Contextual Dependent Click Bandit Algorithm for Web Recommendation

Weiwen Liu[1], Shuai Li[1], and Shengyu Zhang[1,2(✉)]

[1] The Chinese University of Hong Kong, Sha Tin, Hong Kong
[2] Tencent, Shenzhen, China
{wwliu,shuaili,syzhang}@cse.cuhk.edu.hk

Abstract. In recommendation systems, it has been an increasing emphasis on recommending potentially novel and interesting items in addition to currently confirmed attractive ones. In this paper, we propose a contextual bandit algorithm for web page recommendation in the dependent click model (DCM), which takes user and web page features into consideration and automatically balances between exploration and exploitation. In addition, unlike many previous contextual bandit algorithms which assume that the click through rate is a linear function of features, we enhance the representability by adopting the generalized linear models, which include both linear and logistic regressions and have exhibited stronger performance in many binary-reward applications. We prove an upper bound of $\tilde{O}(d\sqrt{n})$ on the regret of the proposed algorithm. Experiments are conducted on both synthetic and real-world data, and the results demonstrate significant advantages of our algorithm.

1 Introduction

Given a search query, a web page recommendation algorithm recommends a list of related web pages based on a certain model of past user behavior and page information [1]. An online learning algorithm for personalized recommender systems aims at learning user preferences and incorporating the user feedback at each time step, while maintaining a high Click-Through Rate (CTR) over a long period of time. Earlier recommendation algorithms mostly focus on recommending the currently confirmed attractive items, and put less emphasis on the potentially valuable items in the future, e.g., the Logistic Regression (LR) [2] and the Factorizations Machines (FM) [3]. It was observed that such algorithms usually lead to suboptimal recommendations in a long term [4]. Besides, though accuracy is a typical target for recommendation, the diversity and the long-term user satisfactory of a recommender system have shown more and more importance [1]. Therefore, special attention should be paid to a balance between exploiting immediate yet suboptimal rewards (exploitation) and exploring uncertain but potentially interesting items which may produce large benefits later (exploration).

Multi-armed bandit (MAB) is a general framework of sequential decision problems, in which a balance between exploration and exploitation is needed [5].

L. Wang and D. Zhu (Eds.): COCOON 2018, LNCS 10976, pp. 39–50, 2018.
https://doi.org/10.1007/978-3-319-94776-1_4

In the basic stochastic setting, we have a number of arms each with an unknown reward distribution. At each time step, we need to select one of them, receiving the reward randomly drawn from the corresponding distribution. The goal is to maximize the total reward over the time, or equivalently, to minimize the regret, which is the difference between our cumulative reward and the reward of always pulling the best arm. Numerous algorithms have been proposed for MAB and they have been successfully applied in many scenarios, such as personalized recommendation [6], clinical trials [7], etc.

The Cascade Model (CM) is a widely used click model in which the recommended web pages are listed in a sequence and the user examines the list from top to bottom until she finds a satisfactory one [8]. This model is particularly suitable for characterizing the user browsing behavior on mobile devices. A number of bandit algorithms were developed and have exhibited prominent effectiveness in cascade model [9–11]. One limit of the model is its assumption that the user clicks at most one of the recommended items, and a natural extension to allowing multiple clicks is the *dependent click model* (DCM), where the user may click more than one items before finding a satisfactory one [12].

In the DCM bandit setting, at each time step t, the learning agent displays an ordered list of K items out of L ground items to the user. The user examines the items in the displayed order and clicks on the *attracted* items. After an item is clicked, the user may either be satisfied and leave, or unsatisfied and proceed to the next item. The user leaves if all K items have been examined, regardless of whether the user has found any satisfactory item or not. If the user leaves with satisfaction, then the learning agent receives a reward of 1; otherwise the reward is 0. However, this reward is not observed by the learning agent, as the agent cannot distinguish between the user leaving with satisfaction or leaving because she has exhausted all items. All the feedback the learning agent receives is the clicking pattern such as 0100110000, in which case the learning agent knows that the user is attracted by the 2nd, 5th and 6th items, but not by the 1st, 3rd and 4th items. However, whether the user is attracted by the rest (the 7th and beyond) remains unknown to the learning agent.

In many modern personalized news/apps/ads recommendation systems, certain features of users are available through registration or historical behaviors, which can be exploited to provide more accurate recommendations [1]. In the bandit setting, these features are usually called the *context*, modeled as a d-dimensional vector that contains information of users or items. In previous studies on contextual bandit in the cascade model, the attraction weight is assumed to be the inner product of the vector of the contextual vector and a fixed but unknown vector θ [6,11,13], i.e. a linear function of the contextual vector (thereby the name *linear bandit*). However, the reward function in real-world applications can be complicated and hardly confined to being linear. With an increasing amount of historical data, stronger models may be preferred for better representability. Besides, logistic regression (LR) has exhibited empirical improvements over the linear model in news recommendation [14]. In this paper,

we go beyond the linear reward model and consider the more general *exponential family distributions*, which include LR as a special case.

Our work has four main contributions. First, we incorporate contextual information into DCM bandit model, and strengthen the linear model by including exponential family distributions. Second, we present a computationally efficient version of our algorithm which may be valuable for practical use. Third, we prove an upper bound of $\tilde{O}(d\sqrt{n})$ on the regret. Fourth, experiments are conducted on both synthetic and real world data, which demonstrate the substantial advantage of our algorithm compared to the typical LR algorithm and the one without utilizing contextual information.

2 Problem Formulation

In this paper, we consider the contextual DCM bandit problem with the generalized linear payoff for list recommendation. Let n be the total number of time steps. Suppose that we have a set $E = \{1, \ldots, L\} = [L]$ of ground items. At each time step t, the learning agent receives a user query. Combining the user query and each arm i gives a contextual vector $x_{i,t} \in \mathbb{R}$ known to the learning agent, whose action is to recommend an ordered list $\mathbf{A}_t = (\mathbf{a}_1^t, \ldots, \mathbf{a}_K^t)$ of K distinct items from E to the user.[1] We say that such an action has length K, and denote by $\Pi_K(E)$ the feasible action set of all ordered lists of K distinct items from E. The user checks the list of items one by one from top to bottom. For each item a, the user is *attracted* with probability $\bar{w}_t(a) \in [0, 1]$, and we will use $\mathbf{w}_t(a) \in \{0, 1\}$ to denote the *attraction weight*, a Bernoulli random variable with mean $\bar{w}_t(a)$ indicating whether the user is attracted by a or not. Denote by $\mathbf{w}_t \in \{0, 1\}^E$ the random vector of these indicators, and by P_w the distribution of $\mathbf{w}_t$. We assume that the attraction vectors are independent across time steps and items, namely $\{\mathbf{w}_t\}_{t=1}^n$ are i.i.d. drawn from a probability distribution P_w.

If the user is attracted by the k-th item a_k in the recommended list, i.e. $\mathbf{w}_t(a_k) = 1$, then she clicks it and examines the item. The user may be satisfied and leave, which happens with probability $\bar{v}_t(k)$ and then the learning agent receives a reward of 1. The user may also find the item unsatisfactory (which happens with probability $1 - \bar{v}_t(k)$, and then continues to check the next item. If all items have been checked and the user has not found any satisfactory item, then the user leaves and the learning agent receives reward 0. The *termination weight* $\mathbf{v}_t(k) \in \{0, 1\}$ is the Bernoulli random variable with mean $\bar{v}_t(k)$. We denote by $\mathbf{v}_t \in \{0, 1\}^K$ the random vector of the termination weights, by P_v its distribution, and assume $\{\mathbf{v}_t\}_{t=1}^n$ to be i.i.d. drawn from P_v.

The above process defines a random $\{0, 1\}$ reward, but note that this reward is not revealed to the learning agent, as the user just leaves after checking some items and does not report whether she finds the item she wants. Indeed, the search engine does not even know when the user leaves. All the feedback that the search engine receives is a sequence of k click indicators $(\mathbf{w}'_1, \ldots, \mathbf{w}'_K)$. Note that $\mathbf{w}'_i$ may not be the same as $\mathbf{w}_i$ as. For example, if the sequence is 0100110000, it

[1] Here and throughout the paper, we use bold letters for random variables.

may be the case that the user leaves at the sixth item with satisfaction. Another case is that the user checks all items without finding anyone satisfactory, but has to leave at the end. This feedback is too limited to admit any good learning algorithm. Therefore, we adopt the same assumption as in [15] that the order $\pi(\bar{v})$ of $\bar{v} = (\bar{v}(1), \ldots, \bar{v}(K))$ is known to the agent, where the order π is a permutation satisfying that $\bar{v}(\pi(1)) \geq \ldots \geq \bar{v}(\pi(K))$. This assumption is practically reasonable as in many cases, though we may not have a precise estimation of each value $\bar{v}(k)$, we do know their relative comparison. (For instance, for typical search engines it may well be the case that π is identity, namely $\bar{v}_t(1) \geq \ldots \geq \bar{v}_t(K)$.) Under this assumption, it can be easily shown that the expected reward is maximized when the items are listed in the decreasing order of their attractiveness.

To give a more formal treatment of the award, consider the reward function $f : \Pi_k(E) \times [0,1]^E \times [0,1]^K \to [0,1]$ defined by

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

where $A = (a_1, \ldots, a_K)$. In this notation, the reward in time step t is $\mathbf{r}_t = f(\mathbf{A}_t, \mathbf{v}_t, \mathbf{w}_t)$. Due to the assumed independence of all $\{\mathbf{v}_t\}$ and $\{\mathbf{w}_t\}$, it is easily seen that for any fixed action A, the expected reward is $f(A, \bar{v}_t, \bar{w}_t)$.

The performance of the learning agent is evaluated by the pseudo-regret, the difference of cumulative reward of the optimal actions and that of the actions of the agent:

$$\mathcal{R}(n) = \mathbb{E}\Big[\sum_{t=1}^{n} \big(f(A_t^*, \bar{v}_t, \bar{w}_t) - f(\mathbf{A}_t, \bar{v}_t, \bar{w}_t)\big)\Big], \tag{2}$$

where

$$A_t^* = \mathrm{argmax}_{A \in \Pi_K(E)} f(A, \bar{v}_t, \bar{w}_t)$$

is the optimal list that maximizes the expected reward in step t.

We adopt the standard assumption that in contextual bandits that all contextual vectors $x_{t,a} \in \mathbb{R}^d$ are assumed to have bounded norm $\|x_{t,a}\|_2 \leq 1$. Besides, we assume that the attraction weight $\mathbf{w}_t(a)$ satisfies the *generalized linear model* (GLM), a flexible extension of the ordinary linear model that previous cascading bandit studies assumed. More precisely, assume that

$$\bar{w}_t(a) = \mathbb{E}[\mathbf{w}_t(a)|\mathcal{H}_t] = \mu(\theta_*^\top x_{t,a}), \tag{3}$$

where $\{\mathcal{H}_t\}_{t=1}^n$ represents the history containing clicks and features up to time t, and θ_* is a fixed but unknown vector $\theta_* \in \mathbb{R}^d$. The *inverse link function* μ is chosen such that $0 \leq \mu(\theta_*^\top x_{t,a}) \leq 1$ for any a and t. This GLM admits a wider range of nonlinear distributions such as Gaussian, binomial, Poisson, gamma distributions, etc. In particular, when the feedback is binary or count variables, the logistic or Poisson regression can be used. Especially in the present DCM setting, the logistic regression fits the web page recommendation better than the linear model [14].

3 Algorithm and Results

3.1 Algorithm

To maximize user satisfaction, two sets of parameters, $\bar{w}_t$ and $\bar{v}_t$ need to be estimated. We assume that the order of the expected termination weight is known to the agent, which in practice can be easily estimated using historical click data. The problem then reduces to the estimation of the mean and variance of the expected attraction weight. Due to the limited feedback, it is unclear whether the user is attracted by the item of the last click position, which is denoted by $\mathcal{C}_t \in \{0, 1, \ldots, K\}$, where $\mathcal{C}_t = 0$ means no item has been clicked. The algorithm therefore simply uses the feedback before $\mathcal{C}_t$ for updates. As introduced before, the random variable $\mathbf{w}_t(a)$ satisfies Eq. (3) with the inverse link function μ assumed to be twice continuously differentiable and strictly increasing. We further assume that μ is a k_μ-Lipschitz function (namely, the first order derivative of μ is upper bounded by k_μ), and that $c_\mu := \inf_{\{\|x\|_2 \le 1, \|\theta - \theta^*\|_2 \le 1\}} \mu'(\theta^\top x) > 0$. For logistic regression, $\mu(x) = 1/(1 + e^{-x})$ and it is easily verified that $c_\mu = 0.1$, $k_\mu = 0.25$ suffice for the requirements. Given the historical information $\{(x_{s,a}, \mathbf{w}_s(\mathbf{a}_k^s)) : s \in [t], a \in E, k \in [\mathcal{C}_s]\}$, where $(x_s, \mathbf{w}_s) \in \mathcal{H}_s$, the estimator $\hat{\theta}_t$ can be efficiently obtained by solving the following equation:

$$\sum_{s=1}^{t} \sum_{k=1}^{\mathcal{C}_s} \left(\mathbf{w}_s(\mathbf{a}_k^s) - \mu(\theta^\top x_{s,\mathbf{a}_k^s}) \right) x_{s,\mathbf{a}_k^s} = 0. \tag{4}$$

For logistic regression, this step can be computed by Newton method. Next, we design an upper confidence bound of the expected attraction weight. Define $\mathbf{V}_t = \lambda I + \sum_{s=1}^{t} \sum_{k=1}^{\mathcal{C}_s} x_{s,\mathbf{a}_k^s} x_{s,\mathbf{a}_k^s}^\top$, we have the following fact by Lemma 3 in [16].

Lemma 1. *For any $\delta \in [1/n, 1)$, with probability at least $1-\delta$, for all $1 \le t \le n$, we have*

$$\|\hat{\theta}_t - \theta_*\|_{\mathbf{V}_t} \le \frac{\sigma}{c_\mu} \sqrt{\frac{d}{2} \log(1 + t/(\lambda d)) + \log(1/\delta)}. \tag{5}$$

Here the l_2-norm of x based on a positive definite matrix A is defined by $\|x\|_A = \sqrt{x^\top A x}$. Building on this, we can bound $|\mu(\hat{\theta}_t^\top x_{t,a}) - \mu(\theta_*^\top x_{t,a})|$ by first applying the definition of k_μ-Lipschitz of function μ and then using the Cauchy-Schwartz inequality.

$$\begin{aligned} |\mu(\hat{\theta}_t^\top x_{t,a}) - \mu(\theta_*^\top x_{t,a})| &\le k_\mu |\hat{\theta}_t^\top x_{t,a} - \theta_*^\top x_{t,a}| \le k_\mu \|\hat{\theta}_t - \theta_*\|_{\mathbf{V}_t} \|x_{t,a}\|_{\mathbf{V}_t^{-1}} \\ &\le \frac{k_\mu \sigma}{c_\mu} \sqrt{\frac{d}{2} \log(1 + t/(\lambda d)) + \log(1/\delta)} \|x_{t,a}\|_{\mathbf{V}_t^{-1}} \end{aligned}$$

Let $\rho(t) = \frac{k_\mu \sigma}{c_\mu} \sqrt{\frac{d}{2} \log(1 + t/(\lambda d)) + \log(1/\delta)}$, and define the upper confidence bound of the expected attraction weight for item a at time t by

$$\mathbf{U}_t(a) = \min\{\mu(\hat{\theta}_{t-1}^\top x_{t,a}) + \rho(t-1) \|x_{t,a}\|_{\mathbf{V}_{t-1}^{-1}}, 1\}, \tag{6}$$

where the first term of $\mathbf{U}_t(a)$ is for exploitation and the second term for exploration. Choosing an item with the maximum $\mathbf{U}_t(a)$ balances the exploration and exploitation. Based on the above discussion, we propose an algorithm given in box Algorithm 1. Firstly, for each item in the ground item set, an upper confidence bound $\mathbf{U}_t \in [0,1]^E$ for the expected attraction weight is calculated. Then the agent uses any $\tilde{v}_t$ that has the same order as $\bar{v}_t$, gets a maximizer $\mathbf{A}_t = \text{argmax}_{A \in \Pi_K(E)} f(A, \tilde{v}_t, \mathbf{U}_t)$, and recommends the list. After user examines the list, the agent observes the last click position $\mathcal{C}_t$, and $\mathbf{w}_t(\mathbf{a}_k^t)$, $k \in [\mathcal{C}_t]$ (Here we adopt the notation that $[0] = \emptyset$). The estimator $\hat{\theta}_t$ of θ_* is then updated based on new feedback. Finally, the related statistics are updated for the next time step.

Algorithm 1. Contextual DCM Bandits with Generalized Linear Payoff (GL-CDCM)

Parameters : $\delta = \frac{1}{\sqrt{n}}$; $\lambda \geq K$
Initialization : $\hat{\theta}_0 = 0$, $\rho(0) = 1$, $\mathbf{V}_0 = \lambda I$
for $t = 1$ to n **do**
  Obtain context $x_{t,a}$ for all $a \in E$
  $\forall a \in E$, compute
  $\mathbf{U}_t(a) = \min\{\mu(\hat{\theta}_{t-1}^\top x_{t,a}) + \rho(t-1)\|x_{t,a}\|_{\mathbf{V}_{t-1}^{-1}}, 1\}$
  $\mathbf{A}_t \leftarrow \text{argmax}_{A \in \Pi_K(E)} f(A, \tilde{v}_t, \mathbf{U}_t)$
  Play $\mathbf{A}_t$ and observe $\mathcal{C}_t$, $\mathbf{w}_t(\mathbf{a}_k^t)$, $k \in [\mathcal{C}_t]$
  Solve $\hat{\theta}_t$ from
  $\sum_{s=1}^{t} \sum_{k=1}^{\mathcal{C}_s} (\mathbf{w}_s(\mathbf{a}_k^s) - \mu(\theta_t^\top x_{s,\mathbf{a}_k^s})) x_{s,\mathbf{a}_k^s} = 0$
  $\mathbf{V}_t \leftarrow \mathbf{V}_{t-1} + \sum_{k=1}^{\mathcal{C}_t} x_{t,\mathbf{a}_k^t} x_{t,\mathbf{a}_k^t}^\top$
end for

3.2 Results

The result on the upper bound on the regret for the proposed contextual DCM bandits is presented in this section. Denote $p_v = \max_{1 \leq t \leq n} \max_{i=1,\ldots,K} (\bar{v}_t(i) - \bar{v}_t(i+1))$ by the maximal difference of expected termination weights between two consecutive positions over all time. The main theorem on the regret is stated as follows.

Theorem 1. *For $n \geq 1$, and the reward function $f(A, v, w) = 1 - \prod_{k=1}^{K}(1 - v(k)w(a_k))$, the pseudo-regret $\mathcal{R}(n)$ of Algorithm 1 has the following bound*

$$\mathcal{R}(n) \leq \frac{4dKp_v k_\mu \sigma}{c_\mu} \sqrt{nK \log\left(\frac{1 + n/(\lambda d)}{\delta}\right) \log(1 + Kn/(\lambda d))}. \tag{7}$$

The theorem shows a $\tilde{O}(d\sqrt{n})$ pseudo-regret bound, which is independent of L, and improves the previous regret bound of [17] by a $\sqrt{\log(n)}$ term, though

our result is under the combinatorial setting. With an additional assumption on item generating process, the result may be further improved by a $\sqrt{d}$-order while sacrificing an increase on order of $\log(n)$ by using Theorem 1 of [16].

Proof. To begin with, we bound the one-step regret at time t, denoted by $\mathcal{R}_t = f(A_t^*, \mathbf{v}_t, \mathbf{w}_t) - f(\mathbf{A}_t, \mathbf{v}_t, \mathbf{w}_t)$, then

$$\begin{aligned}
\mathbb{E}[\mathcal{R}_t|\mathcal{H}_t] &= f(A_t^*, \bar{v}_t, \bar{w}_t) - f(\mathbf{A}_t, \bar{v}_t, \bar{w}_t) \\
&\leq \sum_{k=1}^{K} \bar{v}_t(k)\bar{w}_t(a_k^*) - \sum_{k=1}^{K} \bar{v}_t(k)\bar{w}_t(\mathbf{a}_k^t) \qquad (8) \\
&= \sum_{i=1}^{K} (\bar{v}_t(i) - \bar{v}_t(i+1)) \sum_{k=1}^{i} (\bar{w}_t(a_k^*) - \bar{w}_t(\mathbf{a}_k^t)) \\
&\leq p_v \sum_{i=1}^{K} \sum_{k=1}^{i} (\bar{w}_t(a_k^*) - \bar{w}_t(\mathbf{a}_k^t)), \qquad (9)
\end{aligned}$$

where $\bar{v}_t(i+1) = 0$. The inequality (8) is because of the definition of A_t^* and f, while (9) is by definition of the p_v. We can observe that the problem has reduced to the cascading problem of bounding $\sum_{k=1}^{i}(\bar{w}_t(a_k^*) - \bar{w}_t(\mathbf{a}_k^t)$, which is equal to $\sum_{k=1}^{i} \mu(\theta_*^\top x_{t,a_k^*}) - \mu(\theta_*^\top x_{t,\mathbf{a}_k^t})$. We need the following Lemma 2 to bound this cascade difference.

Lemma 2. *Let $t \geq 1$ and $\mathbf{A}_t = (\mathbf{a}_1^t, ..., \mathbf{a}_i^t)$, $i \in [K]$, we have:*

$$\sum_{k=1}^{i} (\mu(\theta_*^\top x_{t,a_k^*}) - \mu(\theta_*^\top x_{t,\mathbf{a}_k^t})) \leq 2 \sum_{k=1}^{i} \rho(t-1) \|x_{t,\mathbf{a}_k^t}\|_{\mathbf{V}_{t-1}^{-1}}.$$

Proof. Let $A_t^* = (a_1^*, \ldots, a_K^*)$. By the definition of $\mathbf{A}_t$, which is set of items with the largest UCBs placed to the most terminating position, we have $\sum_{k=1}^{i} \mathbf{U}_t(\mathbf{a}_k^*) \leq \sum_{k=1}^{i} \mathbf{U}_t(\mathbf{a}_k^t)$, $i = [K]$, that is,

$$\begin{aligned}
&\textstyle\sum_{k=1}^{i} \mu(\hat{\theta}^\top x_{t,a_k^*}) + \rho(t-1)\|x_{t,a_k^*}\|_{\mathbf{V}_{t-1}^{-1}} \\
&\leq \textstyle\sum_{k=1}^{i} \mu(\hat{\theta}^\top x_{t,\mathbf{a}_k^t}) + \rho(t-1)\|x_{t,a_k^t}\|_{\mathbf{V}_{t-1}^{-1}}. \qquad (10)
\end{aligned}$$

Then

$$\begin{aligned}
&\sum_{k=1}^{i} \mu(\theta_*^\top x_{t,a_k^*}) - \mu(\theta_*^\top x_{t,\mathbf{a}_k^t}) \\
&= \sum_{k=1}^{i} \mu(\theta_*^\top x_{t,a_k^*}) - \mu(\hat{\theta}^\top x_{t,a_k^*}) + \mu(\hat{\theta}^\top x_{t,a_k^*}) - \mu(\hat{\theta}^\top x_{t,\mathbf{a}_k^t}) + \\
&\quad \mu(\hat{\theta}^\top x_{t,\mathbf{a}_k^t}) - \mu(\theta_*^\top x_{t,\mathbf{a}_k^t}) \\
&\leq \rho(t-1) \sum_{k=1}^{i} \|x_{t,a_k^*}\|_{\mathbf{V}_{t-1}^{-1}} + \left(\|x_{t,a_k^t}\|_{\mathbf{V}_{t-1}^{-1}} - \|x_{t,a_k^*}\|_{\mathbf{V}_{t-1}^{-1}} \right) + \|x_{t,a_k^t}\|_{\mathbf{V}_{t-1}^{-1}} \qquad (11)
\end{aligned}$$

$$= 2\rho(t-1)\sum_{k=1}^{i}\|x_{t,a_k^t}\|_{\mathbf{V}_{t-1}^{-1}},$$

where Eq. (11) is obtained by applying (10). Our next step is to bound $\sum_{s=1}^{t}\sum_{k=1}^{i}\left\|x_{s,\mathbf{a}_k^s}\right\|^2_{V_t^{-1}}$ by Lemma 4.4 in [11], when $\lambda \geq K$,

Lemma 3. *If $\lambda \geq K$, then*

$$\sum_{s=1}^{t}\sum_{k=1}^{i}\left\|x_{s,\mathbf{a}_k^s}\right\|^2_{V_t^{-1}} \leq 2d\log(1+\frac{Kt}{\lambda d}).$$

Building upon the previous discussion, we have:

$$\begin{aligned}
\mathcal{R}(n) &= \sum_{t=1}^{n}\mathbb{E}[\mathbb{E}[\mathcal{R}_t|\mathcal{H}_t]] \\
&\leq p_v\sum_{t=1}^{n}\mathbb{E}\left[\sum_{i=1}^{K}\sum_{k=1}^{i}(\bar{w}_t(a_k^*)-\bar{w}_t(\mathbf{a}_k^t))\right] \qquad (12)\\
&\leq p_v\sum_{t=1}^{n}\mathbb{E}\left[\sum_{i=1}^{K}\sum_{k=1}^{i}2\rho(t-1)\left\|x_{t,\mathbf{a}_k^t}\right\|_{\mathbf{V}_t^{-1}}\right] \\
&\leq 2\rho(n)p_v\sum_{t=1}^{n}\mathbb{E}\left[\sum_{i=1}^{K}\sum_{k=1}^{i}\left\|x_{t,\mathbf{a}_k^t}\right\|_{\mathbf{V}_t^{-1}}\right]. \qquad (13)
\end{aligned}$$

where Eq. (12) is due to the tower rule and the inequality (13) holds since $\rho(t)$ increases with t. Applying the Cauchy-Schwarz inequality on the current result, we can derive that:

$$\mathcal{R}(n) \leq 2\rho(n)p_v\mathbb{E}\left[\sqrt{\left(n\sum_{i=1}^{K}\sum_{k=1}^{i}1^2\right)\left(\sum_{t=1}^{n}\sum_{i=1}^{K}\sum_{k=1}^{i}\left\|x_{t,\mathbf{a}_k^t}\right\|^2_{\mathbf{V}_t^{-1}}\right)}\right].$$

Substituting $\rho(n)$ back and applying Lemma 3 back yields our claimed result.

3.3 Computationally Efficient Updates

Though our proposed GL-CDCM enjoys good theoretical properties, the computational cost may be high in some applications. The inverse of a $d \times d$ matrix is computed at each time step while the MLE is calculated using samples up to the current time step, which is increased linearly over time. We provide an iterative optimization solution for GL-CDCM for the logistic regression where $\mu(x) = 1/(1+\exp(-x))$, denoted by GL-CDCM (SGD).

Instead of solving Eq. (4), we use the stochastic logistic gradient at time t

$$\mathbf{g}_t = \sum_{k=1}^{\mathcal{C}_t}\left(\mu(\hat{\theta}_t^\top x_{t,\mathbf{a}_k^t}) - \mathbf{w}_t(\mathbf{a}_k^t)\right)x_{t,\mathbf{a}_k^t}, \qquad (14)$$

and we can update on $\hat{\theta}_t$ by

$$\hat{\theta}_t = \hat{\theta}_{t-1} - \eta \mathbf{g}_t, \tag{15}$$

where η is the learning rate.

Let $\mathbf{C}_t \in \mathbb{R}^{\mathcal{C}_t \times d}$ be the matrix whose rows are the feature vectors of the observed items at time t. Then $\mathbf{V}_t = \mathbf{V}_{t-1} + \mathbf{C}_t^\top \mathbf{C}_t$. Let $\mathbf{G}_t = I + \mathbf{C}_t \mathbf{V}_t^{-1} \mathbf{C}_t^\top$, based on the Woodbury matrix identity [18], $\mathbf{V}_t^{-1}$ can be calculated efficiently using

$$\mathbf{V}_t^{-1} = \mathbf{V}_{t-1}^{-1} - \mathbf{V}_{t-1}^{-1} \mathbf{C}_t^\top \mathbf{G}_t^{-1} \mathbf{C}_t \mathbf{V}_{t-1}^{-1}, \tag{16}$$

in time $O(Kd^2)$.

Therefore, the burden of computing the inverse of a $d \times d$ matrix of is reduced to computing the inverse of a square matrix of dimension at most K, which is always smaller than d and can be much smaller in practice.

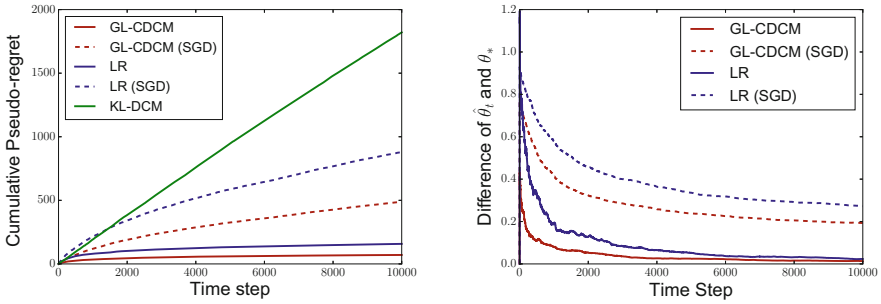
4 Experiments

4.1 Synthetic Data

In this section, we compare our algorithms (GL-CDCM) with the dcmKL-UCB algorithm proposed in [15] (denoted as *KL-DCM* in our comparisons) and the logistic regression (LR) on the synthetic data. Here LR means for each time step t, it conducts logistic regression on all historical data and uses the obtained parameters to choose the current items, which corresponds to selecting arms by values of $\mu(\hat{\theta}_{t-1}^\top x_{t,a})$, instead of $\mathbf{U}_t(a)$ (which has an additional exploration term) in Line 5-6 for our Algorithm 1.

We simulate a scenario of web search as follows. First, we randomly select the model parameter θ_*. Then at each time step t, randomly select contextual vectors $x_{t,a}$ for each item a and expected termination weights $\bar{v}_t$. Then according to Eq. (3), the expected attraction weight $\bar{w}_t$ is computed by the given θ_*. Both attraction weights $\mathbf{w}_t$ and termination weights $\mathbf{v}_t$ are then drawn from Bernoulli distribution with the respective mean. The sigmoid function $\mu(x) = 1/(1 + \exp(-x))$ serves as the inverse link function. The evaluation criterion is the cumulative pseudo-regret defined in Eq. (2).

The curves of the cumulative regrets for these algorithms, i.e. GL-CDCM, GL-CDCM (SGD), LR, LR (SGD) and KL-DCM, under $n = 10^4$ are shown in Fig. 1(a). To further demonstrate the estimation ability of GL-CDCM and LR, the cosine distances between $\hat{\theta}_t$ and θ_*, i.e., $1 - \frac{\hat{\theta}_t^\top \theta_*}{\|\hat{\theta}_t\|_2 \|\theta_*\|_2}$, are calculated and shown in Fig. 1(b), where the value 0 indicates that the learning agent correctly estimates θ^*. We do not show KL-DCM in Fig. 1(b) since it does not estimate the parameter θ_*. As depicted in Fig. 1(a), KL-DCM has the largest regret since it ignores the contextual information. For both GL-CDCM and LR, the SGD version generally has higher regret, which is a price to pay for efficiency. Compared to the LR algorithm, the bandit algorithm balances the exploitation and exploration and therefore has a better performance. Furthermore, the error curve shows that the GL-CDCM converges more quickly than LR.



(a) Cumulative regrets.

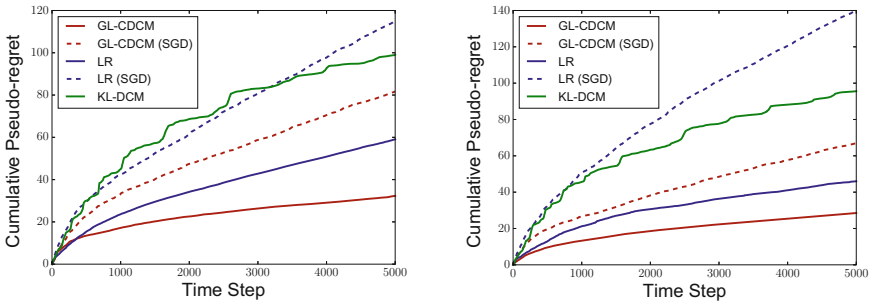
(b) Difference between θ_* and $\hat{\theta}_t$.

Fig. 1. Experimental results of different recommendation algorithms on synthetic data.

4.2 Web Page Recommendation

In this section, we test our algorithms on the Yandex Personalized Web Search dataset [19], which contains 35 million search sessions. Let M be the number of users and L be the number of web pages. We use top 3 most frequent queries for evaluation. Each query corresponds to one DCM which is estimated using PyClick library [8]. In all the algorithms, we assume that the higher positions have higher expected termination weight. In order to derive the feature vectors for web pages, we first construct a sparse matrix $A \in \mathbb{Z}^{M \times L}$ where $A(i, j) \in \mathbb{Z}$ denotes the number that user i clicked on web page j. Then the feature vector is obtained through the SVD decomposition of A, i.e. $A = USV^\top$. We use $V = [v_1; \ldots; v_L] \in \mathbb{R}^{L \times d}$ as the contextual information for the L web pages. We set $d = 200$, $K = 10$, and $L = 100$. The cumulative pseudo-regret over 5000 rounds for our proposed GL-CDCM, GL-CDCM (SGD), LR, LR (SGD) and KL-DCM are shown in Fig. 2. To incorporate the user features, we concatenate user and item features as the contextual information. Let $U = [u_1; \ldots; u_M] \in \mathbb{R}^{M \times d}$, then $x_{i,j} = [u_i, v_j] \in \mathbb{R}^{2d}$ for user i and web page j. The features derived from outer product where $x_{i,j} = u_i \otimes v_j$ are also tested, but the performance is not as good as $x_{i,j} = [u_i, v_j]$. At each time step, a user is randomly selected. Follow the previous setting of the parameters, the results are displayed in Fig. 2(b).

For the setting that only the item features are used, after 5000 rounds, the proposed GL-CDCM obtains a regret of 32.28, which is much lower than 59.08 for LR and 99.09 for KL-DCM. Furthermore, the curve for KL-CDCM forms a stair-step pattern since the ground item set is changing and the algorithm needs to learn from the cold start from time to time. In contrast, GL-CDCM and LR make use of the contextual information, and therefore achieve a better estimation. Compared with LR, which is always exploiting, GL-CDCM explores more and achieves a lower cumulative pseudo-regret. The SGD versions generally have a higher regret for both GL-CDCM and LR, 81.71 for GL-CDCM (SGD) and 114.96 for LR (SGD), but the time complexity reduces significantly. In addition, the proposed GL-CDCM (SGD) still outperforms LR (SGD) because



(a) with only item features (b) with both item and user features

Fig. 2. Experimental results of different recommendation algorithms on Yandex dataset.

of exploration. A similar pattern is also observed in the setting of involving both user and item features, where more useful information are provided and the regrets of GL-CDCM and LR decrease to 28.51 and 46.00, respectively. The experimental results are consistent with our previous discussions and show that our proposed algorithm has better performance even for practical problems, where the assumptions might be violated.

5 Conclusion

In this paper, we present a bandit algorithm (and SGD variant) for web page recommendation that automatically balances the exploration and exploitation. We formulate the problem of DCM bandits with contextual information. The dependent click model (DCM) covers the scenario of multiple clicks and is a popular click model in web search. The contextual information is incorporated in our work to better estimate the expected attraction weight. Under a reasonable assumption on knowing the order of the expected termination weight, we prove a regret bound of $\tilde{O}(d\sqrt{n})$ for the algorithm. A computationally efficient version is also given by removing the expensive step of computing the MLE on a linearly increasing sample set, and reducing the cost of inverting a $d \times d$ matrix. Experimental results confirm the value of exploring, utilizing the contextual information and adopting a generalized linear model.

Acknowledgment. This work is sponsored by Huawei Innovation Research Program.

References

1. Aggarwal, C.C.: Recommender Systems. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-29659-3
2. Richardson, M., Dominowska, E., Ragno, R.: Predicting clicks: estimating the click-through rate for new ads. In: Proceedings of the 16th International Conference on World Wide Web, pp. 521–530. ACM (2007)

3. Rendle, S.: Factorization machines. In: 2010 IEEE 10th International Conference on Data Mining (ICDM), pp. 995–1000. IEEE (2010)
4. Wang, X., Wang, Y., Hsu, D., Wang, Y.: Exploration in interactive personalized music recommendation: a reinforcement learning approach. ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) **11**(1), 7 (2014)
5. Gittins, J., Glazebrook, K., Weber, R.: Multi-armed Bandit Allocation Indices. Wiley, Hoboken (2011)
6. Li, L., Chu, W., Langford, J., Schapire, R.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)
7. Villar, S.S., Bowden, J., Wason, J.: Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. Stat. Sci.: Rev. J. Inst. Math. Stat. **30**(2), 199 (2015)
8. Chuklin, A., Markov, I., de Rijke, M.: Click models for web search. Synth. Lect. Inf. Concepts, Retrieval, Serv. **7**(3), 1–115 (2015)
9. Kveton, B., Wen, Z., Ashkan, A., Szepesvari, C.: Cascading bandits: learning to rank in the cascade model. In: Proceedings of the 32th International Conference on Machine Learning (2015)
10. Kveton, B., Wen, Z., Ashkan, A., Szepesvari, C.: Combinatorial cascading bandits. In: Advances in Neural Information Processing Systems, pp. 1450–1458 (2015)
11. Li, S., Wang, B., Zhang, S., Chen, W.: Contextual combinatorial cascading bandits. In: Proceedings of The 33rd International Conference on Machine Learning, pp. 1245–1253 (2016)
12. Guo, F., Liu, C., Wang, Y.M.: Efficient multiple-click models in web search. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining, pp. 124–131. ACM (2009)
13. Abbasi-Yadkori, Y., Pál, D., Szepesvári, C.: Improved algorithms for linear stochastic bandits. In: Advances in Neural Information Processing Systems, pp. 2312–2320 (2011)
14. Li, L., Chu, W., Langford, J., Moon, T., Wang, X.: An unbiased offline evaluation of contextual bandit algorithms with generalized linear models. In: Proceedings of the Workshop on On-line Trading of Exploration and Exploitation vol. 2, pp. 19–36 (2012)
15. Katariya, S., Kveton, B., Szepesvári, C., Wen, Z.: DCM bandits: learning to rank with multiple clicks. In: Proceedings of The 33rd International Conference on Machine Learning (2016)
16. Li, L., Lu, Y., Zhou, D.: Provable optimal algorithms for generalized linear contextual bandits. In: Proceedings of The 34rd International Conference on Machine Learning (2017)
17. Filippi, S., Cappe, O., Garivier, A., Szepesvári, C.: Parametric bandits: the generalized linear case. In: Advances in Neural Information Processing Systems, pp. 586–594 (2010)
18. Hager, W.W.: Updating the inverse of a matrix. SIAM Rev. **31**(2), 221–239 (1989)
19. Yandex: Yandex personalized web search challenge (2013)