(\gamma_k x_{s, \mathbf{a}_k^s}, \gamma_k \mathbf{w}_{s, \mathbf{a}_k^s})\}_{k \in [\mathbf{O}_s], s \in [t]},$$

we get an  $l^2$ -regularized least-squares estimate of  $\theta_*$  with regularization parameter  $\lambda > 0$ :

$$\hat{\theta}_t = (\mathbf{X}_t^\top \mathbf{X}_t + \lambda I)^{-1} \mathbf{X}_t^\top \mathbf{Y}_t, \quad (2)$$

where  $\mathbf{X}_t \in \mathbb{R}^{(\sum_{s=1}^t \mathbf{O}_s) \times d}$  is the matrix whose rows are  $\gamma_k x_{s, \mathbf{a}_k^s}^\top$  and  $\mathbf{Y}_t$  is a column vector whose elements are  $\gamma_k \mathbf{w}_{s, \mathbf{a}_k^s}$ ,  $k \in [\mathbf{O}_s]$ ,  $s \in [t]$ . Let

$$\mathbf{V}_t = \mathbf{X}_t^\top \mathbf{X}_t + \lambda I = \lambda I + \sum_{s=1}^t \sum_{k=1}^{\mathbf{O}_s} \gamma_k^2 x_{s, \mathbf{a}_k^s} x_{s, \mathbf{a}_k^s}^\top.$$

Then  $\mathbf{V}_t \in \mathbb{R}^{d \times d}$  is a symmetric positive definite matrix. For any symmetric positive definite matrix  $V \in \mathbb{R}^{d \times d}$  and any vector  $x \in \mathbb{R}^{d \times 1}$ , we define the 2-norm of  $x$  based on  $V$  to be  $\|x\|_V = (x^\top V x)^{\frac{1}{2}}$ . Next we obtain a good estimate of the difference between  $\hat{\theta}_t$  and  $\theta_*$  by Theorem 2 in (Abbasi-Yadkori et al., 2011), restated as follows.

**Lemma 4.1** (Abbasi – Yadkori et al., 2011) *Let*

$$\beta_t(\delta) = R \sqrt{\ln \left( \frac{\det(\mathbf{V}_t)}{\lambda^d \delta^2} \right)} + \sqrt{\lambda}. \quad (3)$$

*Then for any  $\delta > 0$ , with probability at least  $1 - \delta$ , for all  $t > 0$ , we have*

$$\|\hat{\theta}_t - \theta_*\|_{\mathbf{V}_t} \leq \beta_t(\delta). \quad (4)$$

Thus with high probability, the estimate  $\hat{\theta}$  lies in the ellipsoid centered at  $\theta_*$  with confidence radius  $\beta_t(\delta)$  under  $\mathbf{V}_t$  norm. Building on this, we can define an upper confidence bound of the expected weight for each base arm  $a$  by

$$\mathbf{U}_t(a) = \min \left\{ \hat{\theta}_{t-1}^\top x_{t,a} + \beta_{t-1}(\delta) \|x_{t,a}\|_{\mathbf{V}_{t-1}^{-1}}, 1 \right\}. \quad (5)$$

The fact that  $\mathbf{U}_t(a)$  is an upper confidence bound of expected weight  $w_{t,a} = \theta_*^\top x_{t,a}$  is proved in the following lemma.

**Lemma 4.2** *When Ineq.(4) holds for time  $t - 1$ , we have*

$$0 \leq \mathbf{U}_t(a) - w_{t,a} \leq 2\beta_{t-1}(\delta) \|x_{t,a}\|_{\mathbf{V}_{t-1}^{-1}}.$$

**Proof.** Note  $w_{t,a} = \theta_*^\top x_{t,a}$ . By Hölder's inequality,

$$\begin{aligned} \left| \hat{\theta}_{t-1}^\top x_{t,a} - \theta_*^\top x_{t,a} \right| &= \left| [\mathbf{V}_{t-1}^{1/2} (\hat{\theta}_{t-1} - \theta_*)]^\top (\mathbf{V}_{t-1}^{-1/2} x_{t,a}) \right| \\ &\leq \|\mathbf{V}_{t-1}^{1/2} (\hat{\theta}_{t-1} - \theta_*)\|_2 \|\mathbf{V}_{t-1}^{-1/2} x_{t,a}\|_2 \\ &= \|\hat{\theta}_{t-1} - \theta_*\|_{\mathbf{V}_{t-1}} \|x_{t,a}\|_{\mathbf{V}_{t-1}^{-1}} \\ &\leq \beta_{t-1}(\delta) \|x_{t,a}\|_{\mathbf{V}_{t-1}^{-1}}. \end{aligned}$$

Because  $1 - \theta_*^\top x_{t,a} \geq 0$  and

$$\begin{aligned} 0 &\leq (\hat{\theta}_{t-1}^\top x_{t,a} + \beta_{t-1}(\delta) \|x_{t,a}\|_{\mathbf{V}_{t-1}^{-1}}) - \theta_*^\top x_{t,a} \\ &\leq 2\beta_{t-1}(\delta) \|x_{t,a}\|_{\mathbf{V}_{t-1}^{-1}}, \end{aligned}$$

the claimed result is obtained.  $\square$

---

### Algorithm 1 C<sup>3</sup>-UCB

---

```

1: Parameters:
2:    $\{\gamma_k \in [0, 1]\}_{k \leq K}; \delta = \frac{1}{\sqrt{n}}; \lambda \geq C_\gamma = \sum_{k=1}^K \gamma_k^2$ 
3: Initialization:
4:    $\hat{\theta}_0 = 0, \beta_0(\delta) = 1, \mathbf{V}_0 = \lambda I, \mathbf{X}_0 = \emptyset, \mathbf{Y}_0 = \emptyset$ 
5: for all  $t = 1, 2, \dots, n$  do
6:   Obtain context  $x_{t,a}$  for all  $a \in E$ 
7:    $\forall a \in E$ , compute
8:      $\mathbf{U}_t(a) = \min\{\hat{\theta}_{t-1}^\top x_{t,a} + \beta_{t-1}(\delta) \|x_{t,a}\|_{\mathbf{V}_{t-1}^{-1}}, 1\}$ 
9:   //Choose action  $\mathbf{A}_t$  using UCBs  $\mathbf{U}_t$ 
10:   $\mathbf{A}_t = (\mathbf{a}_1^t, \dots, \mathbf{a}_{|\mathbf{A}_t|}^t) \leftarrow \mathcal{O}_S(\mathbf{U}_t)$ 
11:  Play  $\mathbf{A}_t$  and observe  $\mathbf{O}_t, \mathbf{w}_t(\mathbf{a}_k^t), k \in [\mathbf{O}_t]$ 
12:  //Update statistics
13:   $\mathbf{V}_t \leftarrow \mathbf{V}_{t-1} + \sum_{k=1}^{\mathbf{O}_t} \gamma_k^2 x_{t, \mathbf{a}_k^t} x_{t, \mathbf{a}_k^t}^\top$ 
14:   $\mathbf{X}_t \leftarrow [\mathbf{X}_{t-1}; \gamma_1 x_{t, \mathbf{a}_1^t}^\top; \dots; \gamma_{\mathbf{O}_t} x_{t, \mathbf{a}_{\mathbf{O}_t}^t}^\top]$ 
15:   $\mathbf{Y}_t \leftarrow [\mathbf{Y}_{t-1}; \gamma_1 \mathbf{w}_t(\mathbf{a}_1^t); \dots; \gamma_{\mathbf{O}_t} \mathbf{w}_t(\mathbf{a}_{\mathbf{O}_t}^t)]$ 
16:   $\hat{\theta}_t \leftarrow (\mathbf{X}_t^\top \mathbf{X}_t + \lambda I)^{-1} \mathbf{X}_t^\top \mathbf{Y}_t$ 
17:   $\beta_t(\delta) \leftarrow R \sqrt{\ln(\det(\mathbf{V}_t)/(\lambda^d \delta^2))} + \sqrt{\lambda}$ 
18: end for  $t$ 
```

---

Based on the above analysis, we design our algorithm as follows. First, the learning agent computes the upper confidence bounds (UCBs)  $\mathbf{U}_t \in [0, 1]^E$  for the expected weights of all base arms in  $E$ . Second, the agent uses the computed UCBs,  $\mathbf{U}_t$ , to select an action  $\mathbf{A}_t = (\mathbf{a}_1^t, \dots, \mathbf{a}_{|\mathbf{A}_t|}^t)$ . Third, assume that the user checks from the first base arm in  $\mathbf{A}_t$  and stops after checking the  $\mathbf{O}_t$ -th base arm and the agents observes  $\mathbf{w}_t(\mathbf{a}_k^t), k \leq \mathbf{O}_t$ . Then, the



learning agent updates  $\mathbf{V}_t, \mathbf{X}_t, \mathbf{Y}_t$  in order to get a newer estimate  $\hat{\theta}_t$  of  $\theta_*$  and new confidence radius  $\beta_t(\delta)$ . Our proposed algorithm,  $C^3$ -UCB, is described in **Algorithm 1**. There we use the notation  $[A; B]$  to denote the matrix obtained by stacking  $A$  and  $B$  vertically like  $\begin{pmatrix} A \\ B \end{pmatrix}$ .

## 4.2. Results

### 4.2.1. GENERAL RESULTS

To state our main theorem, we need some definitions. Let  $p_{t,A}$  be the probability of full observation of  $A$ , that is, the probability of observing all base arms of  $A = (a_1, \dots, a_{|A|})$  at time  $t$ . Let  $p^* = \min_{1 \leq t \leq T} \min_{A \in \mathcal{S}} p_{t,A}$  be the minimal probability that an action could have all base arms observed over all time. The following is the main theorem on the regret achieved by our  $C^3$ -UCB algorithm.

**Theorem 4.3** *Suppose the expected reward function  $f(A, w)$  is a function of expected weights and satisfies the requirements of monotonicity and  $B$ -Lipschitz continuity. Then the  $\alpha$ -regret of our algorithm,  $C^3$ -UCB, satisfies*

$$R^\alpha(n) \leq \frac{2\sqrt{2}B}{p^*} \sqrt{nKd \ln(1 + C_\gamma n/(\lambda d))} \cdot \left( R \sqrt{\ln[(1 + C_\gamma n/(\lambda d))^d n] + \sqrt{\lambda}} + \alpha \sqrt{n} \right) + O\left(\frac{dB R}{p^*} \sqrt{nK \ln(C_\gamma n)}\right), \quad (6)$$

where  $R$  is the sub-Gaussian constant and  $C_\gamma = \sum_{k=1}^K \gamma_k^2 \leq K$ .

The regret bound deteriorates as  $p^*$  gets smaller. But unfortunately, this reciprocal dependence on  $p^*$  is not known to be improvable, even in the special case of disjunctive objective. Indeed, in that case, the bound in Section 2.3 of (Kveton et al., 2015c) also has a reciprocal dependence on  $p^*$ . However, in the special case of conjunctive objective, we will give an improved bound.

**Proof.** (sketch) Assume  $\mathbf{A}_t = (\mathbf{a}_1^t, \dots, \mathbf{a}_{|\mathbf{A}_t|}^t)$ . We first employ the monotonicity and  $B$ -Lipschitz continuity of  $f$  to upper bound regret as follows.

$$R^\alpha(t, \mathbf{A}_t) \leq 2B \sum_{k=1}^{|\mathbf{A}_t|} \beta_{t-1}(\delta) \gamma_k \|x_{t, \mathbf{a}_k^t}\|_{\mathbf{V}_{t-1}^{-1}}. \quad (7)$$

As  $\mathbf{V}_t$  contains only information of observed base arms, our UCB-type algorithm guarantees the sum of the first  $\mathbf{O}_t$  items in the right of Ineq. 7 to be small, with the leftover sum (from  $\mathbf{O}_t + 1$  to  $|\mathbf{A}_t|$ ) out of control. We cope with this by a reduction to the case that  $\mathbf{O}_t = |\mathbf{A}_t|$ .

$$\mathbb{E}[R^\alpha(t, \mathbf{A}_t) | \mathcal{H}_t]$$

$$= \mathbb{E} \left[ R^\alpha(t, \mathbf{A}_t) \mathbb{E} \left[ \frac{1}{p_{t, \mathbf{A}_t}} \mathbb{1}\{\mathbf{O}_t = |\mathbf{A}_t|\} \middle| \mathbf{A}_t \right] \middle| \mathcal{H}_t \right] \leq \frac{1}{p^*} \mathbb{E} [R^\alpha(t, \mathbf{A}_t) \mathbb{1}\{\mathbf{O}_t = |\mathbf{A}_t|\} | \mathcal{H}_t] \dots \quad (8)$$

Combining 7 and 8 gives an upper bound of  $R^\alpha(n)$  in terms of  $\sum_{t=1}^n \sum_{k=1}^{|\mathbf{O}_t|} \gamma_k \|x_{t, \mathbf{a}_k^t}\|_{\mathbf{V}_{t-1}^{-1}}$ . The next lemma upper bounds this sum of norms, squared.

**Lemma 4.4** *If  $\lambda \geq C_\gamma$ , then for any time  $t$ ,*

$$\sum_{s=1}^t \sum_{k=1}^{\mathbf{O}_s} \|\gamma_k x_{s, \mathbf{a}_k^s}\|_{\mathbf{V}_{s-1}^{-1}}^2 \leq 2d \ln(1 + C_\gamma t/(\lambda d)).$$

The claimed result follows from Cauchy-Schwartz inequality. See Appendix for a complete proof.  $\square$

### 4.2.2. DISJUNCTIVE OBJECTIVE

In the problem of cascading recommendation, when recommended with an ordered list of items  $\mathbf{A}_t = (\mathbf{a}_1^t, \dots, \mathbf{a}_{|\mathbf{A}_t|}^t)$ , the user checks the list in that order. The checking process stops if the user selects one item or has checked all items without selecting anyone. The weight of each base arm  $a$  at time  $t$ ,  $\mathbf{w}_t(a)$ , is a  $\{0, 1\}$  value indicating whether the user has selected item  $a$  or not. Then the random variable  $\mathbf{O}_t$  satisfies

$$\mathbf{O}_t = \begin{cases} k, & \text{if } \mathbf{w}_t(\mathbf{a}_j^t) = 0, \forall j < k \text{ and } \mathbf{w}_t(\mathbf{a}_k^t) = 1, \\ |\mathbf{A}_t|, & \text{if } \mathbf{w}_t(\mathbf{a}_j^t) = 0, \forall j \leq |\mathbf{A}_t|. \end{cases}$$

If the user selects the  $k$ -th item, then the learning agent receives a reward  $\gamma_k \in [0, 1]$ . Usually  $\{\gamma_k\}$  are decreasing with  $k$ :

$$1 = \gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_K \geq 0.$$

If the user does not select any item, the agent gets no reward. At the end of time step  $t$ , the learning agent observes  $\mathbf{O}_t, \mathbf{w}_t(\mathbf{a}_k^t), k \leq \mathbf{O}_t$  and receives a reward  $\mathbf{r}_t = \mathbf{w}_t(\mathbf{a}_{\mathbf{O}_t}^t) \gamma_{\mathbf{O}_t}$ . It is not hard to see that  $\mathbf{r}_t = \bigvee_{k=1}^{|\mathbf{A}_t|} \gamma_k \mathbf{w}_t(\mathbf{a}_k^t)$  is the disjunctive of discounted weights, where we use the notation that  $\bigvee_{k=1}^n a_k = \max_{1 \leq k \leq n} a_k$ . Notice that the order of  $\mathbf{A}_t$  affects both the feedback and the reward.

Now let us define a function  $f : \mathcal{S} \times [0, 1]^E \rightarrow [0, 1]$  on  $A = (a_1, \dots, a_{|A|}) \in \mathcal{S}, w = (w(1), \dots, w(L))$  by

$$f(A, w) = \sum_{k=1}^{|A|} \gamma_k \prod_{i=1}^{k-1} (1 - w(a_i)) w(a_k). \quad (9)$$

It is easily verified that  $\mathbf{r}_t = f(\mathbf{A}_t, \mathbf{w}_t)$  and that  $\mathbb{E}[\mathbf{r}_t] = f(\mathbf{A}_t, \theta_*^\top x_t) = f(\mathbf{A}_t, w_t)$ . So  $f$  is the expected reward function and also a function of expected weights. It can

be proved that  $f$  satisfies the properties of monotonicity and 1-Lipschitz continuity. (All verifications are left in Appendix.) We thus have the following corollary.

**Corollary 4.5** *In the problem of cascading recommendation, the expected reward function is defined in Eq. (9), where  $1 = \gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_K \geq 0$ . Then the  $\alpha$ -regret of our algorithm,  $C^3$ -UCB, satisfies*

$$\begin{aligned} R^\alpha(n) &\leq \frac{2\sqrt{2}}{p^*} \sqrt{nKd \ln(1 + C_\gamma n / (\lambda d))} \\ &\quad \cdot \left( \sqrt{d \ln[(1 + C_\gamma n / (\lambda d))^d n] + \sqrt{\lambda}} + \alpha \sqrt{n} \right) \\ &= O\left(\frac{d}{p^*} \sqrt{nK} \ln(C_\gamma n)\right). \end{aligned} \quad (10)$$

#### 4.2.3. CONJUNCTIVE OBJECTIVE

The above formulation is on the disjunctive objective, for which the user stops once she finds a “good” item. Similarly, we could also consider the case of conjunctive objective, for which the user stops once she finds a “bad” item. In this case, we derive a better result.

The Bernoulli random variable  $\mathbf{w}_t(a) \in \{0, 1\}$  indicates the weight of item  $a$  at time  $t$  satisfying Eq. (1). The learning agent observes the first position  $k$  in the given action  $\mathbf{A}_t = (\mathbf{a}_1^t, \dots, \mathbf{a}_{|\mathbf{A}_t|}^t)$  with  $\mathbf{w}_t(\mathbf{a}_k^t) = 0$ . We also consider partial rewards: if the  $k$ -th item is the first item with weight 0, then the learning agent receives reward  $1 - \gamma_k$ ; if all items have weight 1, the learning agent receives reward 1.

The more items reveals, the more reward the agent should receive. This can be used to model scenarios such as on-line surveys, where questions are adaptively given, and the more questions are observed, the more helpful it is to the survey conductor. Based on this, we assume that

$$1 = \gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_K \geq 0.$$

At the end of time step  $t$ , the learning agent observes  $\mathbf{O}_t, \mathbf{w}_t(\mathbf{a}_k^t), \forall k \leq \mathbf{O}_t$  and receives a reward  $\mathbf{r}_t$ :

$$\mathbf{r}_t = \begin{cases} 1 - \gamma_k & \text{if } \mathbf{w}_t(\mathbf{a}_i^t) = 1, \forall i < k, \text{ and } \mathbf{w}_t(\mathbf{a}_k^t) = 0, \\ 1 & \text{if } \mathbf{w}_t(\mathbf{a}_i^t) = 1, \forall i \leq |\mathbf{A}_t|. \end{cases}$$

If we define a function  $f : \mathcal{S} \times [0, 1]^E \rightarrow [0, 1]$  on  $A = (a_1, \dots, a_{|A|}) \in \mathcal{S}$  and  $w = (w(1), \dots, w(L))$  by

$$f(A, w) = \sum_{k=1}^{|A|} (1 - \gamma_k) \prod_{i=1}^{k-1} w(a_i) (1 - w(a_k)) + \prod_{i=1}^{|A|} w(a_i), \quad (11)$$

then it is not hard to verify that  $\mathbf{r}_t = f(\mathbf{A}_t, \mathbf{w}_t)$  and  $\mathbb{E}[\mathbf{r}_t] = f(\mathbf{A}_t, w_t) = f(\mathbf{A}_t, \theta_*^\top x_t)$ . Next is our result for

the conjunctive objective, where the bound in Theorem 4.3 is improved with  $p^*$  replaced by  $\alpha f^*$ . In network routing problem, the  $p^*$  is related with the bad paths, which can make the probability of observing the whole path very small, and  $f^*$  is related with good paths, which makes  $f^*$  usually large.

**Theorem 4.6** *Suppose  $1 = \gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_K \geq 1 - \frac{\alpha}{4} f^*$ , where  $f^* = \min_t f_t^*$  with  $f_t^* = \max_{A \in \mathcal{S}} f(A, w_t)$ . Then the  $\alpha$ -regret of our algorithm,  $C^3$ -UCB, for the conjunctive objective satisfies*

$$\begin{aligned} R^\alpha(n) &\leq \frac{\sqrt{128}}{\alpha f^*} \sqrt{nKd \ln(1 + C_\gamma n / (\lambda d))} \\ &\quad \cdot \left( \sqrt{d \ln[(1 + C_\gamma n / (\lambda d))^d n] + \sqrt{\lambda}} + \alpha \sqrt{n} \right) \\ &= O\left(\frac{d}{\alpha f^*} \sqrt{nK} \ln(C_\gamma n)\right). \end{aligned} \quad (12)$$

**Proof.** (sketch) First, we prove the expected reward function satisfies the properties of monotonicity and Lipschitz continuity. Then we use the following lemma to replace  $\mathbf{A}_t$  with a prefix of  $\mathbf{A}_t$ .

**Lemma 4.7** *Suppose  $1 = \gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_K \geq 0$ . Let  $A = (a_1, \dots, a_{|A|})$ . For the time  $t$  and the conjunctive objective, there exists a prefix  $B$  of  $A$  such that*

$$p_{t,B} \geq \frac{\alpha}{2} f^* - 1 + \gamma_{|B|}, \quad R^\alpha(t, B) \geq \frac{1}{2} R^\alpha(t, A).$$

Therefore we can get

$$\mathbb{E}[R^\alpha(t, \mathbf{A}_t) | \mathcal{H}_t] \leq \frac{8}{\alpha f^*} \mathbb{E} \left[ \beta_{t-1}(\delta) \sum_{k=1}^{\mathbf{O}_t} \|\gamma_k x_{t, \mathbf{a}_k^t}\|_{\mathbf{V}_{t-1}^{-1}} \middle| \mathcal{H}_t \right].$$

Then similar to the proof of Theorem 4.3 and by the Lemma 4.4, we can prove this theorem. See Appendix for the complete proof.  $\square$

### 4.3. Discussions

In the above, we formulate the general setting of contextual combinatorial cascading bandit with position discounts. This setting generalizes several existing results.

The combinatorial cascading bandit (Kveton et al., 2015c) has a setting similar to ours in Section 4.2.3, but theirs has no contextual information or position discounts (i.e. all  $\gamma_k = 1$ ). So our setting in Section 4.2.3 is more general, but our regret bound is of order  $O\left(\frac{d}{f^*} \sqrt{nK} \ln(n)\right)$ , which has

an additional term  $\sqrt{\frac{d \ln(n)}{f^*}}$  compared to their result. The loss is because to handle the contextual information and

obtain a confidence ellipsoid of  $\theta_*$ , we used the technique of linear bandits (Abbasi-Yadkori et al., 2011), whose regret bound has an additional term of  $O(\sqrt{d \ln(n)})$  than the standard stochastic MAB bound  $O(\sqrt{nd \ln(n)})$ . A similar comparison can be made between our Section 4.2.2 and Section 2.3 of (Kveton et al., 2015c).

We can also compare to the work (Qin et al., 2014), where there are no triggering and no position discounts, and all base arms in action  $\mathbf{A}_t$  can be observed. So in their setting, our random variable  $\mathbf{O}_t$  becomes a deterministic value  $|\mathbf{A}_t|$  and the probability,  $p_{t, \mathbf{A}_t}$ , is 1. Thus  $p^* = 1$ . Also all position discounts  $\gamma_k = 1$ . Then our regret bound of Theorem 4.3 is of the same order with theirs. Notice that the Lipschitz bound  $C$  in their setting is  $\sqrt{K}$ .

Our algorithm also applies to the setting of (Chen et al., 2013) by allowing contextual information. Note the probability  $p^*$  in our results should be the probability that all base arms in  $\tilde{S}$  (in their notation) will be triggered.

Our setting generalizes several existing works on cascading feedback, and actually covers more cases. Consider in network routing problem, the observed information is the latency on edges. The latency, or *weight*, on an edge follows the exponential distribution with mean  $\theta_*^\top x$ . Suppose we treat an edge as blocked if its latency is larger than some tolerance  $\tau$ . And we wait the edge to respond for at most  $2\tau$  time. Then the real observed latency is a new random variable from a cut-off exponential distribution with mean  $\theta_*^\top x$ . The edge has reward 1 if it is unblocked; otherwise, the reward is 0. Then the expectation of the reward is a function, say  $g$ , of  $\theta_*^\top x$ . It is not difficult to check that  $g$  satisfies the properties of monotonicity and Lipschitz continuity. A path has reward 1 only when all its edges have rewards 1. So the expected reward of a path is the expected reward function of conjunctive objective, defined in Eq.(11), by replacing  $w$  with  $g(\theta_*^\top x)$ . Such an expected reward function satisfies the properties of monotonicity and Lipschitz continuity, and can not be represented in conjunctive/disjunctive objectives as in (Kveton et al., 2015a;c).

## 5. Experiments

We evaluate our algorithm,  $C^3$ -UCB, in a synthetic setting and two real applications. The results are compared with *CombCascade*, the study most related to ours, and demonstrate the advantage to involve contextual information and position discounts. In experiments, we set the position discounts  $\gamma_k$  to be  $\gamma^{k-1}$  for some  $\gamma$ .

### 5.1. Synthetic Data

In the first experiment, we compare  $C^3$ -UCB to *CombCascade* on synthetic problems. This problem is a contextual cascading bandit with  $L = 100$  items and  $K = 4$ ,

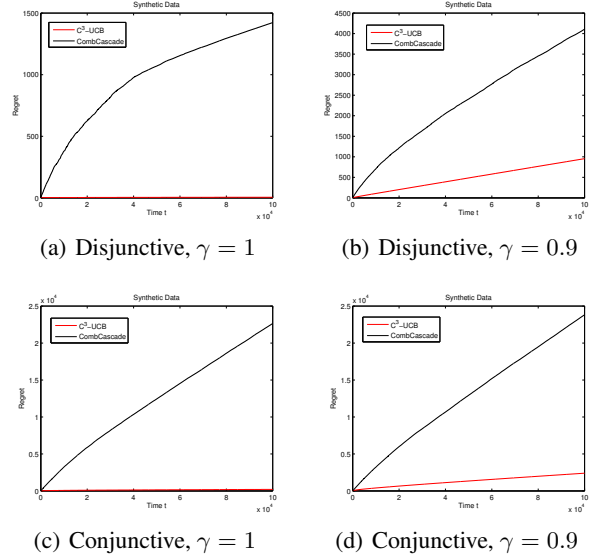


Figure 1. Synthetic Data Set,  $L=100$ ,  $K=4$ ,  $d=10$

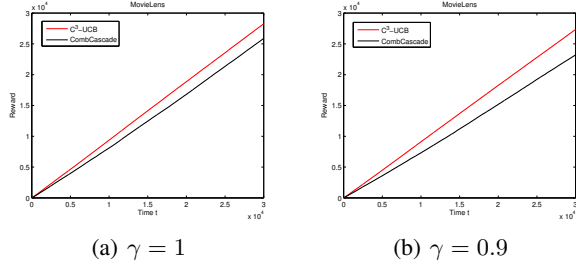
where at each time  $t$  the agent recommends  $K$  items to the user. At first, we randomly choose a  $\theta \in \mathbb{R}^{d-1}$  with  $\|\theta\|_2 = 1$  and let  $\theta_* = (\frac{\theta}{2}, \frac{1}{2})$ . Then at each time  $t$ , we randomly assign  $x'_{t,a} \in \mathbb{R}^{d-1}$  with  $\|x'_{t,a}\|_2 = 1$  to arm  $a$  and use  $x_{t,a} = (x'_{t,a}, 1)$  to be the contextual information for arm  $a$ . This processing will guarantee the inner product  $\theta_*^\top x_{t,a} = \frac{1}{2}(\theta^\top x'_{t,a} + 1) \in [0, 1]$ . Next we generate the weight for arm  $a$  at time  $t$  by a random sample from the Bernoulli distribution with mean  $\theta_*^\top x_{t,a}$ .

We conduct experiments in four settings. Under each setting, the learning agent chooses a set of  $K$  items out of  $L$  ground items. The first two are of disjunctive objective where the learning agent observes a prefix of the chosen  $K$  items until the first one with weight 1; the last two are of conjunctive objective where the learning agent observes from the first item until the first one with weight 0. Notice that with  $\gamma \leq 1$ , the oracle selects a set of  $K$  items with highest UCBs in their decreasing order. The regrets are shown in Fig. 1 and our algorithm outperforms *CombCascade* algorithm because they do not make use of the contextual information.

### 5.2. Movie Recommendation

In this experiment, we evaluate  $C^3$ -UCB on movie recommendation with data set *MovieLens* (Lam & Herlocker, 2015) of 2015.

The learning problem is formulated as follows. There is a big sparse matrix  $A \in \{0, 1\}^{N_1 \times N_2}$  where  $A(i, j) = 1$  denotes user  $i$  has watched movie  $j$ . Next, we split  $A$  as  $H + F$  by putting 1-entry in  $A$  to  $H$  or  $F$  according to a Bernoulli distribution  $\sim \text{Ber}(p)$  for some fixed  $p$ . We

Figure 2. Rewards on MovieLens,  $L = 200$ ,  $K = 4$ ,  $d = 5$ 

regard  $H$  as known information about history “what users have watched” and regard  $F$  as future criterion. We use  $H$  to derive feature vectors of both users and movies by SVD decomposition,  $H = USM^\top$  where  $U = (u_1; \dots; u_{N_1})$  and  $M = (m_1; \dots; m_{N_2})$ . At every time  $t$ , we randomly choose a user  $I_t \in [N_1]$ . Then in the same spirit of (Li et al., 2010), we use  $x_{t,j} = u_{I_t} m_j^\top$ , the outer product of  $u_{I_t}$  and  $m_j$ , as the contextual information for each movie  $j$ . The real weight of movie  $j$  at time  $t$ ,  $w_t(j)$ , is  $F(I_t, m)$ .

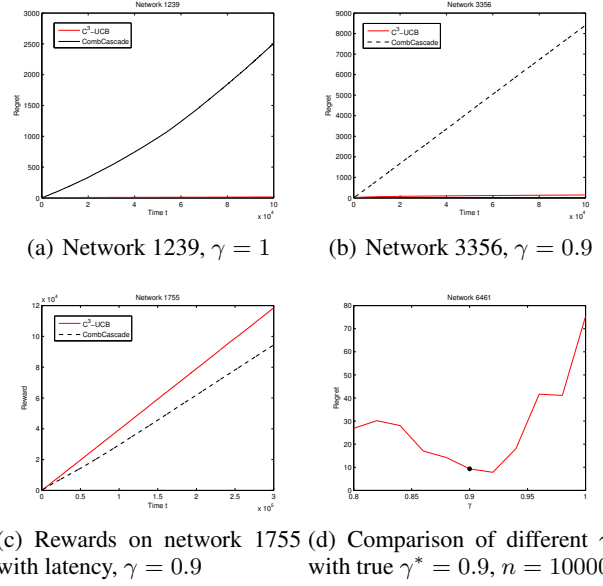
For this experiment, we randomly choose  $L = 200$  movies and recommend  $K = 4$  movies at each time. We experiment with both  $\gamma = 1$  (no position discount) and  $\gamma = 0.9$ , and compare our algorithm with *CombCascade*. The results for rewards are shown in Fig. 2. The rewards of our algorithms are higher than those of *CombCascade* by 9.2% and 17.74% (for  $\gamma = 1$  and 0.9, respectively), which demonstrate the advantage to involve contextual information in real applications.

### 5.3. Network Routing Data

In the last experiment, we evaluate  $C^3$ -UCB on network routing problem with *RocketFuel* dataset (Spring et al., 2004).

The ground set  $E$  is the set of links in the network. Before learning, the environment randomly chooses a  $d$ -dimensional vector  $\theta_* \in [0, 1]^d$ . At each time  $t$ , a pair of source and destination nodes are randomly chosen and the feasible action set  $\mathcal{S}_t$  at time  $t$  contains all simple paths between the source and destination. Any edge  $a$  in the set  $\mathcal{S}_t$  is assigned with a random  $d$ -dimensional contextual information vector  $x_{t,a}$ . Notice here both  $\theta$  and  $x$  have been processed like in Section 5.1 such that  $\theta_*^\top x \in [0, 1]$ . The weight for each edge  $a$  is a sample from Bernoulli distribution with mean  $\theta_*^\top x_{t,a}$ . Then the learning agent recommends a feasible path  $A = (a_1, \dots, a_{|A|})$  between source and destination nodes to maximize the expected reward in the conjunctive objective. We experiment on different position discounts. The regrets are shown in Fig. 3 (a), (b).

Another experiment is conducted on the network with la-

Figure 3. Network,  $d = 10$ 

tency, where the latency on an edge is an exponential random variable with mean  $\theta_*^\top x$  and the reward is 1 if the latency is less than a tolerance  $\tau = 0.75$ . The comparison of the cumulative rewards of our algorithm with *CombCascade* is shown in Fig. 3 (c).

We also conduct an experiment on the influence of  $\gamma$  to the regret. In recommendations, the position discounts represent users’ satisfaction and the learning agent may not know its true value. Suppose the true position discount  $\gamma^* = 0.9$  and the learning agent sets  $\gamma$  by her experience. We experiment on different  $\gamma \in \{0.8, 0.82, \dots, 0.98, 1.0\}$  with the true criterion  $\gamma^* = 0.9$ . The regrets with different  $\gamma$ ’s are illustrated in Fig. 3(d), from which it can be seen that the regret is minimized around the true  $\gamma^*$ . The big gap between the regret at  $\gamma = 1$  and  $\gamma^* = 0.9$  shows that it significantly benefits by exploiting position discounts in the model when such discounts indeed exist in applications.

## 6. Conclusions

In this paper, we propose contextual combinatorial cascading bandits with position discounts, where each action is an ordered list and only a prefix of the action is observed each time. We propose a  $C^3$ -UCB algorithm and prove a cumulative regret bound for general reward functions and two special reward functions. The experiments conducted demonstrate the advantage to involve contextual information and position discounts. In future, we would like to investigate on lower bounds of the regret and cascading on general graphs.



## References

- Abbasi-Yadkori, Yasin, Pál, Dávid, and Szepesvári, Csaba. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pp. 2312–2320, 2011.
- Agrawal, Rajeev, Teneketzis, Demosthenis, and Anantharam, Venkatachalam. Asymptotically efficient adaptive allocation schemes for controlled markov chains: Finite parameter space. *Automatic Control, IEEE Transactions on*, 34(12):1249–1259, 1989.
- Bartók, Gábor, Zolghadr, Navid, and Szepesvári, Csaba. An adaptive algorithm for finite stochastic partial monitoring. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.
- Chen, Shouyuan, Lin, Tian, King, Irwin, Lyu, Michael R, and Chen, Wei. Combinatorial pure exploration of multi-armed bandits. In *NIPS*, 2014.
- Chen, Wei, Wang, Yajun, and Yuan, Yang. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 151–159, 2013.
- Chen, Wei, Wang, Yajun, and Yuan, Yang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *arXiv preprint arXiv:1407.8339*, 2015.
- Combes, Richard, Magureanu, Stefan, Proutiere, Alexandre, and Laroche, Cyrille. Learning to rank: Regret lower bounds and efficient algorithms. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pp. 231–244. ACM, 2015a.
- Combes, Richard, Talebi, M. Sadegh, Proutiere, Alexandre, and Lelarge, Marc. Combinatorial bandits revisited. In *NIPS*, 2015b.
- Gai, Yi, Krishnamachari, Bhaskar, and Jain, Rahul. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking (TON)*, 20(5):1466–1478, 2012.
- Gopalan, Aditya, Mannor, Shie, and mansour, Yishay. Thompson sampling for complex online problems. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.
- Kveton, Branislav, Wen, Zheng, Ashkan, Azin, Eydgahi, Hoda, and Eriksson, Brian. Matroid bandits: Fast combinatorial optimization with learning. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2014.
- Kveton, Branislav, Szepesvári, Csaba, Wen, Zheng, and Ashkan, Azin. Cascading bandits: learning to rank in the cascade model. In *Proceedings of the 32th International Conference on Machine Learning*, 2015a.
- Kveton, Branislav, Wen, Zheng, Ashkan, Azin, and Szepesvári, Csaba. Tight regret bounds for stochastic combinatorial semi-bandits. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015b.
- Kveton, Branislav, Wen, Zheng, Ashkan, Azin, and Szepesvári, Csaba. Combinatorial cascading bandits. *Advances in Neural Information Processing Systems*, 2015c.
- Lam, Shyong and Herlocker, Jon. Movielens 20m dataset. 2015. URL <http://grouplens.org/datasets/movielens/20m/>.
- Li, Lihong, Chu, Wei, Langford, John, and Schapire, Robert E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670. ACM, 2010.
- Lin, Tian, Abrahao, Bruno, Kleinberg, Robert, Lui, John, and Chen, Wei. Combinatorial partial monitoring game with linear feedback and its applications. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 901–909, 2014.
- Lin, Tian, Li, Jian, and Chen, Wei. Stochastic online greedy learning with semi-bandit feedbacks. In *NIPS*, 2015.
- Qin, Lijing, Chen, Shouyuan, and Zhu, Xiaoyan. Contextual combinatorial bandit and its application on diversified online recommendation. In *Proceedings of the 2014 SIAM International Conference on Data Mining (SDM)*, 2014.
- Spring, Neil, Mahajan, Ratul, Wetherall, David, and Anderson, Thomas. Measuring isp topologies with rocketfuel. *Networking, IEEE/ACM Transactions on*, 12(1): 2–16, 2004.