DCM Bandits: Learning to Rank with Multiple Clicks

Sumeet Katariya KATARIYA@WISC.EDU
Department of Electrical and Computer Engineering, University of Wisconsin-Madison

Branislav Kveton KVETON@ADOBE.COM
Adobe Research, San Jose, CA

Csaba Szepesvári SZEPESVA@CS.UALBERTA.CA
Department of Computing Science, University of Alberta

Zheng Wen ZWEN@ADOBE.COM
Adobe Research, San Jose, CA

Abstract

A search engine recommends to the user a list of web pages. The user examines this list, from the first page to the last, and clicks on all attractive pages until the user is satisfied. This behavior of the user can be described by the *dependent click model (DCM)*. We propose *DCM bandits*, an online learning variant of the DCM where the goal is to maximize the probability of recommending satisfactory items, such as web pages. The main challenge of our learning problem is that we do not observe which attractive item is satisfactory. We propose a computationally-efficient learning algorithm for solving our problem, `dcmKL-UCB`; derive gap-dependent upper bounds on its regret under reasonable assumptions; and also prove a matching lower bound up to logarithmic factors. We evaluate our algorithm on synthetic and real-world problems, and show that it performs well even when our model is misspecified. This work presents the first practical and regret-optimal online algorithm for learning to rank with multiple clicks in a cascade-like click model.

1. Introduction

Web pages in search engines are often ranked based on a model of user behavior, which is learned from click data (Radlinski & Joachims, 2005; Agichtein et al., 2006; Chuklin et al., 2015). The cascade model (Craswell et al., 2008) is one of the most popular models of user behavior in web search. Kveton et al. (2015a) and Combes et al. (2015a) recently proposed regret-optimal online learning algorithms for the cascade model. The main limitation of the cascade model is that it cannot model multiple clicks. Although the model was extended to multiple clicks (Chapelle & Zhang, 2009; Guo et al., 2009a;b), it is unclear if it is possible to design computationally and sample efficient online learning algorithms for these extensions.

In this work, we propose an online learning variant of the *dependent click model (DCM)* (Guo et al., 2009b), which we call *DCM bandits*. The DCM is a generalization of the cascade model where the user may click on multiple items. At time t, our learning agent recommends to the user a list of K items. The user examines this list, from the first item to the last. If the examined item attracts the user, the user clicks on it. This is observed by the learning agent. After the user clicks on the item and investigates it, the user may leave or examine more items. If the user leaves, the DCM interprets this as that the user is satisfied and our agent receives a reward of one. If the user examines all items and does not leave on purpose, our agent receives a reward of zero. The goal of the agent is to maximize its total reward, or equivalently to minimize its cumulative regret with respect to the most satisfactory list of K items. Our learning problem is challenging because the agent does not observe whether the user is satisfied. It only observes clicks. This differentiates our problem from cascading bandits (Kveton et al., 2015a), where the user can click on at most one item and this click is satisfactory.

We make four major contributions. First, we formulate an online learning variant of the DCM. Second, we propose a computationally-efficient learning algorithm for our problem under the assumption that the order of the termination

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

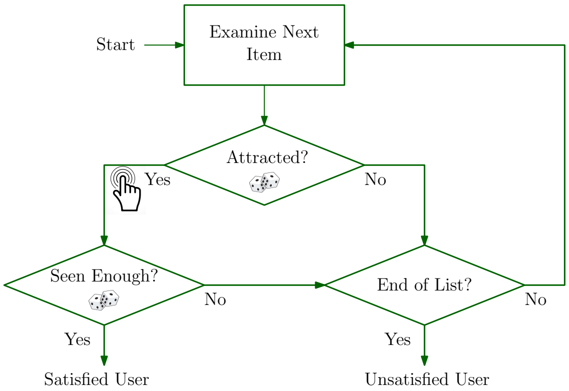


Figure 1. Interaction between the user and items in the DCM.

probabilities in the DCM is known. Our algorithm is motivated by `KL-UCB` (Garivier & Cappe, 2011), and therefore we call it `dcmKL-UCB`. Third, we prove two gap-dependent upper bounds on the regret of `dcmKL-UCB` and a matching lower bound up to logarithmic factors. The key step in our analysis is a novel reduction to cascading bandits (Kveton et al., 2015a). Finally, we evaluate our algorithm on both synthetic and real-world problems, and compare it to several baselines. We observe that `dcmKL-UCB` performs well even when our modeling assumptions are violated.

We denote random variables by boldface letters and write $[n]$ to denote $\{1, \dots, n\}$. For any sets A and B, we denote by A^B the set of all vectors whose entries are indexed by B and take values from A.

2. Background

Web pages in search engines are often ranked based on a model of user behavior, which is learned from click data (Radlinski & Joachims, 2005; Agichtein et al., 2006; Chuklin et al., 2015). We assume that the user scans a list of K web pages $A = (a_1, \dots, a_K)$, which we call *items*. These items belong to some *ground set* $E = [L]$, such as the set of all possible web pages. Many models of user behavior in web search exist (Becker et al., 2007; Richardson et al., 2007; Craswell et al., 2008; Chapelle & Zhang, 2009; Guo et al., 2009a;b). We focus on the dependent click model.

The *dependent click model (DCM)* (Guo et al., 2009b) is an extension of the cascade model (Craswell et al., 2008) to multiple clicks. The model assumes that the user scans a list of K items $A = (a_1, \dots, a_K) \in \Pi_K(E)$ from the first item a_1 to the last a_K, where $\Pi_K(E) \subset E^K$ is the set of all K*-permutations* of E. The DCM is parameterized by L *item-dependent attraction probabilities* $\bar{w} \in [0,1]^E$ and K *position-dependent termination probabilities* $\bar{v} \in [0,1]^K$. After the user *examines* item a_k, the item *attracts* the user with probability $\bar{w}(a_k)$. If the user is attracted by item a_k, the user clicks on the item and *terminates* the search with probability $\bar{v}(k)$. If this happens, the user is *satisfied* with item a_k and does not examine any of the *remaining* items. If item a_k is not attractive or the user does not terminate, the user examines item a_{k+1}. Our interaction model is visualized in Figure 1.

Before we proceed, we would like to stress the following. First, all probabilities in the DCM are independent of each other. Second, the probabilities $\bar{w}(a_k)$ and $\bar{v}(k)$ are *conditioned* on the events that the user examines position k and that the examined item is attractive, respectively. For simplicity of exposition, we drop "conditional" in this paper. Finally, $\bar{v}(k)$ is *not* the probability that the user terminates at position k. This latter probability depends on the items and positions before position k.

It is easy to see that the probability that the user is satisfied with list $A = (a_1, \dots, a_K)$ is $1 - \prod_{k=1}^{K}(1 - \bar{v}(k)\bar{w}(a_k))$. This objective is maximized when the k-th most attractive item is placed at the k-th most terminating position.

3. DCM Bandits

We propose a learning variant of the dependent click model (Section 3.1) and a computationally-efficient algorithm for solving it (Section 3.3).

3.1. Setting

We refer to our learning problem as a *DCM bandit*. Formally, we define it as a tuple $B = (E, P_\text{W}, P_\text{V}, K)$, where $E = [L]$ is a *ground set* of L items; P_W and P_V are probability distributions over $\{0,1\}^E$ and $\{0,1\}^K$, respectively; and $K \leq L$ is the number of recommended items.

The learning agent interacts with our problem as follows. Let $(\mathbf{w}_t)_{t=1}^n$ be n i.i.d. *attraction weights* drawn from distribution P_W, where $\mathbf{w}_t \in \{0,1\}^E$ and $\mathbf{w}_t(e)$ indicates that item e is attractive at time t; and let $(\mathbf{v}_t)_{t=1}^n$ be n i.i.d. *termination weights* drawn from P_V, where $\mathbf{v}_t \in \{0,1\}^K$ and $\mathbf{v}_t(k)$ indicates that the user would terminate at position k if the item at that position was examined and attractive. At time t, the learning agent recommends to the user a list of K items $\mathbf{A}_t = (\mathbf{a}_1^t, \dots, \mathbf{a}_K^t) \in \Pi_K(E)$. The user examines the items in the order in which they are presented and the agent receives observations $\mathbf{c}_t \in \{0,1\}^K$ that indicate the clicks of the user. Specifically, $\mathbf{c}_t(k) = 1$ if and only if the user clicks on item $\mathbf{a}_k^t$, the item at position k at time t.

The learning agent also receives a *binary reward* $\mathbf{r}_t$, which is *unobserved*. The reward is one if and only if the user is satisfied with at least one item in $\mathbf{A}_t$. We say that item e is *satisfactory* at time t when it is attractive, $\mathbf{w}_t(e) = 1$, and its position leads to termination, $\mathbf{v}_t(k) = 1$. The reward can be written as $\mathbf{r}_t = f(\mathbf{A}_t, \mathbf{w}_t, \mathbf{v}_t)$, where $f : \Pi_K(E) \times$

$[0,1]^E \times [0,1]^K \to [0,1]$ is a *reward function*, which we define as

$$f(A, w, v) = 1 - \prod_{k=1}^{K} (1 - v(k) w(a_k))$$

for any $A = (a_1, \dots, a_K) \in \Pi_K(E)$, $w \in [0,1]^E$, and $v \in [0,1]^K$. The above form is very useful in our analysis.

Guo et al. (2009b) assume that the attraction and termination weights in the DCM are drawn *independently* of each other. We also adopt these assumptions. More specifically, we assume that for any $w \in \{0,1\}^E$ and $v \in \{0,1\}^K$,

$$P_{\mathrm{w}}(w) = \prod\nolimits_{e \in E} \mathrm{Ber}(w(e); \bar{w}(e)) \,,$$
$$P_{\mathrm{v}}(v) = \prod\nolimits_{k \in [K]} \mathrm{Ber}(v(k); \bar{v}(k)) \,,$$

where $\mathrm{Ber}(\cdot; \theta)$ is a Bernoulli probability distribution with mean θ. The above assumptions allow us to design a very efficient learning algorithm. In particular, they imply that the expected reward for list A, the probability that at least one item in A is satisfactory, decomposes as

$$\begin{aligned} \mathbb{E}\left[f(A, \mathbf{w}, \mathbf{v})\right] &= 1 - \prod_{k=1}^{K} (1 - \mathbb{E}\left[\mathbf{v}(k)\right] \mathbb{E}\left[\mathbf{w}(a_k)\right]) \\ &= f(A, \bar{w}, \bar{v}) \end{aligned}$$

and depends only on the attraction probabilities of items in A and the termination probabilities $\bar{v}$. An analogous property proved useful in the design and analysis of algorithms for cascading bandits (Kveton et al., 2015a).

We evaluate the performance of a learning agent by its *expected cumulative regret* $R(n) = \mathbb{E}\left[\sum_{t=1}^{n} R(\mathbf{A}_t, \mathbf{w}_t, \mathbf{v}_t)\right]$, where $R(\mathbf{A}_t, \mathbf{w}_t, \mathbf{v}_t) = f(A^*, \mathbf{w}_t, \mathbf{v}_t) - f(\mathbf{A}_t, \mathbf{w}_t, \mathbf{v}_t)$ is the *instantaneous regret* of the agent at time t and

$$A^* = \arg\max\nolimits_{A \in \Pi_K(E)} f(A, \bar{w}, \bar{v})$$

is the *optimal list* of items, the list that maximizes the expected reward. Note that A^* is the list of K most attractive items, where the k-th most attractive item is placed at the k-th most terminating position. To simplify exposition, we assume that the optimal solution, as a set, is unique.

3.2. Learning Without Accessing Rewards

Learning in DCM bandits is difficult because the observations $\mathbf{c}_t$ are not sufficient to determine the reward $\mathbf{r}_t$. We illustrate this problem on the following example. Suppose that the agent recommends $\mathbf{A}_t = (1, 2, 3, 4)$ and observes $\mathbf{c}_t = (0, 1, 1, 0)$. This feedback can be interpreted as follows. The first explanation is that item 1 is not attractive, items 2 and 3 are, and the user does not terminate at position 3. The second explanation is that item 1 is not attractive, items 2 and 3 are, and the user terminates at position 3. In the first case, the reward is zero. In the second case, the reward is one. Since the rewards are unobserved, DCM bandits are an instance of *partial monitoring* (Section 6). However, general algorithms for partial monitoring are not suitable for DCM bandits because their number of actions is exponential in K. Therefore, we make an additional assumption that allows us to learn efficiently.

Algorithm 1 `dcmKL-UCB` for solving DCM bandits.

```
// Initialization
Observe $\mathbf{w}_0 \sim P_{\mathrm{w}}$
$\forall e \in E : \mathbf{T}_0(e) \leftarrow 1$
$\forall e \in E : \hat{\mathbf{w}}_1(e) \leftarrow \mathbf{w}_0(e)$

for all $t = 1, \dots, n$ do
  for all $e = 1, \dots, L$ do
    Compute UCB $\mathbf{U}_t(e)$ using (1)

  // Recommend and observe
  $\mathbf{A}_t \leftarrow \arg\max_{A \in \Pi_K(E)} f(A, \mathbf{U}_t, \tilde{v})$
  Recommend $\mathbf{A}_t$ and observe clicks $\mathbf{c}_t \in \{0,1\}^K$
  $\mathbf{C}_t^{\mathrm{last}} \leftarrow \max\{k \in [K] : \mathbf{c}_t(k) = 1\}$

  // Update statistics
  $\forall e \in E : \mathbf{T}_t(e) \leftarrow \mathbf{T}_{t-1}(e)$
  for all $k = 1, \dots, \min\{\mathbf{C}_t^{\mathrm{last}}, K\}$ do
    $e \leftarrow \mathbf{a}_k^t$
    $\mathbf{T}_t(e) \leftarrow \mathbf{T}_t(e) + 1$
    $\hat{\mathbf{w}}_{\mathbf{T}_t(e)}(e) \leftarrow \dfrac{\mathbf{T}_{t-1}(e)\hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e) + \mathbf{c}_t(k)}{\mathbf{T}_t(e)}$
```

The key idea in our solution is based on the following insight. Without loss of generality, suppose that the termination probabilities satisfy $\bar{v}(1) \geq \ldots \geq \bar{v}(K)$. Then $A^* = \arg\max_{A \in \Pi_K(E)} f(A, \bar{w}, \tilde{v})$ for any $\tilde{v} \in [0,1]^K$ such that $\tilde{v}(1) \geq \ldots \geq \tilde{v}(K)$. Therefore, the *termination probabilities* do not have to be learned if their *order is known*, and we assume this in the rest of the paper. This assumption is much milder than knowing the probabilities. In Section 5, we show that our algorithm performs well even when this order is misspecified.

Finally, we need one more insight. Let

$$\mathbf{C}_t^{\mathrm{last}} = \max\{k \in [K] : \mathbf{c}_t(k) = 1\}$$

denote the position of the *last click*, where $\max \emptyset = +\infty$. Then $\mathbf{w}_t(\mathbf{a}_k^t) = \mathbf{c}_t(k)$ for any $k \leq \min\{\mathbf{C}_t^{\mathrm{last}}, K\}$. This means that the first $\min\{\mathbf{C}_t^{\mathrm{last}}, K\}$ entries of $\mathbf{c}_t$ represent the observations of $\mathbf{w}_t$, which can be used to learn $\bar{w}$.

3.3. `dcmKL-UCB` Algorithm

Our proposed algorithm, `dcmKL-UCB`, is described in Algorithm 1. It belongs to the family of UCB algorithms and is motivated by KL-UCB (Garivier & Cappe, 2011). At time t, dcmKL-UCB operates in three stages. First, it computes the *upper confidence bounds (UCBs)* on the attraction probabilities of all items in E, $\mathbf{U}_t \in [0,1]^E$. The UCB of item e at time t is

$$\begin{aligned}\mathbf{U}_t(e) = \max\{q \in [w, 1] : w = \hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e)\,, \qquad (1)\\ \mathbf{T}_{t-1}(e) D_{\mathrm{KL}}(w \,\|\, q) \leq \log t + 3 \log\log t\}\,,\end{aligned}$$

where $D_{\mathrm{KL}}(p \,\|\, q)$ is the *Kullback-Leibler (KL) divergence* between Bernoulli random variables with means p and q; $\hat{\mathbf{w}}_s(e)$ is the average of s observed weights of item e; and $\mathbf{T}_t(e)$ is the number of times that item e is observed in t steps. Since $D_{\mathrm{KL}}(p \,\|\, q)$ increases in q for $q \geq p$, our UCB can be computed efficiently. After this, dcmKL-UCB selects a list of K items with largest UCBs

$$\mathbf{A}_t = \arg\max{}_{A \in \Pi_K(E)} f(A, \mathbf{U}_t, \tilde{v})$$

and recommends it, where $\tilde{v} \in [0,1]^K$ is any vector whose entries are ordered in the same way as in $\bar{v}$. The selection of $\mathbf{A}_t$ can be implemented efficiently in $O([L+K]\log K)$ time, by placing the item with the k-th largest UCB to the k-th most terminating position. Finally, after the user provides feedback $\mathbf{c}_t$, dcmKL-UCB updates its estimate of $\bar{w}(e)$ for any item e up to position $\min\left\{\mathbf{C}_t^{\mathrm{last}}, K\right\}$, as discussed in Section 3.2.

dcmKL-UCB is initialized with one sample of the attraction weight per item. Such a sample can be obtained in at most L steps as follows (Kveton et al., 2015a). At time $t \in [L]$, item t is placed at the first position. Since the first position in the DCM is always examined, $\mathbf{c}_t(1)$ is guaranteed to be a sample of the attraction weight of item t.

4. Analysis

This section is devoted to the analysis of DCM bandits. In Section 4.1, we analyze the regret of dcmKL-UCB under the assumption that all termination probabilities are identical. This simpler case illustrates the key ideas in our proofs. In Section 4.2, we derive a general upper bound on the regret of dcmKL-UCB. In Section 4.3, we derive a lower bound on the regret in DCM bandits when all termination probabilities are identical. All supplementary lemmas are proved in Appendix A.

For simplicity of exposition and without loss of generality, we assume that the attraction probabilities of items satisfy $\bar{w}(1) \geq \ldots \geq \bar{w}(L)$ and that the termination probabilities of positions satisfy $\bar{v}(1) \geq \ldots \geq \bar{v}(K)$. In this setting, the *optimal solution* is $A^* = (1, \ldots, K)$. We say that item e is *optimal* when $e \in [K]$ and that item e is *suboptimal* when $e \in E \setminus [K]$. The *gap* between the attraction probabilities of suboptimal item e and optimal item e^*,

$$\Delta_{e,e^*} = \bar{w}(e^*) - \bar{w}(e)\,,$$

characterizes the hardness of discriminating the items. We also define the *maximum attraction probability* as $p_{\max} = \bar{w}(1)$ and $\alpha = (1 - p_{\max})^{K-1}$. In practice, $p_{\max}$ tends to be small and therefore α is expected to be large, unless K is also large.

The key idea in our analysis is the reduction to cascading bandits (Kveton et al., 2015a). We define the *cascade reward* for $i \in [K]$ recommended items as

$$f_i(A, w) = 1 - \prod_{k=1}^{i} (1 - w(a_k))$$

and the corresponding *expected cumulative cascade regret* $R_i(n) = \mathbb{E}\left[\sum_{t=1}^n (f_i(A^*, \mathbf{w}_t) - f_i(\mathbf{A}_t, \mathbf{w}_t))\right]$. We bound the cascade regret of dcmKL-UCB below.

Proposition 1. *For any $i \in [K]$ and $\varepsilon > 0$, the expected n-step cascade regret of* dcmKL-UCB *is bounded as*

$$\begin{aligned}R_i(n) \leq \sum_{e=i+1}^{L} \frac{(1+\varepsilon)\Delta_{e,i}(1 + \log(1/\Delta_{e,i}))}{D_{\mathrm{KL}}(\bar{w}(e) \,\|\, \bar{w}(i))} \times\\ (\log n + 3\log\log n) + C\,,\end{aligned}$$

where $C = iL\frac{C_2(\varepsilon)}{n^{\beta(\varepsilon)}} + 7i\log\log n$, and $C_2(\varepsilon)$ and $\beta(\varepsilon)$ are defined in Garivier & Cappe (2011).

Proof. The proof is identical to that of Theorem 3 in Kveton et al. (2015a) for the following reason. Our confidence radii have the same form as those in CascadeKL-UCB; and for any $\mathbf{A}_t$ and $\mathbf{w}_t$, dcmKL-UCB is guaranteed to observe at least as many entries of $\mathbf{w}_t$ as CascadeKL-UCB. ■

To simplify the presentation of our proofs, we introduce *or* function $V : [0,1]^K \to [0,1]$, which is defined as $V(x) = 1 - \prod_{k=1}^K (1 - x_k)$. For any vectors x and y of length K, we write $x \geq y$ when $x_k \geq y_k$ for all $k \in [K]$. We denote the component-wise product of vectors x and y by $x \odot y$, and the restriction of x to $A \in \Pi_K(E)$ by $x|_A$. The latter has precedence over the former. The expected reward can be written in our new notation as $f(A, \bar{w}, \bar{v}) = V(\bar{w}|_A \odot \bar{v})$.

4.1. Equal Termination Probabilities

Our first upper bound is derived under the assumption that all terminations probabilities are the same. The main steps in our analysis are the following two lemmas, which relate our objective to a linear function.

Lemma 1. *Let $x, y \in [0,1]^K$ satisfy $x \geq y$. Then*

$$V(x) - V(y) \leq \sum_{k=1}^K x_k - \sum_{k=1}^K y_k\,.$$

Lemma 2. *Let $x, y \in [0, p_{\max}]^K$ satisfy $x \geq y$. Then*

$$\alpha \left[\sum_{k=1}^{K} x_k - \sum_{k=1}^{K} y_k\right] \leq V(x) - V(y)\,,$$

where $\alpha = (1 - p_{\max})^{K-1}$.

Now we present the main result of this section.

Theorem 1. *Let $\bar{v}(k) = \gamma$ for all $k \in [K]$ and $\varepsilon > 0$. Then the expected n-step regret of* `dcmKL-UCB` *is bounded as*

$$R(n) \leq \frac{\gamma}{\alpha} \sum_{e=K+1}^{L} \frac{(1+\varepsilon)\Delta_{e,K}(1 + \log(1/\Delta_{e,K}))}{D_{\mathrm{KL}}(\bar{w}(e) \,\|\, \bar{w}(K))} \times (\log n + 3 \log\log n) + C\,,$$

where $C = \frac{\gamma}{\alpha}\left(KL\frac{C_2(\varepsilon)}{n^{\beta(\varepsilon)}} + 7K \log\log n\right)$, and $C_2(\varepsilon)$ and $\beta(\varepsilon)$ are from Proposition 1.

Proof. Let $\mathbf{R}_t = R(\mathbf{A}_t, \mathbf{w}_t, \mathbf{v}_t)$ be the stochastic regret at time t and

$$\mathcal{H}_t = (\mathbf{A}_1, \mathbf{c}_1, \dots, \mathbf{A}_{t-1}, \mathbf{c}_{t-1}, \mathbf{A}_t)$$

be the *history* of the learning agent up to choosing list $\mathbf{A}_t$, the first $t-1$ observations and t actions. By the tower rule, we have $R(n) = \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right]\right]$, where

$$\begin{aligned}\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right] &= f(A^*, \bar{w}, \bar{v}) - f(\mathbf{A}_t, \bar{w}, \bar{v}) \\ &= V(\bar{w}|_{A^*} \odot \bar{v}) - V(\bar{w}|_{\mathbf{A}_t} \odot \bar{v})\,.\end{aligned}$$

Now note that the items in A^* can be permuted such that any optimal item in $\mathbf{A}_t$ matches the corresponding item in A^*, since $\bar{v}(k) = \gamma$ for all $k \in [K]$ and $V(x)$ is invariant to the permutation of x. Then $\bar{w}|_{A^*} \odot \bar{v} \geq \bar{w}|_{\mathbf{A}_t} \odot \bar{v}$ and we can bound $\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right]$ from above by Lemma 1. Now we apply Lemma 2 and get

$$\begin{aligned}\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right] &\leq \gamma \left[\sum_{k=1}^{K} \bar{w}(a_k^*) - \sum_{k=1}^{K} \bar{w}(\mathbf{a}_k^t)\right] \\ &\leq \frac{\gamma}{\alpha}\left[f_K(A^*, \bar{w}) - f_K(\mathbf{A}_t, \bar{w})\right]\,.\end{aligned}$$

By the definition of $R(n)$ and from the above inequality, it follows that

$$R(n) \leq \frac{\gamma}{\alpha} \sum_{t=1}^{n} \mathbb{E}\left[f_K(A^*, \bar{w}) - f_K(\mathbf{A}_t, \bar{w})\right] = \frac{\gamma}{\alpha} R_K(n)\,.$$

Finally, we bound $R_K(n)$ using Proposition 1. ■

4.2. General Upper Bound

Our second upper bound holds for any termination probabilities. Recall that we still assume that `dcmKL-UCB` knows the order of these probabilities. To prove our upper bound, we need one more supplementary lemma.

Lemma 3. *Let $x \in [0,1]^K$ and x' be the permutation of x whose entries are in decreasing order, $x'_1 \geq \ldots \geq x'_K$. Let the entries of $c \in [0,1]^K$ be in decreasing order. Then*

$$V(c \odot x') - V(c \odot x) \leq \sum_{k=1}^{K} c_k x'_k - \sum_{k=1}^{K} c_k x_k\,.$$

Now we present our most general upper bound.

Theorem 2. *Let $\bar{v}(1) \geq \ldots \geq \bar{v}(K)$ and $\varepsilon > 0$. Then the expected n-step regret of* `dcmKL-UCB` *is bounded as*

$$R(n) \leq (1+\varepsilon) \sum_{i=1}^{K} \frac{\bar{v}(i) - \bar{v}(i+1)}{\alpha} \times \sum_{e=i+1}^{L} \frac{\Delta_{e,i}(1 + \log(1/\Delta_{e,i}))}{D_{\mathrm{KL}}(\bar{w}(e) \,\|\, \bar{w}(i))} (\log n + 3 \log\log n) + C\,,$$

where $\bar{v}(K+1) = 0$, $C = \sum_{i=1}^{K} \frac{\bar{v}(i) - \bar{v}(i+1)}{\alpha}\left(iL\frac{C_2(\varepsilon)}{n^{\beta(\varepsilon)}} + 7i \log\log n\right)$, and $C_2(\varepsilon)$ and $\beta(\varepsilon)$ are from Proposition 1.

Proof. Let $\mathbf{R}_t$ and $\mathcal{H}_t$ be defined as in the proof of Theorem 1. The main challenge in this proof is that we cannot apply Lemma 1 as in the proof of Theorem 1, because we cannot guarantee that $\bar{w}|_{A^*} \odot \bar{v} \geq \bar{w}|_{\mathbf{A}_t} \odot \bar{v}$ when the termination probabilities are not identical. To overcome this problem, we rewrite $\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right]$ as

$$\begin{aligned}\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right] = {} & [V(\bar{w}|_{A^*} \odot \bar{v}) - V(\bar{w}|_{\mathbf{A}'_t} \odot \bar{v})] + \\ & [V(\bar{w}|_{\mathbf{A}'_t} \odot \bar{v}) - V(\bar{w}|_{\mathbf{A}_t} \odot \bar{v})]\,,\end{aligned}$$

where $\mathbf{A}'_t$ is the permutation of $\mathbf{A}_t$ where all items are in the decreasing order of their attraction probabilities. From the definitions of A^* and $\mathbf{A}'_t$, $\bar{w}|_{A^*} \odot \bar{v} \geq \bar{w}|_{\mathbf{A}'_t} \odot \bar{v}$, and we can apply Lemma 1 to bound the first term above. We bound the other term by Lemma 3 and get

$$\begin{aligned}\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right] &\leq \sum_{k=1}^{K} \bar{v}(k)(\bar{w}(a_k^*) - \bar{w}(\mathbf{a}_k^t)) \\ &= \sum_{i=1}^{K} [\bar{v}(i) - \bar{v}(i+1)] \sum_{k=1}^{i} (\bar{w}(a_k^*) - \bar{w}(\mathbf{a}_k^t))\,,\end{aligned}$$

where we define $\bar{v}(K+1) = 0$. Now we bound each term $\sum_{k=1}^{i}(\bar{w}(a_k^*) - \bar{w}(\mathbf{a}_k^t))$ by Lemma 2, and get from the definitions of $R(n)$ and $R_i(n)$ that

$$R(n) \leq \sum_{i=1}^{K} \frac{\bar{v}(i) - \bar{v}(i+1)}{\alpha} R_i(n)\,.$$

Finally, we bound each $R_i(n)$ using Proposition 1. ■

Note that when $\bar{v}(k) = \gamma$ for all $k \in [K]$, the above upper bound reduces to that in Theorem 1.

4.3. Lower Bound

Our lower bound is derived on the following class of problems. The ground set are L items $E = [L]$ and K of these items are optimal, $A^* \subseteq \Pi_K(E)$. The attraction probabilities of items are defined as

$$\bar{w}(e) = \begin{cases} p & e \in A^* \\ p - \Delta & \text{otherwise}\,, \end{cases}$$

where p is a common attraction probability of the optimal items, and Δ is the gap between the attraction probabilities of the optimal and suboptimal items. The number of positions is K and their termination probabilities are identical, $\bar{v}(k) = \gamma$ for all positions $k \in [K]$. We denote an instance of our problem by $B_{\text{LB}}(L, A^*, p, \Delta, \gamma)$; and parameterize it by L, A^*, p, Δ, and γ. The key step in the proof of our lower bound is the following lemma.

Lemma 4. *Let $x, y \in [0,1]^K$ satisfy $x \geq y$. Let $\gamma \in [0,1]$. Then $V(\gamma x) - V(\gamma y) \geq \gamma[V(x) - V(y)]$.*

Our lower bound is derived for consistent algorithms as in Lai & Robbins (1985). We say that the algorithm is *consistent* if for any DCM bandit, any suboptimal item e, and any $\alpha > 0$, $\mathbb{E}[\mathbf{T}_n(e)] = o(n^\alpha)$; where $\mathbf{T}_n(e)$ is the number of times that item e is observed in n steps, the item is placed at position $\mathbf{C}_t^{\text{last}}$ or higher for all $t \leq n$. Our lower bound is derived below.

Theorem 3. *For any DCM bandit B_{LB}, the regret of any consistent algorithm is bounded from below as*

$$\liminf_{n \to \infty} \frac{R(n)}{\log n} \geq \gamma\alpha \frac{(L-K)\Delta}{D_{\text{KL}}(p - \Delta \,\|\, p)}\,.$$

Proof. The key idea of the proof is to reduce our problem to a cascading bandit. By the tower rule and Lemma 4, the n-step regret in DCM bandits is bounded from below as

$$R(n) \geq \gamma \mathbb{E}\left[\sum_{t=1}^{n} (f_K(A^*, \mathbf{w}_t) - f_K(\mathbf{A}_t, \mathbf{w}_t))\right]\,.$$

Moreover, by the tower rule and Lemma 2, we can bound the n-step regret in cascading bandits from below as

$$\begin{aligned} R(n) &\geq \gamma\alpha \mathbb{E}\left[\sum_{t=1}^{n}\left(\sum_{k=1}^{K} \mathbf{w}_t(a_k^*) - \sum_{k=1}^{K} \mathbf{w}_t(\mathbf{a}_k^t)\right)\right] \\ &\geq \gamma\alpha\Delta \sum_{e=K+1}^{L} \mathbb{E}[\mathbf{T}_n(e)]\,, \end{aligned}$$

where the last step follows from the facts that the expected regret for recommending any suboptimal item e is Δ, and that the number of times that this item is recommended in n steps is bounded from below by $\mathbf{T}_n(e)$. Finally, for any consistent algorithm and item e,

$$\liminf_{n \to \infty} \frac{\mathbb{E}[\mathbf{T}_n(e)]}{\log n} \geq \frac{\Delta}{D_{\text{KL}}(p - \Delta \,\|\, p)}\,,$$

by the same argument as in Lai & Robbins (1985). Otherwise, the algorithm would not be able to distinguish some instances of B_{LB} where item e is optimal, and would have $\Omega(n^\alpha)$ regret for some $\alpha > 0$ on these problems. Finally, we chain the above two inequalities and this completes our proof. ■

4.4. Discussion

We derive two gap-dependent upper bounds on the n-step regret of dcmKL-UCB, under the assumptions that all termination probabilities are identical (Theorem 1) and that their order is known (Theorem 2). Both bounds are logarithmic in n, linear in the number of items L, and decrease as the number of recommended items K increases. The bound in Theorem 1 grows linearly with γ, the common termination probability at all positions. Since smaller γ result in more clicks, we show that the regret decreases with more clicks. This is in line with our expectation that it is easier to learn from more feedback.

The upper bound in Theorem 1 is tight on problem $B_{\text{LB}}(L, A^* = [K], p = 1/K, \Delta, \gamma)$ from Section 4.3. In this problem, $1/\alpha \leq e$ and $1/e \leq \alpha$ when $p = 1/K$; and then the upper bound in Theorem 1 and the lower bound in Theorem 3 reduce to

$$O\left(\gamma(L-K)\tfrac{\Delta(1+\log(1/\Delta))}{D_{\text{KL}}(p-\Delta \,\|\, p)} \log n\right)\,,$$
$$\Omega\left(\gamma(L-K)\tfrac{\Delta}{D_{\text{KL}}(p-\Delta \,\|\, p)} \log n\right)\,,$$

respectively. The bounds match up to $\log(1/\Delta)$.

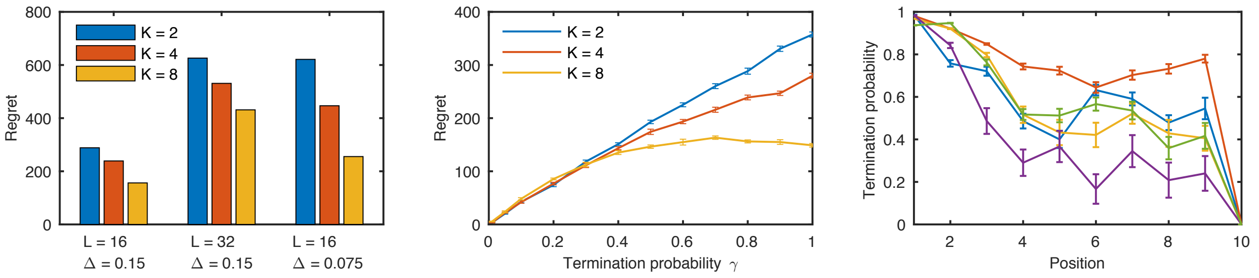
5. Experiments

We conduct three experiments. In Section 5.1, we validate that the regret of dcmKL-UCB scales as suggested by Theorem 1. In Section 5.2, we compare dcmKL-UCB to multiple baselines. Finally, in Section 5.3, we evaluate dcmKL-UCB on a real-world dataset.

5.1. Regret Bounds

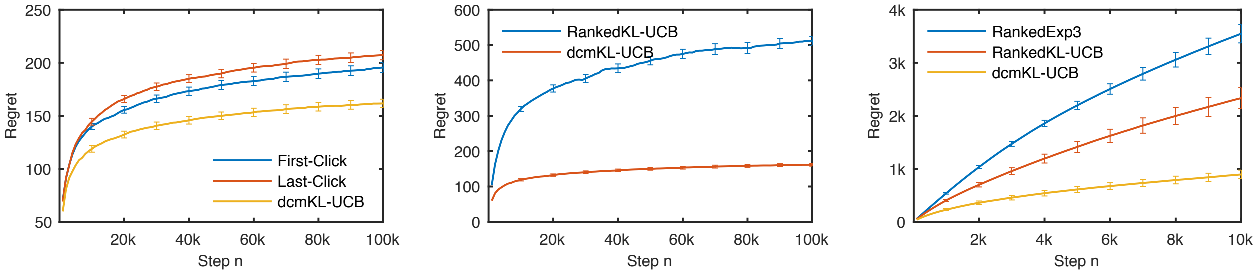
In the first experiment, we validate the behavior of our upper bound in Theorem 1. We experiment with the class of problems $B_{\text{LB}}(L, A^* = [K], p = 0.2, \Delta, \gamma)$, which is presented in Section 4.3. We vary L, K, Δ, and γ; and report the regret of dcmKL-UCB in $n = 10^5$ steps.

Figure 2a shows the n-step regret of dcmKL-UCB as a function of L, K, and Δ for $\gamma = 0.8$. We observe three trends.



(a) (b) (c)

Figure 2. **a**. The n-step regret of dcmKL-UCB in $n = 10^5$ steps on the problem in Section 5.1. All results are averaged over 20 runs. **b**. The n-step regret of dcmKL-UCB as a function of the common termination probability γ and K. **c**. The termination probabilities in the DCMs of 5 most frequent queries in the Yandex dataset.



(a) (b) (c)

Figure 3. **a**. The n-step regret of dcmKL-UCB and two heuristics on the problem in Section 5.2. **b**. The n-step regret of dcmKL-UCB and RankedKL-UCB on the same problem. **c**. The n-step regret of dcmKL-UCB, RankedKL-UCB, and RankedExp3 in the Yandex dataset.

First, the regret increases when the number of items L increases. Second, the regret decreases when the number of recommended items K increases. These dependencies are suggested by our $O(L - K)$ upper bound. Finally, we observe that the regret increases when Δ decreases.

Figure 2b shows the n-step regret of dcmKL-UCB as a function of γ and K, for $L = 16$ and $\Delta = 0.15$. We observe that the regret grows linearly with γ, as suggested by Theorem 1, when $p < 1/K$. This trend is less prominent when $p > 1/K$. We believe that this is because the upper bound in Theorem 1 is loose when $\alpha = (1 - p)^{K-1}$ is small, and this happens when p is large.

5.2. First Click, Last Click, and Ranked Bandits

In the second experiment, we compare dcmKL-UCB to two single-click heuristics and ranked bandits (Section 6). The heuristics are motivated by CascadeKL-UCB, which learns from a single click (Kveton et al., 2015a). The first heuristic is dcmKL-UCB where the feedback $\mathbf{c}_t$ is altered such that it contains only the first click. This method can be viewed as a conservative extension of CascadeKL-UCB to multiple clicks and we call it First-Click. The second heuristic is dcmKL-UCB where the feedback $\mathbf{c}_t$ is modified such that it contains only the last click. This method was suggested by Kveton et al. (2015a) and we call it Last-Click. We also compare dcmKL-UCB to RankedKL-UCB, which is a ranked bandit with KL-UCB. The base algorithm in RankedKL-UCB is the same as in dcmKL-UCB, and therefore we believe that this comparison is fair. All methods are evaluated on problem $B_{\text{LB}}(L = 16, A^* = [4], p = 0.2, \Delta = 0.15, \gamma = 0.5)$ from Section 4.3.

The regret of dcmKL-UCB, First-Click, and Last-Click is shown in Figure 3a. The regret of dcmKL-UCB is clearly the lowest among all compared methods. We conclude that dcmKL-UCB outperforms both baselines because it does not discard or misinterpret any feedback in $\mathbf{c}_t$.

The regret of RankedKL-UCB and dcmKL-UCB is reported in Figure 3b. We observe that the regret of RankedKL-UCB is three times higher than that of dcmKL-UCB. Note that $K = 4$. Therefore, this validates our hypothesis that dcmKL-UCB can learn K times faster than a ranked bandit, because the regret of dcmKL-UCB is $O(L - K)$ (Section 4.4) while the regret in ranked bandits is $O(KL)$ (Section 6).

5.3. Real-World Experiment

In the last experiment, we evaluate dcmKL-UCB on the *Yandex* dataset (Yandex), a search log of 35M search sessions. In each query, the user is presented 10 web pages and may click on multiple pages. We experiment with 20 most frequent queries from our dataset and estimate one DCM per query, as in Guo et al. (2009b). We compare `dcmKL-UCB` to `RankedKL-UCB` (Section 5.2) and `RankedExp3`. The latter is a ranked bandit with `Exp3`, which can learn correlations among recommended positions. We parameterize `Exp3` as suggested in Auer et al. (1995). All compared algorithms assume that higher ranked positions are more valuable, as this would be expected in practice. This is not necessarily true in our DCMs (Figure 2c). However, this assumption is quite reasonable because most of our DCMs have the following structure. The first position is the most terminating and the most attractive item tends to be much more attractive than the other items. Therefore, any solution that puts the most attractive item at the first position performs well. All methods are evaluated by their average regret over all 20 queries, with 5 runs per query.

Our results are reported in Figure 3c and we observe that `dcmKL-UCB` outperforms both ranked bandits. At $n = 10$k, for instance, the regret of `dcmKL-UCB` is at least two times lower than that of our best baseline. This validates our hypothesis that `dcmKL-UCB` can learn much faster than ranked bandits (Section 5.2), even in practical problems where the model of the world is likely to be misspecified.

6. Related Work

Our work is closely related to *cascading bandits* (Kveton et al., 2015a; Combes et al., 2015a). Cascading bandits are an online learning variant of the cascade model of user behavior in web search (Craswell et al., 2008). Kveton et al. (2015a) proposed a learning algorithm for these problems, `CascadeKL-UCB`; bounded its regret; and proved a matching lower bound up to logarithmic factors. The main limitation of cascading bandits is that they cannot learn from multiple clicks. DCM bandits are a generalization of cascading bandits that allows multiple clicks.

Ranked bandits are a popular approach in learning to rank (Radlinski et al., 2008; Slivkins et al., 2013). The key idea in ranked bandits is to model each position in the recommended list as a separate bandit problem, which is solved by some *base bandit algorithm*. In general, the algorithms for ranked bandits learn $(1 - 1/e)$ approximate solutions and their regret is $O(KL)$, where L is the number of items and K is the number of recommended items. We compare `dcmKL-UCB` to ranked bandits in Section 5.

DCM bandits can be viewed as a partial monitoring problem where the reward, the satisfaction of the user, is unobserved. Unfortunately, general algorithms for partial monitoring (Agrawal et al., 1989; Bartok et al., 2012; Bartok & Szepesvari, 2012; Bartok et al., 2014) are not suitable for DCM bandits because their number of actions is exponential in the number of recommended items K. Lin et al. (2014) and Kveton et al. (2015b) proposed algorithms for combinatorial partial monitoring. The feedback models in these algorithms are different from ours and therefore they cannot solve our problem.

The feasible set in DCM bandits is combinatorial, any list of K items out of L is feasible, and the learning agent observes the weights of individual items. This setting is similar to stochastic combinatorial semi-bandits, which are often studied with linear reward functions (Gai et al., 2012; Chen et al., 2013; Kveton et al., 2014; 2015c; Wen et al., 2015; Combes et al., 2015b). The differences in our work are that the reward function is non-linear and that the feedback model is less than semi-bandit, because the learning agent does not observe the attraction weights of all recommended items.

7. Conclusions

In this paper, we study a learning variant of the dependent click model, a popular click model in web search (Chuklin et al., 2015). We propose a practical online learning algorithm for solving it, `dcmKL-UCB`, and prove gap-dependent upper bounds on its regret. The design and analysis of our algorithm are challenging because the learning agent does not observe rewards. Therefore, we propose an additional assumption that allows us to learn efficiently. Our analysis relies on a novel reduction to a single-click model, which still preserves the multi-click character of our model. We evaluate `dcmKL-UCB` on several problems and observe that it performs well even when our modeling assumptions are violated.

We leave open several questions of interest. For instance, the upper bound in Theorem 1 is linear in the common termination probability γ. However, Figure 2b shows that the regret of `dcmKL-UCB` is not linear in γ for $p > 1/K$. This indicates that our upper bounds can be improved. We also believe that our approach can be contextualized, along the lines of Zong et al. (2016); and extended to more complex cascading models, such as influence propagation in social networks, along the lines of Wen et al. (2016).

To the best of our knowledge, this paper presents the first practical and regret-optimal online algorithm for learning to rank with multiple clicks in a cascade-like click model. We believe that our work opens the door to further developments in other, perhaps more complex and complete, instances of learning to rank with multiple clicks.

Acknowledgments

This work was supported by the Alberta Innovates Technology Futures and NSERC.

References

Agichtein, Eugene, Brill, Eric, and Dumais, Susan. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th Annual International ACM SIGIR Conference*, pp. 19–26, 2006.

Agrawal, Rajeev, Teneketzis, Demosthenis, and Anantharam, Venkatachalam. Asymptotically efficient adaptive allocation schemes for controlled i.i.d. processes: Finite parameter space. *IEEE Transactions on Automatic Control*, 34(3):258–267, 1989.

Auer, Peter, Cesa-Bianchi, Nicolo, Freund, Yoav, and Schapire, Robert. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pp. 322–331, 1995.

Bartok, Gabor and Szepesvari, Csaba. Partial monitoring with side information. In *Proceedings of the 23rd International Conference on Algorithmic Learning Theory*, pp. 305–319, 2012.

Bartok, Gabor, Zolghadr, Navid, and Szepesvari, Csaba. An adaptive algorithm for finite stochastic partial monitoring. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

Bartok, Gabor, Foster, Dean, Pal, David, Rakhlin, Alexander, and Szepesvari, Csaba. Partial monitoring - classification, regret bounds, and algorithms. *Mathematics of Operations Research*, 39(4):967–997, 2014.

Becker, Hila, Meek, Christopher, and Chickering, David Maxwell. Modeling contextual factors of click rates. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pp. 1310–1315, 2007.

Chapelle, Olivier and Zhang, Ya. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th International Conference on World Wide Web*, pp. 1–10, 2009.

Chen, Wei, Wang, Yajun, and Yuan, Yang. Combinatorial multi-armed bandit: General framework, results and applications. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 151–159, 2013.

Chuklin, Aleksandr, Markov, Ilya, and de Rijke, Maarten. *Click Models for Web Search*. Morgan & Claypool Publishers, 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*, 2015a.

Combes, Richard, Talebi, Mohammad Sadegh, Proutiere, Alexandre, and Lelarge, Marc. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems 28*, pp. 2107–2115, 2015b.

Craswell, Nick, Zoeter, Onno, Taylor, Michael, and Ramsey, Bill. An experimental comparison of click position-bias models. In *Proceedings of the 1st ACM International Conference on Web Search and Data Mining*, pp. 87–94, 2008.

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*, 20(5):1466–1478, 2012.

Garivier, Aurelien and Cappe, Olivier. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceeding of the 24th Annual Conference on Learning Theory*, pp. 359–376, 2011.

Guo, Fan, Liu, Chao, Kannan, Anitha, Minka, Tom, Taylor, Michael, Wang, Yi Min, and Faloutsos, Christos. Click chain model in web search. In *Proceedings of the 18th International Conference on World Wide Web*, pp. 11–20, 2009a.

Guo, Fan, Liu, Chao, and Wang, Yi Min. Efficient multiple-click models in web search. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*, pp. 124–131, 2009b.

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*, pp. 420–429, 2014.

Kveton, Branislav, Szepesvari, Csaba, Wen, Zheng, and Ashkan, Azin. Cascading bandits: Learning to rank in the cascade model. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015a.

Kveton, Branislav, Wen, Zheng, Ashkan, Azin, and Szepesvari, Csaba. Combinatorial cascading bandits. In *Advances in Neural Information Processing Systems 28*, pp. 1450–1458, 2015b.

Kveton, Branislav, Wen, Zheng, Ashkan, Azin, and Szepesvari, Csaba. Tight regret bounds for stochastic combinatorial semi-bandits. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 2015c.

Lai, T. L. and Robbins, Herbert. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

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*, pp. 901–909, 2014.

Radlinski, Filip and Joachims, Thorsten. Query chains: Learning to rank from implicit feedback. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 239–248, 2005.

Radlinski, Filip, Kleinberg, Robert, and Joachims, Thorsten. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 784–791, 2008.

Richardson, Matthew, Dominowska, Ewa, and Ragno, Robert. Predicting clicks: Estimating the click-through rate for new ads. In *Proceedings of the 16th International Conference on World Wide Web*, pp. 521–530, 2007.

Slivkins, Aleksandrs, Radlinski, Filip, and Gollapudi, Sreenivas. Ranked bandits in metric spaces: Learning diverse rankings over large document collections. *Journal of Machine Learning Research*, 14(1):399–436, 2013.

Wen, Zheng, Kveton, Branislav, and Ashkan, Azin. Efficient learning in large-scale combinatorial semi-bandits. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Wen, Zheng, Kveton, Branislav, and Valko, Michal. Influence maximization with semi-bandit feedback. *CoRR*, abs/1605.06593, 2016.

Yandex. Yandex personalized web search challenge. https://www.kaggle.com/c/yandex-personalized-web-search-challenge, 2013.

Zong, Shi, Ni, Hao, Sung, Kenny, Ke, Nan Rosemary, Wen, Zheng, and Kveton, Branislav. Cascading bandits for large-scale recommendation problems. In *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence*, 2016.