Fatigue-Aware Bandits for Dependent Click Models

Junyu Cao,[1*] **Wei Sun,**[2] **Zuo-Jun (Max) Shen,**[1] **Markus Ettl**[2]
[1]University of California, Berkeley, California 94720
[2]IBM Research, Yorktown Height, New York 10591

Abstract

As recommender systems send a massive amount of content to keep users engaged, users may experience fatigue which is contributed by 1) an overexposure to irrelevant content, 2) boredom from seeing too many similar recommendations. To address this problem, we consider an online learning setting where a platform learns a policy to recommend content that takes user fatigue into account. We propose an extension of the Dependent Click Model (DCM) to describe users' behavior. We stipulate that for each piece of content, its attractiveness to a user depends on its intrinsic relevance and a discount factor which measures how many similar contents have been shown. Users view the recommended content sequentially and click on the ones that they find attractive. Users may leave the platform at any time, and the probability of exiting is higher when they do not like the content. Based on user's feedback, the platform learns the relevance of the underlying content as well as the discounting effect due to content fatigue. We refer to this learning task as "fatigue-aware DCM Bandit" problem. We consider two learning scenarios depending on whether the discounting effect is known. For each scenario, we propose a learning algorithm which simultaneously explores and exploits, and characterize its regret bound.

1 Introduction

Recommender systems increasingly influence how users discover content. Some well-known examples include Facebook's News Feed, movie recommendations on Netflix, Spotify's Discover Weekly playlists, etc. To compete for users' attention and time, platforms push out a vast amount of content when users browse their websites or mobile apps. While a user sifts through the proliferation of content, her experience could be adversely influenced by: 1) *marketing fatigue* which occurs due to an overexposure to irrelevant content, and 2) *content fatigue* which refers to the boredom from seeing too many similar recommendations.

Motivated by this phenomenon, we consider an online learning setting with a "fatigue-aware" platform, i.e., it learns a policy to select a sequence of recommendations for a user, while being conscious that both the choices of content and their placement could influence the experience. We propose a variant of the Dependent Click Model (DCM), where a user can click on multiple items[1] that she finds attractive. For each piece of content, its attractiveness depends on its intrinsic relevance and a discount factor that measures how many similar recommendations have already been shown to this user. The user may leave the platform at any time, and the probability of exiting is higher when she does not like the content, reflecting the presence of marketing fatigue. Meanwhile, showing too much relevant but similar content could also reduce its attractiveness due to content fatigue and drive bored users to abandon the platform. The platform whose objective is to maximize the total number of clicks needs to learn users' latent preferences in order to determine the optimal sequence of recommendations, based on users' feedback. We refer to this online learning task which the platform faces as *fatigue-aware DCM bandits*.

Our contribution of our work is fourfold. Firstly, we propose a novel model which captures the effects of marketing fatigue and content fatigue on users' behavior. In the original DCM model (Guo, Liu, and Wang 2009; Katariya et al. 2016b), users can only exit the platform upon clicking on a recommendation. In reality, users may leave at any time, and they are more likely to exit when they are not engaged with the recommendations. We extend the DCM model to capture marketing fatigue by incorporating exiting behavior after both non-clicks and clicks. To reflect content fatigue, we incorporate a discount factor into DCM such that it penalizes repetitive content of similar types and promotes diversity in its recommendations. Secondly, even in the offline setting where all the information is known, the optimization problem to select a sequence of recommendation which is central to the learning task is combinatorial in nature without a straightforward efficient algorithm. We propose a polynomial-time algorithm for this problem. Thirdly, for the online setting, we first consider a scenario where the discount factor is known (e.g., when it can be estimated from historical data). We propose a learning algorithm and quantify the regret bound. Lastly, we consider a more general scenario where both the discount factor and intrinsic relevance

[*] Correspondence to: Junyu Cao (jycao@berkeley.edu)

[1] We use "content" and "items" interchangeably in this work.

of items need to be learned. This is a significantly more challenging setting to analyze because 1) we only observe partial feedback on item's attractiveness which depends on both the discount factor and the relevance; 2) the discount factor is dependent on the content that we recommend. For this scenario, we also establish a regret bound for the algorithm that we propose.

Our paper is organized as follows. In Section 2, we review related work. In Section 3, we introduce the fatigue-aware DCM model and provide an offline algorithm to find the optimal sequence when all parameters are known. In Section 4 and 5, we introduce our learning problems for two scenarios depending on whether the discount factor is known and analyze the regret associated with the proposed algorithms. In Section 6, we perform several numerical experiments to evaluate the performance of our algorithms.

2 Related Literature

Cascade Model (Chuklin, Markov, and Rijke 2015; Craswell et al. 2008) is a popular model that describes user's single-click behavior. Several variants of this model have been considered in the bandit literature (Combes et al. 2015; Kveton et al. 2015a; 2015b). One of its limitations is that the positions of the items do not influence the reward since a user is expected to browse the list of recommendations until she clicks on the content that she likes. DCM (Guo, Liu, and Wang 2009) generalizes Cascade Model to allow multiple clicks by incorporating a parameter to indicate the probability that a user resumes browsing after clicking on a recommendation. On the other hand, if a user does not click, she views the next item with certainty unless the sequence runs out. Katariya et al. 2016b analyzed one type of DCM bandit settings. However, the reward in Katariya et al. 2016b does not exactly correspond to the number of clicks. In our DCM bandit setting, the rewards exactly correspond to the number of users' clicks, which is an ubiquitous measure on recommenders' effectiveness. In addition, as our model allows users to exit at any time, the position of the items matters as users may exit the platform early if they do not like what has been shown initially.

Alternative multi-click models include Position-Based Model (PBM) (Richardson, Dominowska, and Ragno 2007). In PBM, the click probability of an item is a product of a static position-based examination probability and the item's intrinsic relevance score. The following papers have investigated PBM bandits. More specifically, in Lagrée, Vernade, and Cappé 2016, the position-based examination probability is assumed known, whereas this quantity has to be learned together with item attractiveness in Komiyama, Honda, and Takeda 2017. In particular, Komiyama, Honda, and Takeda 2017 derive a parameter-dependent regret bound, which assumes that the gap between parameters is strictly positive. While our model also includes a position-dependent parameter which discounts the attractiveness of an item, this quantity does not merely depend on the position. The discount factor also depends on the order and the content types of the recommendations. For our setting, we derive a parameter-independent regret bound.

Our work also bears some resemblance to the stochastic rank-1 bandits (Katariya et al. 2016a; 2017), where a row arm and a column arm are pulled simultaneously in each around, and the reward corresponds to the product of the two values. In our setting, item attractiveness is a product of the discount factor and item's intrinsic relevance. However, in rank-1 bandits, one is only interested in determining the particular pair that yields the highest reward, whereas we need to learn and rank all items, which is significantly harder.

An important topic of recommender systems is fatigue control (Kapoor et al. 2015; Ma, Liu, and Shen 2016). One way to manage content fatigue is to introduce diversity in recommendations (Radlinski, Kleinberg, and Joachims 2008; Ziegler et al. 2005). The linear submodular function has been used to tackle the diversification problem in bandit setting (Yu, Fang, and Tao 2016; Yue and Guestrin 2011). Warlop, Lazaric, and Mary 2018 propose a reinforcement learning framework that uses a linear reward structure which captures the effect of the recent recommendations on user's preferences. However, instead of recommending specific items as in our work, Warlop, Lazaric, and Mary 2018 restrict to recommending genres or content types to a user. While Cao and Sun 2019; Wang and Tulabandhula 2019 have also studied marketing fatigue in an online setting, their setting only captures a single click, whereas our model allows multiple clicks and incorporates the effect of content fatigue in addition to marketing fatigue.

3 Problem Formulation

In this section, we formally introduce the fatigue-aware DCM model and present an algorithm to determine the optimal sequence of recommendations in an offline setting where all underlying parameters are known.

3.1 Setting

Suppose there are N available items (e.g., songs, videos, articles) for the platform to choose from, denoted as $[N]$. Each item belongs to one of the K types (e.g., genres, categories, topics). Let $\mathbf{S} = (S_1, \cdots, S_m)$ denote a sequence of recommended items. For each item i, u_i denotes its intrinsic relevance to a user, and $C(i) = j$ if item i belongs to type j. We define $h_i(\mathbf{S})$ as the number of items with type $C(i)$ shown before item i. Similar to Warlop, Lazaric, and Mary 2018, we model the impact of showing too many similar contents as a discount factor on items' attractiveness. More precisely, define the attractiveness of item i as $z_i(\mathbf{S}) := f(h_i(\mathbf{S}))u_i$, where $f(\cdot)$ represents the discount factor on users' preferences due to content fatigue, which depends on how many items of the same type have been shown. $f(\cdot)$ is assumed to be a decreasing function. Without loss of generality, we assume $f(0) = 1$.

Given a list of recommendations $\mathbf{S} = (S_1, \cdots, S_m)$, a user examines it sequentially, starting from S_1. The user clicks on item i when she finds it attractive. The platform receives a reward for every click. DCM models the multi-click behavior by incorporating a parameter g which is the probability that the user will see more recommendations after a click. In case of no click or skip, unlike in the original DCM

(Guo, Liu, and Wang 2009; Katariya et al. 2016b) where a user examines the next item with probability 1, we use q to denote the probability that the user will resume browsing. As users are more likely to stay on the platform and continue browsing after consuming a piece of good content[2], we let $q \leq g$. The interaction between a user and recommended content is illustrated in Fig 1.

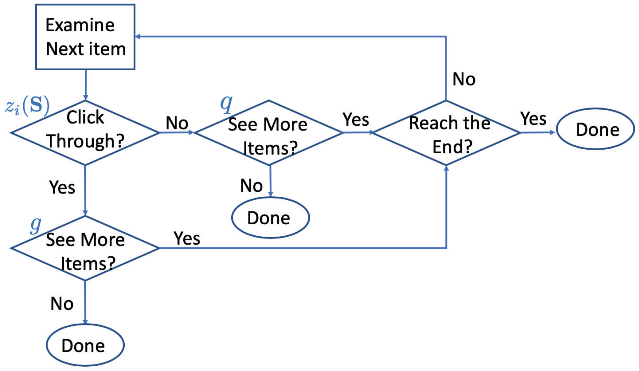

Figure 1: User behavior in a fatigue-aware DCM.

The probability of clicking on item i from a sequence $\mathbf{S}$, denoted as $\mathbb{P}_i(\mathbf{S})$, depends on its position in the list, its own content, as well as the content of other items shown previously. Formally,

$$\mathbb{P}_i(\mathbf{S}) = \begin{cases} u_i, & \text{if } i \in S_1 \\ \prod_{k=1}^{l-1} \left(g z_{I(k)}(\mathbf{S}) + q(1 - z_{I(k)}(\mathbf{S})) \right) z_i(\mathbf{S}) & \text{if } i \in S_l \end{cases}$$

where $I(\cdot)$ denotes the index function, i.e., $I(k) = i$ if and only if $S_k = \{i\}$. When i is the first item of the sequence, the probability of clicking is simply u_i, which is its intrinsic relevance. For the remainder of the sequence, the probability of clicking on i as the l^{th} item is the joint probability that 1) the user finds item i attractive after taking content fatigue into account, $z_i(\mathbf{S})$; 2) she remains on the platform after examining the first $l-1$ items, $\prod_{k=1}^{l-1} \left(g z_{I(k)}(\mathbf{S}) + (1 - z_{I(k)}(\mathbf{S}))q \right)$, which accounts for both clicking and skipping behavior.

3.2 Platform's optimization problem

The platform's objective is to maximize the total number of clicks by optimizing the sequence of recommended items. We use $R(\mathbf{S})$ to denote the platform's expected reward, i.e., $E[R(\mathbf{S})] = \sum_{i \in [N]} \mathbb{P}_i(\mathbf{S})$. Thus, the platform's optimization problem can be written as

$$\max_{\mathbf{S}} \quad E[R(\mathbf{S})] \qquad (1)$$
$$s.t. \quad S_i \cap S_j = \emptyset, \forall i \neq j.$$

The constraint requires no duplicated items in the recommendations. We define $\mathbf{S}^*$ as the optimal sequence of content, and R^* as the corresponding maximum reward. While this problem is combinatorial in nature, Theorem 3.1 shows that this problem is polynomial-time solvable and the optimal sequence $\mathbf{S}^*$ can be identified by Algorithm 1.

Algorithm 1: Determine the optimal sequence $\mathbf{S}^*$ to the offline combinatorial problem

```
for $i = 1 : K$ do
    Sort $u_j$ for $C(j) = i$ in an decreasing order $o(\cdot)$
      where $o(j) = r$ if item $j$ is at the $(r+1)^{th}$
      position;
    Set $\lambda_j = u_j f(o(j))$;
end
Sort $\lambda_j$ for all $j = 1 : N$ in an decreasing order $o'(\cdot)$
  where $o'(r) = j$ if item $j$ is at the $r^{th}$ position;
Set $\mathbf{S} = (o'(1), o'(2), \cdots, o'(N))$.
```

Theorem 3.1 *Algorithm 1 finds the optimal sequence* $\mathbf{S}^*$.

Due to the space limit, we present the proof outline in the main paper and the detailed proofs are included in the Supplementary Material.

Proof outline: We prove the result by contradiction. If the optimal sequence $\mathbf{S}^*$ is not ordered by Algorithm 1, then there exists a neighboring pair $I(i)$ and $I(i+1)$ such that either 1) $I(i)$ and $I(i+1)$ belong to the same type and $u_{I(i)} < u_{I(i+1)}$ or 2) $I(i)$ and $I(i+1)$ belong to different types and $f(h_{I(i)}(\mathbf{S}^*))u_{I(i)} < f(h_{I(i+1)}(\mathbf{S}^*))u_{I(i+1)}$. We then show that swapping items $I(i)$ and $I(i+1)$ increases the expected reward, which is a contradiction to $\mathbf{S}^*$ being optimal. □

For each category, Algorithm 1 first ranks items based on their intrinsic relevance. If an item has the $(r+1)^{th}$ largest relevance score within a category, we will then multiply the score with $f(r)$ to compute its attractiveness. Next, the algorithm ranks items across all categories in their decreasing attractiveness. The complexity of Algorithm 1 is $O(N \log N)$. We also observe that the optimal sequence is completely determined by items' intrinsic relevance $\mathbf{u}$ and the discount factor f, and is independent of the resuming probabilities g and q. In other words, the platform only needs to learn $\mathbf{u}$ and f for the online learning task. Nevertheless, the resuming probabilities influence the learning speed as we will investigate in the following sections.

4 Learning with Known Discount Factor f

In the previous section, we assume all the parameters are known to the platform. It is natural to ask what the platform should do in the absence of such knowledge. Beginning from this section, we will present learning algorithms to the fatigue-aware DCM bandit problem and characterize the corresponding regret bound. We first investigate a scenario where the discounting effect caused by content fatigue is known (e.g., it could be estimated from historical data), and the platform needs to learn the item's intrinsic relevance

[2] We provide some evidence of this phenomenon based on the estimates from a real dataset, with more details in Section 6.

$\mathbf{u}$ to determine the optimal sequence. We want to emphasize the differences between this setting and other learning-to-rank problems (Combes et al. 2015; Kveton et al. 2015a; 2015b). Firstly, in our setting, only partial feedback to a list is observed as the user might leave before viewing all the recommendations. Secondly, as the order of the content influences their attractiveness, the click rate of an item is non-stationary. Thirdly, in many previous learning-to-rank related problems, the position-biased examine probability is usually assumed to be fixed for each position. However, in our setting, this examine probability depends on the content of previously shown recommendations.

Assume users arrive at time $t = 1, \cdots, T$. For each user t, the platform determines a sequence of items $\tilde{\mathbf{S}}^t$. We evaluate a learning algorithm by its expected cumulative rewards under policy π, or equivalently, its expected cumulative regret which is defined as $Regret_\pi(T;\mathbf{u}) = E_\pi\left[\sum_{t=1}^{T} R(\mathbf{S}^*, \mathbf{u}) - R(\tilde{\mathbf{S}}^t, \mathbf{u})\right]$.

4.1 Algorithm FA-DCM-P

Our strategy, which we call Fatigue-Aware DCM with Partial information (FA-DCM-P), is a UCB-based policy. We first define an unbiased estimator for $\mathbf{u}$ and its upper confidence bound.

We define the pair (i, j) as an event that item j is the $(i+1)^{th}$ recommendation from the category $C(j)$. That is, i recommendations from the same category $C(j)$ have been shown before item j. We define $\mathcal{T}_{ij}$ as the epoch for the event (i, j), i.e., $t \in \mathcal{T}_{ij}$ if and only if user t observes the event (i, j). Let $T_{ij}(t)$ denote the number of times that the event (i, j) occurs by time t, i.e., $T_{ij}(t) = |\mathcal{T}_{ij}(t)|$. Define $T_j(t) = \sum_i T_{ij}(t)$. Let z_{ij}^t be the binary feedback for user t observing the event (i, j), where $z_{ij}^t = 1$ indicates that the user clicks, and $z_{ij}^t = 0$ means no click or skip. Define $\hat{u}_{j,t}$ as the estimator for u_j, and $u_{j,t}^{UCB}$ as its upper confidence bound, i.e.,

$$\hat{u}_{j,t} = \frac{1}{T_j(t)} \sum_i \sum_{t \in \mathcal{T}_{ij}} \frac{z_{ij}^t}{f_i}, \qquad u_{j,t}^{UCB} = \hat{u}_{j,t} + \sqrt{2\frac{\log t}{T_j(t)}}. \tag{2}$$

In Algorithm FA-DCM-P, for a user arriving at time t, we use $\mathbf{u}_{t-1}^{UCB}$ to calculate the current optimal sequence of recommendations. Define w_t as the last item examined by user t, which occurs when one of the following feedback is observed: 1) the user exits after clicking on w_t; 2) the user views w_t without clicking and exits; 3) the sequence runs out. When the last examined item w_t is shown, we update $T_j(t)$ and the upper confidence bound to $\mathbf{u}_t^{UCB}$ based on the feedback.

4.2 Regret analysis for Algorithm FA-DCM-P

In this section, we quantify the regret bound for our proposed algorithm. We first derive the following lemmas which will be used in the regret analysis.

Algorithm 2: [FA-DCM-P] An algorithm for fatigue-aware DCM bandit when f is known

```
Initialization: Set $u_{i,0}^{UCB} = 1$ and $T_i(0) = 0$ for all $i \in [N]$; $t = 1$;
while $t < T$ do
    Compute $\tilde{\mathbf{S}}^t = \arg\max_{\mathbf{S}} \quad E[R(\mathbf{S}, \mathbf{u}_{t-1}^{UCB})]$ according to Theorem 3.1;
    Offer sequence $\tilde{\mathbf{S}}^t$, observe the user's feedback;
    update $T_i(t)$; update $\mathbf{u}_t^{UCB}$ according to Equation (2); $t = t + 1$;
end
```

Lemma 4.1 *For any t and $j \in [N]$ we have*

$$P\left(u_{j,t}^{UCB} - \sqrt{8\frac{\log t}{T_j(t)}} < u_j < u_{j,t}^{UCB}\right) \geq 1 - \frac{2}{t^4}.$$

Lemma 4.2 *Assume $\mathbf{S}^*$ is the optimal sequence of messages. Under the condition that $0 \leq \mathbf{u} \leq \mathbf{u}'$, we have*

$$E[R(\mathbf{S}^*, \mathbf{u}')] \geq E[R(\mathbf{S}^*, \mathbf{u})].$$

Proof of Lemma 4.2. We prove this theorem by induction. For the optimal sequence $\mathbf{S}^* = (S_1, S_2, \cdots, S_N)$, since $u'_{I(N)} \geq u_{I(N)}$, we have $E[R(S_N, \mathbf{u}')] \geq E[R(S_N, \mathbf{u})]$. Assume the inequality

$$E[R((S_{k+1}, \cdots, S_N), \mathbf{u}')] \geq E[R((S_{k+1}, \cdots, S_N), \mathbf{u})]$$

holds, and now we prove that $E[R((S_k, \cdots, S_N), \mathbf{u}')] \geq E[R((S_k, \cdots, S_N), \mathbf{u})]$. Since

$$\begin{aligned}
&E[R((S_k, \cdots, S_N), \mathbf{u}')] \\
&= u'_{I(k)} + (gu'_{I(k)} + q(1 - u'_{I(k)}))E[R((S_k, \cdots, S_N), \mathbf{u}')] \\
&= u'_{I(k)} + ((g - q)u'_{I(k)} + q)E[R((S_k, \cdots, S_N), \mathbf{u}')] \\
&\geq u_{I(k)} + ((g - q)u_{I(k)} + q)E[R((S_k, \cdots, S_N), \mathbf{u})] \\
&= E[R((S_k, \cdots, S_N), \mathbf{u})].
\end{aligned}$$

Therefore, we reach the desired result. ■

Theorem 4.3 *The regret of Algorithm FA-DCM-P during time T is bounded by*

$$Regret_\pi(T) \leq C\left(\frac{1 - g^N}{1 - g}\right)^{3/2} \sqrt{NT \log T}.$$

for some constant C.

Proof outline: Define event $A_{i,t} = \{u_{i,t}^{UCB} - \sqrt{8\frac{\log t}{T_i(t)}} < u < u_{i,t}^{UCB}\}$ and $E_t = \bigcap_{i=1}^{N} A_{i,t}$. On the "large probability" event E_t, with Lemma 4.2, we obtain $E[R(\tilde{\mathbf{S}}^t, \mathbf{u})] \leq E[R(\mathbf{S}^*, \mathbf{u})] \leq E[R(\mathbf{S}^*, \mathbf{u}^{UCB})] \leq E[R(\tilde{\mathbf{S}}^t, \mathbf{u}^{UCB})]$. It implies that $E[R(\mathbf{S}^*, \mathbf{u})] - E[R(\tilde{\mathbf{S}}^t, \mathbf{u})] \leq E[R(\tilde{\mathbf{S}}^t, \mathbf{u}^{UCB})] - E[R(\tilde{\mathbf{S}}^t, \mathbf{u})]$. Let $I_t(\cdot)$ denote the index function for user t. We then show that the cumulative difference on event E_t can be bounded from above by

$$E_\pi \left[\sum_{t=1}^{T} \sum_{i=1}^{|\tilde{\mathbf{S}}^t|} \prod_{k=1}^{i-1} ((g-q) f(h_{I_t(k)}(\tilde{\mathbf{S}}^t)) u^{UCB}_{I_t(k),t} + q) \right.$$

$$\left. f(h_{I_t(i)}(\tilde{\mathbf{S}}^t)) u^{UCB}_{I_t(i),t} 1(E_t) \right] - E_\pi \left[\sum_{t=1}^{T} \sum_{i=1}^{|\tilde{\mathbf{S}}^t|} \prod_{k=1}^{i-1} ((g-q) \right.$$

$$\left. f(h_{I_t(k)}(\tilde{\mathbf{S}}^t)) u_{I_t(k)} + q) f(h_{I_t(i)}(\tilde{\mathbf{S}}^t)) u_{I_t(i)} 1(E_t) \right]$$

$$\leq \frac{1-g^N}{1-g} E_\pi \left[\sum_{t=1}^{T} \sum_{j=1}^{N} \kappa_{j,t} (u^{UCB}_{j,t} - u_{j,t}) 1(E_t) \right],$$

where $\kappa_{j,t} = \prod_{k=1}^{l_t(j)-1} ((g-q) f(h_{I_t(k)}(\tilde{\mathbf{S}}^t)) u_{I_t(k)} + q)$. It denotes the probability that user t observes item j and $l_t(i)$ specifies the position of item i in $\tilde{\mathbf{S}}^t$. We then show that the regret term can be further bounded by $\frac{1-g^N}{1-g} \sqrt{2 \log T} \sum_{i \in [N]} \sqrt{E_\pi[T_i(T)]}$. Since $\sum_{i \in [N]} E_\pi[T_i(T)] \leq \frac{1-g^N}{1-g} T$, we can bound it by $C_1 (\frac{1-g^N}{1-g})^{3/2} \sqrt{NT \log T}$. To complete the proof, we show that on the "small probability" event E_t^c, the regret can also be bounded. □

The following results can be shown under two special cases of the DCM setting.

Corollary 4.4 *When $g = q = 1$, the regret can be bounded by*

$$Regret_\pi(T) \leq CN^2 \sqrt{T \log T}.$$

Corollary 4.5 *When at most L items can be recommended to a single user, the regret can be bounded by*

$$Regret_\pi(T) \leq CL^{3/2} \sqrt{NT \log T}.$$

Corollary 4.4 characterizes the regret bound under a special setting where users never exit the platform and browse all the recommendations. In theorem 4.3, we do not limit the length of the sequence as online recommenders continuously send content to users to keep them on the platforms. If at most L items will be recommended, the correponding regret is shown in Corollary 4.5. The detailed proofs can be found in the Supplementary Material.

5 Learning with Unknown Discount Factor f

In this section, we consider a more general scenario where the discount factor f is also unknown and needs to be learned, in conjunction with learning items' intrinsic relevance scores $\mathbf{u}$. Before delving into our approach, we first discuss a few alternative approaches and their trade-offs. A straightforward approach towards this problem is to treat each combination of $f(i)$ and u_j as an arm whose expected reward is $z_{ij} := f(i) u_j$. However, it is not clear how to solve the offline combinatorial problem. An alternative approach is to view the problem as a generalized linear bandit which can be solved by GLM-UCB (Li, Lu, and Zhou 2017). Taking logarithm, we have $\log(z_{ij}) = \log(f(i)) + \log(u_j)$, which is a linear model. However, this approach is problematic for the following reasons: a) When u_j and $f(i)$ go to 0, $\log(u_j)$ and $\log(f(i))$ can go to negative infinity, which implies that the parameter space is unbounded; b) In each step, GLM-UCB needs to compute the maximum likelihood estimators, which is computationally costly. We now present our approach, a UCB-based algorithm for this learning task which circumvents the limitations associated with the alternatived approaches discussed earlier.

5.1 Algorithm FA-DCM

Throughout this section, we define M as a threshold such that after showing M items from the same category, the discounting effect on item attractiveness due to content fatigue stays the same. That is, $f(r) = f(r+1)$ for $r \geq M$. Note that the maximum possible value for M could be N.

Following the notations introduced in Section 4.1, we define the unbiased estimator for f_i and u_j as

$$\hat{f}_i^t = \frac{1}{\hat{T}_i(t)} \sum_j \sum_{r \in \mathcal{T}_{ij}(t)} \frac{z_{ij}^r}{\hat{\mu}_j}, \quad \text{and} \quad \hat{u}_j^t = \frac{\sum_{r \in \mathcal{T}_{0j}(t)} z_{0j}^r}{T_{0j}(t)}, \tag{3}$$

where $\hat{T}_i(t) = \sum_j T_{ij}(t)$, the subscript in T_{0j} and z_{0j}^r represents the event $(0, j)$, that is, item j is the first message from category $C(j)$ to be shown. The corresponding upper confidence bounds at time t are defined as

$$f_t^{UCB}(i) = \hat{f}_i^t + \Delta_i^t, \quad \text{and} \quad u_{j,t}^{UCB} = \hat{u}_{j,t} + \sqrt{2 \frac{\log t}{T_{0j}(t)}}, \tag{4}$$

where

$$\Delta_i^t = \sum_j \sum_{r \in \mathcal{T}_{ij}(t)} \frac{z_{ij}^r}{\hat{T}_i(t)} \frac{1}{(\hat{\mu}_j^t)^2} \left(1 - \frac{1}{\hat{\mu}_j^t} \sqrt{\frac{\log t}{T_{0j}(t)}}\right)$$

$$1\left(\hat{\mu}_j^t \geq \sqrt{\frac{\log t}{T_{0j}(t)}}\right) \sqrt{\frac{\log t}{T_{0j}(t)}} + \sqrt{\frac{\log t}{\hat{T}_i(t)}}.$$

Δ_i^t consists of two parts: the first part refers to the exploration term related to μ_j for all $j \in [N]$, and the second part is the exploration term with respect to $f(i)$.

We propose a learning algorithm, which we refer to as FA-DCM for the scenario when both the function f and $\mathbf{u}$ need to be learned. Define $\theta^{UCB} = (f^{UCB}, \mathbf{u}^{UCB})$ and $\theta = (f, \mathbf{u})$. At time t, we recommend a sequence $\tilde{\mathbf{S}}^t$ based on θ_{t-1}^{UCB}. Suppose there exists an item i which we have not collected sufficient feedback such that $T_i(t) < \alpha T^{2/3}$ where α is a tuning parameter. This item will then be inserted to the beginning of the sequence $\tilde{\mathbf{S}}^t$ to guarantee that it will be examined. This procedure ensures that we obtain a reliable estimate for u_i, since the discount factor $f(0) = 1$ for the first item. Once the feedback of the user at time t is observed, we then update f_t^{UCB} and $\mathbf{u}_t^{UCB}$. Finally, it may be possible that some estimated values of $f_t^{UCB}(i)$ violate the decreasing property of f, i.e., $f_t^{UCB}(i) > f_t^{UCB}(i-1)$, then we need to correct them to enforce the property.

Algorithm 3: [FA-DCM] An algorithm for fatigue-aware DCM when f is unknown

```
Initialization: Set $u_{j,0}^{UCB} = 1$ and $f^{UCB}(i) = 1$ for all $j \in [N]$ and $i \in [M]$ ; $t = 1$;
while $t < T$ do
    Compute $\tilde{\mathbf{S}}^t = argmax_{\mathbf{S}} \quad E[R(\mathbf{S}, \theta_{t-1}^{UCB})]$ according to Theorem 3.1;
    if there exists $i \in [N]$ such that $T_i(t) < \alpha T^{2/3}$ then
        Insert item $i$ to the beginning of of $\tilde{\mathbf{S}}^t$ as the first item;
    end
    Offer sequence $\tilde{\mathbf{S}}^t$, observe the user's feedback;
    Update $u_{j,t}^{UCB}$ for all $j$ and $f_t^{UCB}(i)$ for all $i$ according to Equation (4);
    for $i = 1 : M$ do
        if $f_t^{UCB}(i) > f_t^{UCB}(i-1)$ then
            $f_t^{UCB}(i) = f_t^{UCB}(i-1)$;
        end
    end
    $t = t + 1$;
end
```

5.2 Regret analysis for Algorithm FA-DCM

Define $V_i^t = \frac{1}{\hat{T}_i} \sum_j \sum_{t \in \mathcal{T}_{ij}(t)} \frac{z_{ij}^t}{\mu_j}$. We first have the following lemma to bound the difference between V_i^t and f_i.

Lemma 5.1 *For any t, $P(|V_i^t - f(i)| \geq \sqrt{\frac{\log t}{\hat{T}_i(t)}}) \leq \frac{2}{t^2}$.*

Lemma 5.2 *For any $1 \leq i \leq M$, we have $\sum_{t=1}^{T} P(|\hat{f}_t(i) - f(i)| \geq \Delta_i^t) \leq CN$, for some constant C, where $\hat{f}_t(i)$ is defined in Equation* (3).

Proof outline: First we note that

$$P\left(|\hat{f}_t(i) - f(i)| \geq \Delta_i^t\right)$$

$$\leq P\left(\left|\frac{1}{\hat{T}_i(t)} \sum_j \sum_{r \in \mathcal{T}_{ij}} z_{ij}^r \left(\frac{1}{\hat{\mu}_j} - \frac{1}{\mu_j}\right)\right| \geq \right.$$

$$\sum_j \sum_{r \in \hat{\mathcal{T}}_{ij}} \frac{z_{ij}^r}{\hat{T}_i(t)} \frac{1}{\hat{\mu}_j^2} \left(1 - \frac{1}{\hat{\mu}_j} \sqrt{\frac{\log t}{T_{0j}(t)}}\right)$$

$$\left. 1\left(\hat{\mu}_j \geq \sqrt{\frac{\log t}{T_{0j}(t)}}\right) \sqrt{\frac{\log t}{T_{0j}(t)}}\right) \tag{5}$$

$$+ P\left(\left|\frac{1}{\hat{T}_i(t)} \sum_j \sum_{r \in \mathcal{T}_{ij}} \frac{z_{ij}^r}{\mu_j} - f(i)\right| \geq \sqrt{\frac{\log t}{T_i(t)}}\right) \tag{6}$$

For Equation (5), we can bound it above by

$$\sum_j \frac{2}{t^2} + \exp\left(-\frac{1}{2} T_{0j}(t) \mu_j^2\right) + P\left(\sqrt{\frac{\log t}{T_{0j}(t)}} < \frac{\mu_j}{2}\right).$$

For Equation (6), using Lemma 5.1, we have

$$P\left(\left|\frac{1}{\hat{T}_i(t)} \sum_j \sum_{r \in \mathcal{T}_{ij}} \frac{z_{ij}^r}{\mu_j} - f(i)\right| \geq \sqrt{\frac{\ln t}{T_i(t)}}\right) \leq \frac{2}{t^2}.$$

It implies that

$$\sum_{t=1}^{T} P(|\hat{f}_t(i) - f(i)| \geq \Delta_i^t) \leq CN$$

for some constant C. □

Lemma 5.3 *Assume $\mathbf{S}^*$ is the optimal sequence of messages. Under the condition that $0 \leq \mathbf{u} \leq \mathbf{u}'$ and $0 \leq f \leq f'$, we have*

$$E[U(\mathbf{S}^*, f', \mathbf{u}')] \geq E[U(\mathbf{S}^*, f, \mathbf{u})].$$

Proof: Similar to the proof of Lemma 4.2, we can prove it by induction. □

Theorem 5.4 *The regret of Algorithm FA-DCM during time T is bounded by*

$$Regret_\pi(T) \leq C \left(\frac{1 - g^N}{1 - g}\right)^2 \sqrt{NT^{4/3} \log T}.$$

Proof outline: Define the following events, $A_{j,t} = \{u_{j,t}^{UCB} - \sqrt{8 \frac{\log t}{T_j(t)}} < u_j < u_{j,t}^{UCB}\}$, $B_{i,t} = \{f_t^{UCB}(i) - 2\Delta_i^t < f(i) < f_t^{UCB}(i)\}$, and $E_t = \bigcap_{j=1}^{N} (A_{j,t}) \bigcap_{i=1}^{M} (B_{i,t})$. Let $\tilde{f}_t^{UCB}(i) = \min_{j \leq i} f_t^{UCB}(j)$, $\tilde{B}_{i,t} = \left\{\tilde{f}_t^{UCB}(i) - 2\Delta_i^t < f(i) < \tilde{f}_t^{UCB}(i)\right\}$ and $\tilde{E}_t = \bigcap_{j=1}^{N} (A_{j,t}) \bigcap_{i=1}^{M} (\tilde{B}_{i,t})$. Note that if E_t holds, $\tilde{E}_t$ also holds. We consider time t is in the exploration phase, denoted as $t \in \mathcal{E}$, if there exists $j \in [N]$ such that $T_{0j}(t) < \alpha t^{2/3}$. Similar to the proof for Theorem 4.3, the regret quantity $E[\sum_{t \notin \mathcal{E}} R(\mathbf{S}^*, \mathbf{u}) - R(\tilde{\mathbf{S}}^t, \mathbf{u})]$ on event $\tilde{E}_t$ for $t \notin \mathcal{E}$ can be first bounded from above by

$$\frac{1 - g^N}{1 - g} \left(E_\pi \left[\sum_{t \notin \mathcal{E}} \sum_{j=1}^{N} \kappa_{j,t} (u_{j,t}^{UCB} - u_j) 1(\tilde{E}_t)\right]\right.$$

$$\left. + E_\pi \left[\sum_{t \notin \mathcal{E}} \sum_{i=1}^{M} \tilde{\kappa}_{i,t} (\tilde{f}_t^{UCB}(i) - f(i)) 1(\tilde{E}_t)\right]\right),$$

where $\tilde{\kappa}_{i,t} = \sum_{\tilde{l}_t(i)} \prod_{k=1}^{\tilde{l}_t(i)-1} ((g - q) f(h_{I_t(k)}(\tilde{\mathbf{S}}^t)) u_{I_t(k)} + q)$ denotes the probability of observing an item as the i^{th} recommendation within its own category. The regret term can be further bounded by $C \left(\frac{1-g^N}{1-g}\right)^2 \sqrt{NT^{4/3} \log T}$. Finally using Lemma 5.2, we can bound the "small probability" event E_t^c and the cumulative regret on $t \in \mathcal{E}$. □

6 Numerical Experiments

In this section, we perform four sets of numerical experiments to evaluate the performance of our online learning algorithms. In the first two experiments, we investigate the robustness of Algorithm FA-DCM-P and FA-DCM respectively. In the last two experiments, we compare our algorithm with a benchmark using simulated and real datasets respectively.

6.1 Robustness study

Experiment I: With a known discount factor In this experiment, we investigate the robustness of Algorithm FA-DCM-P, when the discount factor f is known to the platform. We consider a setting with three categories, and each contains 10 items. Items' intrinsic relevance $\mathbf{u}$ is uniformly generated from $[0, 0.5]$. The known discount factor is set to $f(r) = \exp(-0.1 * (r - 1))$. We fix the resuming probability after non-clicks to $q = 0.7$, and compare three cases by varying the resuming probabilities after clicks, i.e, Case 1: $g = 0.95$; Case 2: $g = 0.85$; Case 3: $g = 0.75$. For each case, we run 20 independent simulations.

Experiment II: With an unknown discount factor We assume both f and $\mathbf{u}$ are unknown in this experiment, and study the robustness of Algorithm FA-DCM. There are three categories which contain 10 items each. We consider three cases, where we use $q = 0.7$ as in Experiment I and vary p and the discount factor f. More precisely, Case 4: $g = 0.85$, $f(r) = \exp(-0.1(r - 1))$; Case 5: $g = 0.85$, $f(r) = \exp(-0.15(r-1))$; Case 6: $g = 0.75$, $f(r) = \exp(-0.1(r-1))$.

Result: The left plot in Fig 2 shows the average regret for Algorithm FA-DCM-P as the solid line and its 95% sampled confidence interval as the shaded area, generated from 20 independent simulations for each case. Average values for the regret at $T = 10,000$ are 307.52, 277.77 and 265.73 for Case 1, 2 and 3 respectively. The result shows that the regret decreases as g decreases, which is consistent with the regret bound shown in Theorem 4.3.

The right plot in Fig 2 shows the results for Algorithm FA-DCM, where the average regret at $T = 10,000$ is 1237.82, 1178.24, and 991.97 for Case 4, 5 and 6 respectively. Comparing Case 4 and 5, we find that when the discount factor increases, the regret decreases. Meanwhile, comparing Case 4 and 6, it shows that the regret increases with g, which is consistent with the conclusion in Theorem 5.4. Comparing Case 2 and 4, as well as Case 3 and 6, the simulation results show that the regret is much larger when f also needs to be learned.

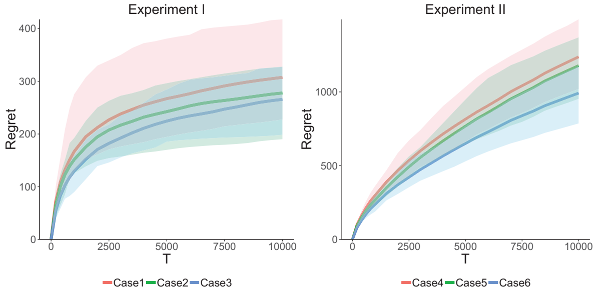

Figure 2: Simulation results for Experiments I and II

6.2 Comparison with a benchmark algorithm

Since there is no other algorithm addressing our specific setting, we choose the explore-then-exploit algorithm as a benchmark for Algorithm FA-DCM. During the exploration time steps, items are randomly selected. For the first t time steps, we ensure that there are $\beta \log t$ periods for exploration where β is a tuning parameter. That is, if the number of exploration periods before t is less than $\beta \log t$, then we choose t as the exploration period where items are randomly selected. During the exploitation period, we use the empirical mean of $\mathbf{u}$ and f to find the optimal sequence. In both Experiment III and IV, we run 20 independent simulations to compare the performance of our algorithm against the benchmark.

Experiment III: With simulated data We set parameters $g = 0.75$, $q = 0.7$ and $f(r) = \exp(-0.1(r - 1))$. Items' intrinsic relevance $\mathbf{u}$ is uniformly generated from $[0, 0.5]$.

Experiment IV: With real data The data is from Taobao[3], which contains 26 million ad display and click logs from 1,140,000 randomly sampled users from the website of Taobao for 8 days (5/6/2017-5/13/2017). To fit the data into our framework, we use long period of inactivity from the last interaction with the website as a proxy for exiting the platform. Based on the data, we estimate that the average probability to resume browsing after non-clicks is 0.823, i.e., $q = 0.823$. Meanwhile, the probability to resume browsing after clicking is higher, i.e., $g = 0.843$. For the discount factor, we use $f(r) = \exp(-0.1(r-1))$. In this experiment, we consider five categories of items, and select the top 20 selling products from each category. We estimate the click probabilities of these 100 products, and use them as the groundtruth values for u_i.

Result: Fig 3 compares the performance of our algorithm and the benchmark based on 20 independent simulations. Our algorithm outperforms the benchmark in both experiments, highlighting the benefits of simultaneous exploration and exploitation. In particular, in Experiment III which is shown as the left plot, the average regret for the benchmark is 1279.88 at T = 10,000, which is 29.02% higher than that of Algorithm FA-DCM. Meanwhile, in Experiment IV as shown as the right figure, the average regret for Algorithm FA-DCM is 910.87, while the regret for the benchmark is 1519.20 at $T = 100,000$.

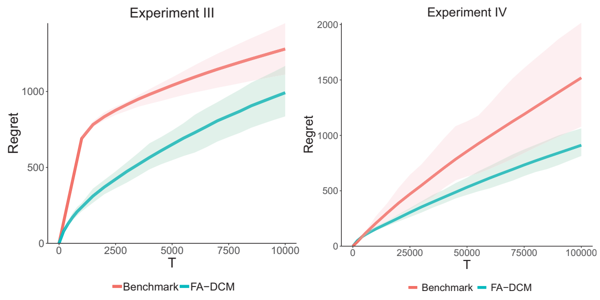

Figure 3: Simulation results for Experiments III and IV

[3] https://tianchi.aliyun.com/datalab/dataSet.html?spm=5176.100073.0.0.14d53ea7Rleuc9\&dataId=56

7 Conclusion

In this work, we investigated a fatigue-aware DCM bandit problem for an online recommender. The setting takes into account of user's fatigue which is triggered by an overexposure to irrelevant content, as well as the boredom one might experience from seeing too much similar content. Depending on whether the discount factor f is known to the platform, we proposed two online learning algorithms and quantified its regert bound for the DCM bandit setting. We further showed the regret bounds for two learning algorithms and used numerical experiments to illustrate the performance.

There are several interesting future directions. One natural extension is to consider the contextual version of this model by incorporating item features and user attributes, such that the system is capable of producing personalized recommendations. Another direction for future work is to approach the fatigue-aware DCM bandits via Thompson Sampling, although the regret analysis remains a challenging problem. Last but not least, as users' preferences often change with time, it would be interesting to incorporate non-stationary item intrinsic relevance (u_i) into the model.

References

Cao, J., and Sun, W. 2019. Dynamic learning of sequential choice bandit problem under marketing fatigue. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*.

Chuklin, A.; Markov, I.; and Rijke, M. d. 2015. Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 7(3):1–115.

Combes, R.; Magureanu, S.; Proutiere, A.; and Laroche, C. 2015. Learning to rank: Regret lower bounds and efficient algorithms. *ACM SIGMETRICS Performance Evaluation Review* 43(1):231–244.

Craswell, N.; Zoeter, O.; Taylor, M.; and Ramsey, B. 2008. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*, 87–94. ACM.

Guo, F.; Liu, C.; and Wang, Y. M. 2009. Efficient multiple-click models in web search. In *Proceedings of the second acm international conference on web search and data mining*, 124–131. ACM.

Kapoor, K.; Subbian, K.; Srivastava, J.; and Schrater, P. 2015. Just in time recommendations: Modeling the dynamics of boredom in activity streams. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, 233–242. ACM.

Katariya, S.; Kveton, B.; Szepesvari, C.; Vernade, C.; and Wen, Z. 2016a. Stochastic rank-1 bandits. *arXiv preprint arXiv:1608.03023*.

Katariya, S.; Kveton, B.; Szepesvari, C.; and Wen, Z. 2016b. Dcm bandits: Learning to rank with multiple clicks. In *International Conference on Machine Learning*, 1215–1224.

Katariya, S.; Kveton, B.; Szepesvári, C.; Vernade, C.; and Wen, Z. 2017. Bernoulli rank-1 bandits for click feedback. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2001–2007. AAAI Press.

Komiyama, J.; Honda, J.; and Takeda, A. 2017. Position-based multiple-play bandit problem with unknown position bias. In *Advances in Neural Information Processing Systems*, 4998–5008.

Kveton, B.; Szepesvari, C.; Wen, Z.; and Ashkan, A. 2015a. Cascading bandits: Learning to rank in the cascade model. In *International Conference on Machine Learning*, 767–776.

Kveton, B.; Wen, Z.; Ashkan, A.; and Szepesvari, C. 2015b. Combinatorial cascading bandits. In *Advances in Neural Information Processing Systems*, 1450–1458.

Lagrée, P.; Vernade, C.; and Cappé, O. 2016. Multiple-play bandits in the position-based model. In *Advances in Neural Information Processing Systems*, 1597–1605.

Li, L.; Lu, Y.; and Zhou, D. 2017. Provably optimal algorithms for generalized linear contextual bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2071–2080. JMLR. org.

Ma, H.; Liu, X.; and Shen, Z. 2016. User fatigue in online news recommendation. In *Proceedings of the 25th International Conference on World Wide Web*, 1363–1372. International World Wide Web Conferences Steering Committee.

Radlinski, F.; Kleinberg, R.; and Joachims, T. 2008. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, 784–791. ACM.

Richardson, M.; Dominowska, E.; and Ragno, R. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*, 521–530. ACM.

Wang, Y., and Tulabandhula, T. 2019. Thompson sampling for a fatigue-aware online recommendation system. *arXiv preprint arXiv:1901.07734*.

Warlop, R.; Lazaric, A.; and Mary, J. 2018. Fighting boredom in recommender systems with linear reinforcement learning. In *Advances in Neural Information Processing Systems*, 1757–1768.

Yu, B.; Fang, M.; and Tao, D. 2016. Linear submodular bandits with a knapsack constraint. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Yue, Y., and Guestrin, C. 2011. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems*, 2483–2491.

Ziegler, C.-N.; McNee, S. M.; Konstan, J. A.; and Lausen, G. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, 22–32. ACM.