Learning to Plan Variable Length Sequences of Actions with a Cascading Bandit Click Model of User Feedback

Anirban Santara
Google Research

Gaurav Aggarwal
Google Research

Shuai Li
Shanghai Jiao Tong University

Claudio Gentile
Google Research

Abstract

Motivated by problems of ranking with partial information, we introduce a variant of the cascading bandit model that considers flexible length sequences with varying rewards and losses. We formulate two generative models for this problem within the generalized linear setting, and design and analyze upper confidence algorithms for it. Our analysis delivers tight regret bounds which, when specialized to standard cascading bandits, results in sharper guarantees than previously available in the literature. We evaluate our algorithms against a representative sample of cascading bandit baselines on a number of real-world datasets and show significantly improved empirical performance.

1 INTRODUCTION

A well-known problem in content recommendation is the generation of *slates* of items whereby, given a set of available items and a limited number of available slots, the goal of the system is to come up with an ordered sequence of items to be arranged in the slots so as to best fulfil some goal, like improving the experience of the user at hand. Applications are ubiquitous, from web search to news recommendation and from computational advertising to web page content optimization. These are among the most prominent motivating applications behind the more abstract problem often called *learning to rank.*

The cascade model (e.g., Chuklin et al. (2015)) for ranking problems has emerged as a simple and effective way to model user behavior in a number of applications. In this model, the user scans the slate sequentially from top to bottom and clicks on the first item they find attractive, disregarding all subsequent items in the slate. The length of the slate may vary widely across applications, ranging from a few items in computational advertising to dozens in news recommendation to hundreds in web search. In these and many other dynamic domains, one has to deal with a near continuous stream of new items to be recommended, along with new users to be served. Out of the collected user feedback, and in the face of a constantly evolving content universe and set of targeted users, the learning system is expected to maintain over time a good mapping between user/item features and item rankings.

In order to encompass a variety of learning-to-rank applications for dynamic environments, we introduce a generalized version of the well-known cascading bandit model of Kveton et al. (2015a). Our model considers flexible sequence length with varying rewards and losses. The problem is broadly described by position-dependent rewards r_j and losses ℓ_j. These parameters measure how well the ranking system is doing depending on the position j of the first positive signal, as well as the potential loss associated with a sequence of j negative signals. Since rewards are positive and losses are negative, and the two sequences are decreasing with j (in particular, ℓ_j becomes more and more *negative* as j increases), this model is intended to capture a natural planning trade-off: If we commit to a long sequence, we may increase our chance of success (positive reward), but also expose ourselves to the risk of a very negative loss if all signals on that sequence turn out to be negative.

This trade-off is typical in planning scenarios where each negative signal in the sequence is indeed a *cost* for the system. As a relevant and motivating example, suppose we want to deploy our planning algorithm within a payment system (e.g., Stripe) where, at each round we process one transaction, and the goal is to find payment "routes" to fulfill the transaction. Here each payment attempt with a chosen route is an item in the list, and it comes with a cost for the system.

Proceedings of the 25[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

A positive signal on a route corresponds to payment fulfillment through that route, while a negative signal corresponds to a payment failure. Every unsuccessful attempt reduces the net reward gathered by a subsequent success, and may translate into bigger losses if in the end the payment is not fulfilled. This provides a new use case of cascade models since we have to predict a ranked sequence of routes for the payment to be fulfilled with as few retries as possible. Note that the length of the ranked sequence can be large and flexible which further aligns this application to our setting.

Our contribution. In this paper, we describe two contextual upper confidence bandit algorithms for this problem, specifically focusing on the case of long ranked sequences. We analyze the two algorithms both theoretically and experimentally. Our theoretical analysis delivers tighter regret guarantees than previous investigations. In particular, we obtain a regret bound of the form $\sqrt{bT}$, where T is the time horizon and b is the length of the ranked sequences, as opposed to $b\sqrt{T}$ achieved by prior work in cascading bandits. We then validate our algorithms experimentally on well-known benchmark datasets, and show significantly improved performance as compared to state-of-the-art algorithms proposed in the cascading bandit literature.

Related work. The study of cascading bandit models for ranking problems has been initiated by Kveton et al. (2015a). The authors study the problem of learning to rank items on a fixed number of slots under the so-called cascade click model of user behavior. Li et al. (2016); Zong et al. (2016); Li and Zhang (2018) investigate large-scale variants where the reward of an item follows (generalized) linear structure. Cheung et al. (2019) gives an analysis for Thompson sampling. Cascading bandits have also been studied under more general click models, which can recover the standard cascade click model as well as other classical click models in the literature of online learning to rank (e.g., Zoghi et al. (2017); Lattimore et al. (2018); Li et al. (2019)). Li and De Rijke (2019) considers cascading bandits in non-stationary environments, and Hiranandani et al. (2020) studies more comprehensive cascading models of user behavior that account for both position bias and diversity of recommendations. Kveton et al. (2015a) also consider algorithms where exploration occurs at the top of the list by reversing the order in which items are presented.[1] All these works consider the case of sequences with fixed length and, when specialized to the original cascading bandit model of Kveton et al. (2015a) or generalized linear variants thereof, their analyses deliver regret guarantees with a suboptimal dependence on the length of the sequence, which is a main theoretical concern in this paper. An in-context regret bound comparison to many of these works is carried out in Section 3. Further related work is discussed in Appendix C.

2 SETTING, MAIN NOTATION

We formalize our problem of contextual bandits with long and variable length sequences as follows. Learning proceeds in a discrete sequence of *time steps* (or *rounds* or *trials*). At each time t, the learner processes a *transaction* having at its disposal a (finite) set of *actions* (or *items*) $A_t = \{x_{1,t}, x_{2,t}, \ldots, x_{k_t,t}\} \subseteq A = \{x \in \mathbb{R}^d : ||x||_2 \leq 1\}$, each action being described by a d-dimensional feature vector of (Euclidean) norm at most one.[2] Set A_t is our *context* information at time t, while set A is the universe of all possible actions. Collectively, A_t may include information about the specific context in which learning is applied. In a payment scenario, this will typically include the transaction amount, the buyer and seller identities (or features thereof), the credit card company identity (or features thereof), etc. In a news recommendation problem this may include user features, news-of-the-day topic features, and so on. Each action corresponds to an item available at time t. The learning problem is parameterized by a decreasing (or non-increasing) sequence of rewards $r_{1,t}, r_{2,t}, \ldots$ and a decreasing (or non-increasing) sequence of losses $\ell_{0,t}, \ell_{1,t}, \ell_{2,t}, \ldots$, where

$$1 \geq r_{1,t} \geq r_{2,t} \geq \ldots > 0$$

$$0 > \ell_{0,t} \geq \ell_{1,t} \geq \ell_{2,t} \geq \ldots > -1 \ .$$

The rewards are positive, while the losses are negative. The dependence on t of these quantities emphasizes the potential dependence of these values on the current context. E.g., in the payment scenario, $r_{i,t}$ is often proportional to the amount of the current transaction. Moreover, to set the scale of these parameters, we shall assume throughout that $r_{i,t} \in [0, 1]$ and $\ell_{i,t} \in [-1, 0]$ for all i and t. Finally, each transaction may be accompanied by a *budget* value b_t that bounds from above the number of allowed retries, as defined next.

In round t, the algorithm is compelled to play an ordered sequence of actions $J_t = \langle x_{j_{1,t}}, x_{j_{2,t}}, \ldots x_{j_{s_t,t}} \rangle$, where each component vector $x_{j_{i,t}}$ is taken from A_t. We call J_t a *retry sequence* or simply a *sequence*[3]. The set of all such sequences J_t corresponds to the *action space* available to the learner at time t. Notice that

[1] This idea has been further explored in Combes et al. (2015), where an optimal analysis is given that, however, only applies to a non-contextual scenario with a fixed number of arms.

[2] This normalization is done for notational convenience only; any bounded action space would work here.

[3] A sequence might have repeated actions, but for simplicity we assume here each component of J_t is distinct.

the length s_t of J_t is part of the action selected by the learner (that is, the algorithm has to decide the length of the sequence as well). This length s_t determines the number of retries on the transaction at time t. J_t can also be empty; in such a case we have $s_t = 0$ and write $J_t = \langle\rangle$. The budget constraint b_t requires s_t to satisfy $s_t \leq b_t$. In general, b_t may depend on time, and there are practical scenarios where this is indeed advisable, e.g., a payment system where the number of attempts depends on the transaction amount.

Sequence J_t has associated rewards and losses as detailed next. Upon committing to J_t, if $J_t = \langle\rangle$ we simply suffer loss (or negative reward) $\ell_{0,t}$ and go to the next round. Otherwise, the first item $x_{j_{1,t}}$ is attempted. If $x_{j_{1,t}}$ is successful we gather reward $r_{1,t}$ and stop, going to the next round. If $x_{j_{1,t}}$ is unsuccessful, $x_{j_{2,t}}$ is attempted. If $x_{j_{2,t}}$ is successful we gather reward $r_{2,t}$ and again stop. In this way, finally, $x_{j_{s_t,t}}$ is attempted. If $x_{j_{s_t,t}}$ is successful we gather reward $r_{s_t,t}$ and stop. Otherwise, we "give up" and incur loss $\ell_{s_t,t}$. A pictorial illustration is given in Figure 1.

The more traditional scenario considered in past investigations (e.g., Kveton et al. (2015a); Combes et al. (2015); Zong et al. (2016); Li and Zhang (2018)), called "vanilla" in our experiments, is recovered by simply setting $r_{i,t} = 1$ and $\ell_{i,t} = 0$ for all i and t.

The general effort behind this parametrization for rewards and losses is to capture the tension between a potentially small reward of a successful late retry and a potentially small loss of an early give up. On one hand, the earlier is the success in a sequence J_t the higher the reward we gain. On the other, the later we give up (after many unsuccessful attempts) the higher is the loss we incur. Notice that this tension does not arise in the above-mentioned "vanilla" scenario.

For simplicity, in our model rewards and losses incurred at time t only depend on the position of the items in sequence J_t, rather than the actually played item in that position. Also, upon processing the transaction at time t, the algorithm has to commit to the entire sequence J_t, that is, this sequence cannot be changed on the fly based on partial observations we are gathering on that sequence.[4] So, this is indeed a (parametric) cascading bandit model.

After playing J_t at time t, the algorithm observes the reward associated with J_t, which is generated as follows. Let the *outcome* vector Y_t be a Boolean vector

[4]This is typically the case when the system is serving ranked content to (human) users, but it also applies to the payment problem in the introduction when the only signal we observe on a failed payment is the failure itself, with no extra information. In this case, it is easy to see that an optimal solution will be non-adaptive within the sequence.

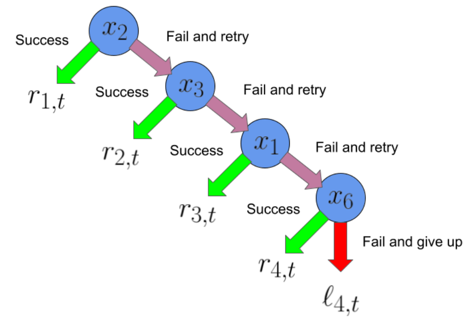

Figure 1: An illustration of the action space. Suppose at time t the algorithm can select among items in $A_t = \{x_1, \ \ldots, \ x_{10}\}$, and that $b_t = 5$. In this example $J_t = \langle x_2, x_3, x_1, x_6\rangle$. If action x_2 succeeds, we gather $r_{1,t}$, while if x_2 fails, x_3, x_1, and x_6 are tried in turn. If all these actions fail, we incur loss $\ell_{4,t}$. Also, notice that if x_3 was successful, we do not actually see whether x_1 and x_6 would have been successful or not.

$Y_t = (y_{1,t}, \ldots, y_{|A_t|,t}) \in \{0,1\}^{|A_t|}$. Then we can define the reward $R_t(J_t, Y_t)$ of sequence J_t at time t (i.e., on the transaction occurring at time t) w.r.t. outcome Y_t as follows (for ease of notation, we drop subscript t and leave the dependence on A_t implicit):

$$R(J, Y) = r_1 y_{j_1} + \ldots + r_s y_{j_s} \prod_{i=1}^{s-1}(1 - y_{j_i}) + \ell_s \prod_{i=1}^{s}(1 - y_{j_i}) \tag{1}$$

if $J \neq \langle\rangle$, and $R(J, Y) = \ell_0$ otherwise, where s is the length of J. The above simply encodes the decision list exemplified by Figure 1, with the addition that if $J_t = \langle\rangle$ the algorithm decides to give up immediately, hence incurring loss $\ell_{0,t}$, irrespective of the outcome vector Y_t. As in standard cascading bandits, the algorithm does not observe the entire outcome vector Y_t, in fact, it specifically observes those components of Y_t allowing to determine the actual value of reward $R_t(J_t, Y_t)$. We learn a generative model of Y_t as described next.

2.1 Generative model

We loosely follow Zong et al. (2016); Li and Zhang (2018); Hiranandani et al. (2020). Given the special form of the reward function, all we need to model are specific conditional probabilities. In order to properly define a generative model for Y_t, we start off by formally viewing Y_t as a Boolean random vector $Y_t = (y_{1,t}, \ldots, y_{|A_t|,t}) \in \{0,1\}^{|A_t|}$ with joint distribution $p_{Y_t}(A_t)$. Notice that Y_t's components need not be independent. The marginals and relevant conditional distributions of $p_{Y_t}(A_t)$ are defined as follows. For brevity, let $p(x_j)$ denote the (marginal) probability that item x_j succeeds, and

$$p(x_j \mid x_{i_1}, \ldots, x_{i_k}) \tag{2}$$

be the probability that x_j succeeds given that $x_{i_1}, \ldots, x_{i_k}$ have all failed. Once all conditional probabilities (2) for all $x_j, x_{i_1}, \ldots, x_{i_k}$ are available, we are automatically defining the generative process for the outcome Y_t which is relevant to a sequence $J_t = \langle x_{j_{1,t}}, x_{j_{2,t}}, \ldots, x_{j_{s_t,t}} \rangle$. This is because, for the sake of computing $R_t(J_t, Y_t)$, the relevant events associated with Y_t are those encoded by the strings

$$\langle 1 \rangle, \langle 0, 1 \rangle, \ldots, \langle \underbrace{0, \ldots, 0}_{s_t - 1 \ zeroes}, 1 \rangle, \langle \underbrace{0, \ldots, 0}_{s_t \ zeroes} \rangle \ , \qquad (3)$$

where the order of components within each string is determined by J_t, and,

$$\mathbb{P}\left(\langle \underbrace{0, \ldots, 0}_{k \ zeroes}, 1 \rangle \right) = \prod_{i=1}^{k-1} \Big(1 - p(x_{j_{i,t}} \,|\, x_{j_{1,t}}, \ldots, x_{j_{i-1,t}}) \Big) \times p(x_{j_{k,t}} \,|\, x_{j_{1,t}}, \ldots, x_{j_{k-1,t}}) \ ,$$

for $k = 0, \ldots, s_t - 1$,

$$\mathbb{P}\left(\langle \underbrace{0, \ldots, 0}_{s_t \ zeroes} \rangle \right) = \prod_{i=1}^{k} \Big(1 - p(x_{j_{i,t}} \,|\, x_{j_{1,t}}, \ldots, x_{j_{i-1,t}}) \Big) \ .$$

We will soon give (2) a parametric form. For now, observe that, based on the above generative model, we can define the expected reward $\mathbb{E}_{Y_t}[R_t(J_t, Y_t)]$ of J_t on A_t w.r.t. the random draw of Y_t. Specifically, if we take an expectation of (1) we obtain (we again drop subscript t for readability):

$$\begin{aligned} &\mathbb{E}_Y[R(J,Y)] \\ &= r_1\, p(x_{j_1}) + \ldots \\ &\ + r_s\, p(x_{j_s} \,|\, x_{j_1}, \ldots, x_{j_{s-1}}) \prod_{i=1}^{s-1} \Big(1 - p(x_{j_i} \,|\, x_{j_1}, \ldots, x_{j_{i-1}}) \Big) \\ &\ + \ell_s \prod_{i=1}^{s} \Big(1 - p(x_{j_i} \,|\, x_{j_1}, \ldots, x_{j_{i-1}}) \Big) \ , \qquad (4) \end{aligned}$$

and $\mathbb{E}_Y[R(J,Y)] = \ell_0$ if $J = \langle \rangle$. The involved conditional probabilities (2) are the only ones that matter in computing the expected reward $\mathbb{E}_Y[R(J,Y)]$. This quantity can be either positive or negative, due to the fact that the last term in (4) is negative.

For a given pair (A_t, b_t), a natural benchmark to compare to is the *Bayes optimal* sequence $J_t^* = \langle x_{j^*_{1,t}}, x_{j^*_{2,t}}, \ldots, x_{j^*_{s^*_t,t}} \rangle$, that is, the sequence J_t that maximizes $\mathbb{E}_{Y_t}[R_t(J_t, Y_t)]$ over all possible sequences built on A_t, of length at most b_t. Recall that J_t^* is computed by knowing beforehand all probabilities (2) for all candidate sequences J_t. Consequently, we define the time-t (pseudo) *regret* of an algorithm that commits to J_t on A_t as $\mathbb{E}_{Y_t}[R_t(J_t^*, Y_t)] - \mathbb{E}_{Y_t}[R_t(J_t, Y_t)]$, and its cumulative regret over T rounds on the sequence of pairs $(A_1, b_1), (A_2, b_2) \ldots, (A_T, b_T)$ as

$$\sum_{t=1}^{T} \mathbb{E}_{Y_t}[R_t(J_t^*, Y_t)] - \mathbb{E}_{Y_t}[R_t(J_t, Y_t)] \ .$$

Our goal is to make the above quantity as small as possible (with high probability). Next, we formulate a parametric model for the conditional probabilities (2), and show: (i) how to compute J_t^*, and (ii) how to define the contextual bandit algorithms that determines J_t so as to make the cumulative regret small.

2.2 Parametric model

Given our universe of actions $A = \{x \in \mathbb{R}^d : ||x||_2 \le 1\}$, we associate each item x with a so-called *coverage* vector $c(x) = (c_1(x), \ldots, c_{d'}(x)) \in [0,1]^{d'}$, where d' is the dimensionality of a latent space of *topics*.[5] The coverage $c_i(A')$ of a (finite) set $A' \subseteq A$ of items on topic i is a monotone and sub-modular function on sets, e.g., $c_i(A') = 1 - \prod_{x \in A'}(1 - c_i(x))$, with $c_i(\emptyset) = 0$. Here we slightly abuse the notation and set $c_i(x) = c_i(\{x\})$. Following, e.g., Yue and Guestrin (2011); Hiranandani et al. (2020), we then define the d'-dimensional vector $c'(x_j \mid x_{i_1}, \ldots, x_{i_k})$ of *coverage differences*, whose i-th component is

$$c_i(\{x_{i_1}, \ldots, x_{i_k}, x_j\}) - c_i(\{x_{i_1}, \ldots, x_{i_k}\}) \in [0,1] \ .$$

Since such vectors have only positive components, we shift them to their center so as both positive and negative components exist,[6] and then divide by a constant that makes their norm at most 1. For instance, we may set

$$\bar{c}_i(x_j \mid x_{i_1}, \ldots, x_{i_k}) = \frac{1}{\sqrt{d'}} \big(2c'_i(x_j \mid x_{i_1}, \ldots, x_{i_k}) - 1 \big)$$

to be the i-th component of the transformed vector $\bar{c}(\{x_{i_1}, \ldots, x_{i_k}\})$ of coverage differences.

Our parametric model is represented by a d'-dimensional vector $u \in \mathbb{R}^{d'}$ with the link function[7]

[5]Such coverage vectors can be obtained based on domain knowledge. E.g., they may be obtained as a latent probability distribution after training a Gaussian Mixture Model where the d' Gaussian centroids represent the latent topics, and $c_i(x)$ is the probability that x belongs to topic i according to the mixture model. This is essentially what we do in our experiments in Section 5.

[6]This re-centering is simply aimed at improving the numerical properties of the resulting estimators. This is not a strictly necessary step and, as such, it does not change the semantics of the setting.

[7]As the reader can easily see, the content of this paper can be seamlessly extended to more general link functions (see, e.g., the treatment in Gentile and Orabona (2012)) but, for simplicity of presentation, we restrict to the sigmoidal link.

$\sigma : \mathbb{R} \to [0,1]$, $\sigma(z) = \frac{\exp(z)}{1+\exp(z)}$. Specifically we set the conditional probability as

$$p(x_j \mid x_{i_1}, \ldots, x_{i_k}) = \sigma(\bar{c}(x_j \mid x_{i_1}, \ldots, x_{i_k})^\top u) \ . \quad (5)$$

Hence the marginal probabilities $p(x)$ and conditional probabilities $p(x_j \mid x_{i_1}, \ldots, x_{i_k})$ are encoded as generalized linear functions with unknown parameter vector u, where the feature representation of x_j depends on $x_{i_1}, \ldots, x_{i_k}$. The idea is that if the additional topic-wise diversity brought up by x_j as compared to the already selected $x_{i_1}, \ldots, x_{i_k}$ is relevant w.r.t. the weight vector u, then the probability that x_j is successful given that $x_{i_1}, \ldots, x_{i_k}$ has failed should be large. The opposite happens if the additional diversity contributed by x_j is indifferent w.r.t. u.

We now separate two cases: (i) the independent outcome case, where only marginal probabilities $p(x)$ are needed, and (ii) the more general dependent outcome case, where also the conditional probabilities $p(x_j \,|\, x_{i_1}, \ldots, x_{i_k})$ have to be considered. As we will see in the sequel, (ii) reduces to (i), up to the computation of J_t^*. For the independent case we can simply set $\bar{c}(x \,|\, x_1, \ldots, x_k) = x$, for all $x, x_1, \ldots, x_k$, and $d' = d$ to save notations, which makes $p(x) = \sigma(u^\top x)$.

3 INDEPENDENT OUTCOMES

This is the simplest possible setting where the Boolean vector Y_t has independent components. In this case, in (2) we have $p(x_j \,|\, x_{i_1}, \ldots, x_{i_k}) = p(x_j)$ for all $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$, and x_j. Hence there is no reason to model conditional probabilities, and we restrict to modeling $p(x) = \sigma(u^\top x)$. Moreover, in this case, Bayes is formulated only by means of marginal probabilities $p(x_i)$, and can be shown[8] to reduce to sorting items in A_t in decreasing order of $p(x_j)$ and cutting the sequence so obtained at the appropriate place by a brute-force search over all lengths $s \le b_t$.

The bandit algorithm corresponding to (or mimicking) the above Bayes computation is described in Algorithm 1. In this pseudo-code and elsewhere, we use the notation $Y_t \downarrow J_t$, henceforth called outcome *projected* onto the retry sequence, to denote the binary string of the form (3) which encodes the components of outcome vector Y_t that are revealed by playing sequence J_t. Recall Figure 1 for an example: If $Y_t = (0,0,1,1,0,1,1,0,0,1)$ and $J_t = \langle x_1, x_2, x_7, x_{10} \rangle$ we have $Y_t \downarrow J_t = \langle 0,0,1 \rangle$, that is, playing J_t when the outcome is Y_t reveals the components of Y_t in the order determined by J_t up to the first 1 in Y_t. In this example, we observe the 1st, the 2nd, and the 7th component of Y_t. Notice, in particular, that we do not observe Y_t's 10th component.

[8] Due to space limitations, all proofs are postponed to the appendix.

Algorithm 1 replaces the true marginal probabilities $p(x_j) = \sigma(u^\top x_j)$ with upper confidence estimations $\widehat{p}_{j,t} = \sigma(\hat{\Delta}_{j,t} + \epsilon_{j,t})$, and then mimics the Bayes optimal computation to determine J_t. The update rule is a second-order descent method on an appropriate loss function (logistic, in this case) associated with the link function σ. Notice that the items x_j which do not occur in $Y_t \downarrow J_t$ have $s_{j,t} = 0$, hence they do not contribute to the update of M_t or w_t. Yet, it is important to emphasize that $s_{i,t}$ can be zero (that is, the corresponding component $y_{t,i}$ is not observed) also due to the fact that an earlier item than x_i in J_t has been successful. In Algorithm 1 the update $w_{c_t+j-1} \to w_{c_t+j}$ is done by computing a standard Newton step.

A convenient way of viewing the way the algorithm works is as follows. The time horizon is split into rounds $t = 1, 2, \ldots, T$, each round containing multiple update steps. At the beginning of round t, the algorithm commits to a sequence J_t of length $\widehat{s}_t$ using the weight vector w_{c_t} available at *the beginning* of that round. Then feedback sequence $Y_t \downarrow J_t$ of length $\widehat{s}'_t \le \widehat{s}_t$ is observed and a sequence $j = 1, \ldots, \widehat{s}'_t$ of updates are executed within round t. The remaining $\widehat{s}_t - \widehat{s}'_t$ are those corresponding to $s_{j,t} = 0$.

Notice that, unlike the cascading contextual bandit algorithms available in the literature (e.g., Zong et al. (2016); Li et al. (2016); Li and Zhang (2018); Liu et al. (2018a); Li (2019); Li et al. (2019); Hiranandani et al. (2020)), our Algorithm 1 clearly tells apart through the update rule the actions in the sequence J_t that have been observed to be failures ($s_{j,t} = -1$) and those that have not been observed at all ($s_{j,t} = 0$). It is this richer update rule that allow us to prove a sharper regret guarantee than those available in the literature. Also, as shown in our experiments (Section 5) this update rule turns out to be significantly more effective in practice.

It is also worth observing that the vanilla scenario where $r_{j,t} = 1$ and $\ell_{j,t} = 0$ for all j and t or even, more generally, in the case where only the losses $\ell_{j,t}$ are zero, the Bayes optimal sequence J_t^* has length b_t, and so is the length $\widehat{s}_t$ of the sequence J_t computed by Algorithm 1. This is very easy to see from the definition of function $E(\Delta_1, \ldots, \Delta_s)$ in (6) in Algorithm 1's pseudocode: All terms in the sum there are *strictly* positive, since so are all multiplicative factors involving $\sigma(\Delta_i)$ and in the vanilla scenario $r_{i,t} = 1$ for all i and t. In this case, the sequence $\widehat{J}_{t,s}$ maximizing the function $\widehat{\mathbb{E}}_{Y_t}[R(\widehat{J}_{t,s}, Y_t)]$ therein is forced to be of maximal length b_t. For the very same reason, in the vanilla scenario also the Bayes optimal sequence J_t^* will be of

---

Algorithm 1 Simplified contextual bandit algorithm in the independent case with link function $\sigma(x) = \frac{\exp(x)}{1+\exp(x)}$.

Input: Maximal budget $b > 0$, learning rate $\eta > 0$, exploration parameter $\alpha \geq 0$
Init: $M_0 = bI \in \mathbb{R}^{d\times d}$, $w_1 = 0 \in \mathbb{R}^d$, $c_1 = 1$
For $t = 1, 2, \ldots, T$

1. Get:
   - Set of actions $A_t = \{x_{1,t}, \ldots, x_{|A_t|,t}\} \subseteq \{x \in \mathbb{R}^d \,:\, ||x|| \leq 1\}$,
   - budget $b_t \leq b$;
2. For $x_j \in A_t$, set $\hat{\Delta}_{j,t} = x_j^\top w_{c_t}$;
3. Compute J_t :
   - Let $\widehat{J}_{t,s} = \langle x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,s}} \rangle$ be made of the s largest items in A_t in non-increasing order of $\widehat{p}_{j,t}$, where:
     - $\widehat{p}_{j,t} = \sigma(\hat{\Delta}_{j,t} + \epsilon_{j,t})$,
     - $\epsilon_{j,t}^2 = \alpha\, x_j^\top M_{c_t-1}^{-1} x_j$,
   - Set $\widehat{s}_t = \arg\max_{s=0,1,\ldots,b_t} \widehat{\mathbb{E}}_{Y_t}[R(\widehat{J}_{t,s}, Y_t)]$, with

$$\widehat{\mathbb{E}}_{Y_t}[R(\widehat{J}_{t,s}, Y_t)] = \begin{cases} E\left(\hat{\Delta}_{\widehat{j}_{t,1},t} + \epsilon_{\widehat{j}_{t,1},t}, \ldots, \hat{\Delta}_{\widehat{j}_{t,s},t} + \epsilon_{\widehat{j}_{t,s},t}\right) & \text{if } s \geq 1 \\ \ell_{0,t} & \text{otherwise ,} \end{cases}$$

     where

$$\begin{aligned} &E(\Delta_1, \Delta_2, \ldots, \Delta_s) \\ &\quad = r_{1,t}\,\sigma(\Delta_1) + r_{2,t}\,\sigma(\Delta_2)(1-\sigma(\Delta_1)) + \ldots + r_{s,t}\,\sigma(\Delta_s)\prod_{i=1}^{s-1}(1-\sigma(\Delta_i)) + \ell_{s,t}\prod_{i=1}^{s}(1-\sigma(\Delta_i)) \ ; \end{aligned} \tag{6}$$

   - Finally, $J_t = \widehat{J}_{t,\widehat{s}_t}$;
4. Observe feedback $Y_t \downarrow J_t = \begin{cases} \langle y_{t,\widehat{j}_{t,1}}, y_{t,\widehat{j}_{t,2}}, \ldots, y_{t,\widehat{j}_{t,\widehat{s}'_t}} \rangle = \langle 0, \ldots, 0, 1\rangle, & \text{for some } \widehat{s}'_t \leq \widehat{s}_t \qquad \text{or} \\ \langle y_{t,\widehat{j}_{t,1}}, y_{t,\widehat{j}_{t,2}}, \ldots, y_{t,\widehat{j}_{t,\widehat{s}_t}} \rangle = \langle 0, \ldots, 0, 0\rangle \end{cases}$
5. **For** $j = 1, \ldots, \widehat{s}_t$ (in the order of occurrence of items in J_t) update:

$$M_{c_t+j-1} = M_{c_t+j-2} + |s_{j,t}| x_j x_j^\top, \qquad w_{c_t+j} = w_{c_t+j-1} + \eta\,\sigma(-s_{j,t}\, w_{c_t+j-1}^\top x_j)\, s_{j,t}\, M_{c_t+j-1}^{-1} x_j \quad ,$$

   where

$$s_{j,t} = \begin{cases} 1 & \text{If } y_{t,j} \text{ is observed and } y_{t,j} = 1 \\ -1 & \text{If } y_{t,j} \text{ is observed and } y_{t,j} = 0 \\ 0 & \text{If } y_{t,j} \text{ is not observed ,} \end{cases}$$

6. $c_{t+1} \leftarrow c_t + \widehat{s}_t$.

---

maximal length b_t.

The next theorem, which is the main result of this section, applies to a version of Algorithm 1 where the learning rate η and the exploration parameter α are given specific values depending on the problem parameters b, d, T, δ – see Algorithm 2 in Appendix A for details.

Theorem 1. *Assume there exists $D > 0$ such that $u^\top x_j \in [-D, D]$ for all $x_j \in A$,*[9] *and let $b = \max_t b_t$. Then a version of Algorithm 1 exists such that with probability at least $1-\delta$, with $\delta < 1/e$, the cumulative regret of this algorithm run with link function σ satisfies*

$$\begin{aligned} &\sum_{t=1}^{T} \mathbb{E}_{Y_t}[R(J_t^*, Y_t)] - \mathbb{E}_{Y_t}[R(J_t, Y_t)] \\ &\qquad\qquad \leq 4\sqrt{T\alpha(b,d,T,\delta,D)}\, d\log(1+T) \ , \end{aligned}$$

where $\alpha(b,d,T,\delta,D) = O\left[e^{2D}\left(b + d\,\log\left(1+\frac{bdT}{\delta}\right)\right)\right]$, the big-oh hiding additive and multiplicative constants independent of T, d, b, D, and δ.

Proof sketch. The proof first shows (Lemma 2 in Ap-

[9] Notice that since we have assumed $||x_j||_2 \leq 1$ for all vectors x_j, we also have $||u||_2 \leq D$.

pendix A) a fundamental monotonicity property of function $E(\Delta_1, \ldots, \Delta_s)$ in (6), by virtue of which an upper confidence exploration scheme can be defined. Then, we relate in Lemma 3 the one-time regret of the algorithm to how close $\hat{\Delta}_{j,t}$ turns out to be to the corresponding $\Delta_{j,t} = u^\top x_{j,t}$:

$$\mathbb{E}_{Y_t}[R(J_t^*, Y_t)] - \mathbb{E}_{Y_t}[R(J_t, Y_t)] \leq \sum_{i=1}^{s_t} \epsilon_{j_i,t} \prod_{h=1}^{i-1} (1 - \sigma(\Delta_{j_h,t})) ,$$

where $J_t = \langle x_{j_1,t}, \ldots, x_{j_{s_t},t} \rangle$, and $\epsilon_{j_i,t}$ is such that $|\hat{\Delta}_{j,t} - \Delta_{j,t}| \leq \epsilon_{j,t}$. Notice that the contribution to the bound of each item in the list shrinks as we move down the list. This feature, which turns out to be key to the sharpness of the analysis. This gives rise to the $b \to \sqrt{b}$ gain which leverages the way the algorithm updates its own weights. Then, we rely on a somewhat standard analysis of online Newton step algorithms (the one we use is an adaptation of Hazan et al. (2007); Gentile and Orabona (2012)) to quantify the above approximation levels $\epsilon_{j,t}$. This bound can be found within the proof of Theorem 1 in Appendix A, where it is shown that $\epsilon_{j,t}^2 \leq \left(x_{j,t}^\top \bar{M}_{c_t-1}^{-1} x_{j,t} \right) \alpha(b, d, T, 2\delta, D)$, where $\bar{M}_{c_t-1}$ is the (conditional) average of the matrix M_{c_t-1} the algorithm uses for the update, where the random bits $|s_{i,t}|$ therein get replaced by their expectations $\prod_{h=1}^{i-1}(1 - \sigma(\Delta_{j_h,t}))$, which are the same as the quantities occurring in the one-time regret bound mentioned above. Summing over t, relying on standard inequalities, and bounding the effect of the delayed feedback inherent in the learning protocol (Lemma 6) concludes the proof. □

Remark 1. *The dependence on e^D above is common to all logistic bandit bounds,*[10] *and is due to the non-linear shape of $\sigma(\cdot)$ (see, e.g., Filippi et al. (2010); Gentile and Orabona (2012); Zhang et al. (2016); Li et al. (2017); Faury et al. (2020), where it takes the form of an upper bound on $1/\sigma'(\cdot)$). Also notice that D is meant to be a constant here. As for the dependence on the sequence length b, our bound has the form $\widetilde{O}(\sqrt{bT})$.*

Regret bound comparison. Many papers have tackled the problem of cascading bandits with contextual information, some of them adopting a linear model assumption (e.g., Zong et al. (2016); Li et al. (2016, 2019); Hiranandani et al. (2020)), others a generalized linear model assumption (e.g., Li and Zhang (2018); Liu et al. (2018a); Li (2019)). Most of these papers have been chiefly motivated by learning-to-rank tasks applied to recommendation problems. Our usage of cascading bandits may be motivated by widely different application domains, where the sequence J_t can potentially be far longer than the ranked list of items typically served to the user of an online content provider. So, we are interested in both the dependence on the time horizon T and the maximal length b. Our bound of the form $\sqrt{bT}$ improves on past results in contextual cascading bandits, where the dependence on b is either of the form $b\sqrt{T}$ (Zong et al. (2016); Li et al. (2019); Hiranandani et al. (2020)) or of the form $b\sqrt{bT}$ (Liu et al. (2018a)) or of the form $e^b + \sqrt{bT}$ (Li and Zhang (2018); Li (2019)) or even of the form $\frac{1}{p^*}\sqrt{bT}$ (Li et al. (2016)), where p^* is the smallest probability of any sequence of length b, which can easily be exponentially small in b, even in the case of independent outcomes considered here. Many of these results are specific to the "vanilla" scenario, which we recover as a special case of our setting. Recall that in the vanilla case there is no loss of generality in restricting to sequences of maximal length b_t, hence our improved regret guarantee directly applies to that case as well.

[10]This actually applies only to the so-called frequentist regret bounds, which are the ones considered here. Switching to a Bayesian regret guarantee allows one to give bounds which, under some conditions, are independent of D – see Dong et al. (2019). Staying within the realm of frequentist guarantees, it might be possible to improve Theorem 1 by following the more refined self-concordant analysis contained in Faury et al. (2020). This analysis allows one to move the multiplicative dependence on e^D from $\sqrt{T}$ to a lower order term in T.

4 DEPENDENT OUTCOMES

Starting from the parametric model of Section 2.2, we can write the conditional probabilities as

$$p(x_{j_{k+1}} \,|\, x_{j_1}, \ldots, x_{j_k}) \;=\; \sigma(\Delta_{j_1,\ldots,j_k,j_{k+1}}) ,$$

where $\Delta_{j_1} = c(x_{j_1})^\top u$ and

$$\Delta_{j_1,\ldots,j_k,j_{k+1}} = c(x_{j_{k+1}} \mid x_{j_1}, \ldots, x_{j_k})^\top u ,$$

for all $k \geq 1$. With this notation, and the function $E(\cdot, \ldots, \cdot)$ defined in (6), the expected regret (4) can be written as

$$\mathbb{E}_Y[R(J,Y)] = \begin{cases} E(\Delta_{j_1}, \ldots, \Delta_{j_1,\ldots,j_s}) & \text{if } J \neq \langle\rangle \\ \ell_0 & \text{otherwise} . \end{cases}$$

The algorithm operating with the above generative model is an adaptation of the one we presented for the independent case. The main difference here is that we use conditional probabilities computed from coverage difference vectors. Notice that calculating J^* may be computationally intractable. Yet, having at our disposal an oracle that maximizes $\mathbb{E}_Y[R(J,Y)]$ over J, we could clearly carry out a formal regret analysis similar to the one in Theorem 1. As in Hiranandani et al. (2020), we resort to a greedy algorithm to reduce the computational complexity. Specifically, we give an order over all candidate items based on their coverage difference vectors $c(\cdot \mid x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}})$ w.r.t. the already listed items. Then the empirical mean and upper confidence levels are computed based on these difference vectors, while the length of the sequence is chosen based on a search over all possible length values with the computed upper confidence levels.

Below we describe a simple greedy algorithm operating on true probabilities $p(x_{j_{k+1}} \mid x_{j_1}, \ldots, x_{j_k})$, and give the pseudocode of its bandit counterpart in Appendix B. The bandit version of this algorithm will be tested in our experimental comparison in Section 5.

For convenience, we drop subscript t. On the set of available actions A, the algorithm builds sequence $J_s = \langle x_{j_1}, x_{j_2}, \ldots, x_{j_s} \rangle$ of length $s \leq b_t$ as follows. For $k = 1, \ldots, s$, append to $\langle x_{j_1}, x_{j_2}, \ldots, x_{j_{k-1}} \rangle$ item

$$x_{j_k} = \arg \max_{x \in A \setminus \{x_{j_1}, \ldots, x_{j_{k-1}}\}} p(x \mid x_{j_1}, \ldots, x_{j_{k-1}}) \ . \quad (7)$$

The analysis bounds the *scaled* cumulative regret, also considered in previous work (e.g., Hiranandani et al. (2020)), where one-time regret is defined as

$$\mathbb{E}_Y[\gamma(s_t^*) R(J_t^*, Y)] - \mathbb{E}_Y[R(J_t, Y)] \, . \quad (8)$$

The analysis leverages the fact that the greedy algorithm gives an approximation ratio $0 < \gamma(s_t^*) < 1$, with some mild assumptions on rewards r_i and losses ℓ_i. Then by a result similar to Theorem 1 for the independent case, we can derive a regret bound of the form $\sqrt{\alpha(b, d', T, \delta, D) T} \, d' \log T$ – see Appendix B.

5 EXPERIMENTS

In order to demonstrate the efficacy of the proposed algorithms, we present our experimental results on ranking tasks defined on the Million Songs Bertin-Mahieux et al. (2011), Yelp Inc. and Crawford (2017), MovieLens-25M Harper and Konstan (2015), and MNIST LeCun et al. (1998) datasets. We compare our algorithms (Algorithm 1, called "Independent" here, abbreviated as "Ind", and Algorithm 3 in Appendix B, called "Dependent", abbreviated as "Dep") to a number of exploration-exploitation baselines in the cascading bandits literature,[11] specifically to the CascadeLinUCB algorithm of Zong et al. (2016) (called "C-UCB" later on), the GL-CDCM algorithm of Liu et al. (2018b) which relies on a generalized linear model with the original Maximum Likelihood Estimator (MLE) as in Filippi et al. (2010), an ϵ-greedy version of our Algorithm 2 (called "Eps" later on), a purely random policy (called "Rand" later on), and two more baselines obtained from Ind and Dep by reversing the order in which the items are presented. It has been suggested Kveton et al. (2015a); Combes et al. (2015) that reversing the order may have the advantage of speeding up learning since it allows the system to gather more feedback on low quality items. We call these two inverse ranking baselines "Ind-Inv" and "Dep-Inv", respectively.

Datasets and preprocessing. We describe preprocessing on MovieLens-25M. The Million Songs and Yelp datasets have been treated similarly. MovieLens-25M contains ratings of $59,047$ movies by $162,541$ users, and is popularly studied in the recommendation system literature. We sample $10,000$ movies at random and calculate the SVD of the corresponding $162,541 \times 10,000$ ratings matrix into 10 principal components. The projection matrices from the SVD are used to compute embeddings of dimension $d = 10$ for the remaining $49,047$ movies for training the bandit algorithms. The embeddings are normalized to unit L_2-norm and the dataset is shuffled randomly. In every round of bandit learning, the algorithm is presented with a non-overlapping chunk of movies as arms (A_t). The chunk size is 100 (except for the last one, which is of size 47). The outcome of an arm is decided by the mean rating received by the corresponding movie in the dataset. If this mean rating is greater than its median value in the dataset the outcome is a success, else is a failure. For Dep (and Dep-Inv) the 49,047 SVD-projected d-dimensional vectors have been used to compute coverage vectors through a Gaussian Mixture Model (GMM) with d' centroids.

As for MNIST, this is a multi-class classification dataset. We designed 10 ranking tasks out of it, one for each of the 10 classes in the dataset. Each task has one class as the "pivot-class" giving success, all other classes yielding failure. The algorithm must rank a collection of samples to have an item of the pivot-class (if present) as high up in the list as possible. See Appendix D for further details.

Scenarios. We study two reward/loss scenarios. The first one, called *"Vanilla"*, is designed to reproduce the standard setting studied in the traditional cascading bandit literature: $r_{j,t} = 1$ and $\ell_{j,t} = 0$, for all t and $j = 1, 2, \ldots, b_t$. The second one, called *"Exponential"*, is comprised of exponentially decaying rewards and losses, and is designed to incentivize early success: $r_{j,t} = \frac{1}{2^{j-1}}$, and $\ell_{j,t} = \frac{4}{5} \times \frac{1}{2^j} - 1$, for all t and $j = 0, 1, \ldots, b_t$. Notice that in this case $r_{1,t} = 1$ and

[11] This set of baselines are collectively a good pool of representatives of the relevant literature. In particular, we do not compare to traditional learning to rank methods that do not rely on exploration/exploitation, since the partial information structure of our problem would make this comparison somewhat questionable.

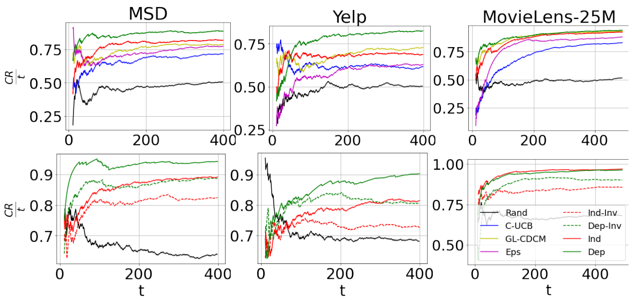

Figure 2: Average cumulative reward $CR(t)/t$ as a function of $t = 1, \ldots, T$ for the various algorithms on Million Songs Dataset (MSD) (1st column), Yelp (2nd column), and MovieLens-25M (3rd column). Vanilla scenarios (with $b_t = 1$) are in the top row, exponential scenarios (with $b_t = 10$) are in the bottom row. Notice that when $b_t = 1$, Ind = Ind-Inv and Dep = Dep-Inv, so only 6 curves are displayed on the top plots. Also, for exponential rewards (bottom row), we do not have C-UCB, GL-CDCM and Eps baselines as they are only defined for the vanilla reward scenario. Dep (green curve) performs best across the datasets, with an exception of MovieLens-25M, where Ind (red curve) performs comparably.

$\ell_{0,t} = -0.2$. The exponential scenario captures the true essence of the proposed models since it remains sensitive to early success even for larger budgets.

Tuning of Hyperparameters. We run a fine grid-search over the space of hyperparameters of each algorithm and only report the results corresponding to the combination of hyperparameters that obtains the largest final cumulative reward. We search the value of learning rate η in the range $1.0 - 100.0$, exploration parameter α on a logarithmic scale in the range $10^{-9} - 10.0$, ϵ in ϵ-greedy in $0.01 - 0.5$, L2 regularization weight λ in our implementation of the GL-CDCM baseline Liu et al. (2018b) on a logarithmic scale in the range $10^{-7} - 10^3$ and the number d' of latent components for the proposed dependent algorithm between 3 and 30.

Results. Figure 2 and Figure 3 contain some elements of our experimental comparison among all algorithms, further results are contained in Appendix D. We evaluate the algorithms in terms of their time-averaged Cumulative Reward (CR) obtained over all rounds of training by computing, for each algorithm, the fraction of reward/loss units accumulated per time step, up to time t, for $t = 1, \ldots, T$. If a given dataset has T chunks then each algorithm is trained for exactly T rounds. Figure 2 shows the variation of $\frac{CR(t)}{t}$ over rounds of training for two relevant scenarios. In the exponential scenario, we restrict to comparing Ind, and Dep to Rand, Ind-Inv and Dep-Inv, since the other baselines are not designed to cope with it.

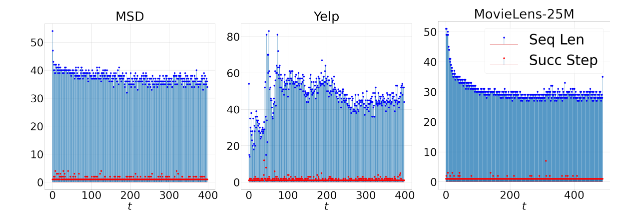

Figure 3: Ind operating on the three datasets MSD (left), Yelp (middle), and MovieLens (right) in the exponential scenario with $b_t = 100$. The plots report, for each chunk of the datasets (x-axis), the length $\widehat{s}_t$ of the sequence J_t computed by Ind ("Seq Len") along with the position where the first success is observed ("Succ Step"), that is, value $\widehat{s}'_t$ for J_t (y-axis) – please recall the notation in Algorithm 1. Chunks where success is not observed are excluded. The algorithm never saturates budget b_t, while achieving success within the first few items ($\widehat{s}'_t \leq 12$ for all t where success is achieved).

Notice that for small b_t in the vanilla scenario and all values of b_t in the exponential scenario, achieving higher CR is synonymous of early success, and we observe that the proposed dependent algorithm ("Dep") outperforms the other algorithms in these scenarios, an exception being MovieLens-25M, where the proposed Independent ("Ind") algorithm performs comparably. The "Inv" versions of Ind and Dep do not happen to be strong competitors, but perhaps on MNIST in the vanilla scenario (Table 7 in Appendix D).

Figure 3 reports the inner behavior of Ind on three datasets when deciding on the actual length of sequence J_t on each chunk. The algorithm never saturates the budget length. Further results are provided in the appendix, where similar trends as those reported here can be observed. In particular, Appendix D reports a more extensive study on how the performances in the two scenarios vary with different choices of b_t.

6 CONCLUSIONS

We have introduced a cascading bandit model with flexible sequences and varying rewards and losses. The model is specifically focused on applications, like web search or payment systems, where the item sequence can be significantly long. We have analyzed two algorithms with improved regret guarantees, and have empirically demonstrated their competitiveness against a number of baselines on popular real-world datasets.

References

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

Branislav Kveton, Csaba Szepesvari, Zheng Wen, and

Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *International Conference on Machine Learning*, pages 767–776. PMLR, 2015a.

S. Li, B. Wang, S. Zhang, and W. Chen. Contextual combinatorial cascading bandits. In *Proc. of the 33rd International Conference on Machine Learning*, volume 48. JMLR: W&CP, 2016.

S. Zong, H Ni, K. Sung, N. R. Ke, Z. Wen, and B. Kveton. Cascading bandits for large-scale recommendation problems. In *Proc. of The 31th Uncertainty in Artificial Intelligence Conference*, 2016.

S. Li and S. Zhang. Online clustering of contextual cascading bandits. In *Proc. of the 32nd AAAI Conference on Artificial Intelligence*, pages 3554–3561, 2018.

Wang Chi Cheung, Vincent Tan, and Zixin Zhong. A thompson sampling algorithm for cascading bandits. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 438–447. PMLR, 2019.

Masrour Zoghi, Tomas Tunys, Mohammad Ghavamzadeh, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. Online learning to rank in stochastic click models. In *International Conference on Machine Learning*, pages 4199–4208. PMLR, 2017.

Tor Lattimore, Branislav Kveton, Shuai Li, and Csaba Szepesvári. Toprank: a practical algorithm for online stochastic ranking. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 3949–3958, 2018.

S. Li, T. Lattimore, and C. Szepesvari. Online learning to rank with features. In *Proc. of the 36th International Conference on Machine Learning*, volume 97, pages 3856–3865. PMLR, 2019.

Chang Li and Maarten De Rijke. Cascading non-stationary bandits: Online learning to rank in the non-stationary cascade model. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019.

G. Hiranandani, H. Singh, P. Gupta, I. A. Burhanuddin, Z. Wen, and B. Kveton. Cascading linear submodular bandits: Accounting for position bias and diversity in online learning to rank. In *Proc. of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115, pages 722–732. PMLR, 2020.

Richard Combes, Stefan Magureanu, Alexandre Proutiere, and Cyrille Laroche. Learning to rank: Regret lower bound and efficient algorithms. In *SIGMETRICS*, 2015.

Y. Yue and C. Guestrin. Linear submodular bandits and their application to diversified retrieval. In *Proc. NIPS*, pages 2483–2491, 2011.

C. Gentile and F. Orabona. On multilabel classification and ranking with partial feedback. In *Advances in Neural Information Processing Systems*, volume 25, pages 1151–1159. Curran Associates, Inc., 2012.

W. Liu, S. Li, and S. Zhang. Contextual dependent click bandit algorithm for web recommendation. In *Proc. International Computing and Combinatorics Conference*, pages 39–50. Springer, 2018a.

S. Li. Online clustering of contextual cascading bandits. In *arXiv:1711.08594*, 2019.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Mach. Learn.*, 69(2-3):169–192, 2007.

Shi Dong, Tengyu Ma, and Benjamin Van Roy. On the performance of thompson sampling on logistic bandits. In *Conference on Learning Theory*, pages 1158–1160. PMLR, 2019.

Louis Faury, Marc Abeille, Clément Calauzènes, and Olivier Fercoq. Improved optimistic algorithms for logistic bandits. In *37th ICML*, 2020.

S. Filippi, O. Cappe', A. Garivier, and C. Szepesvari. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pages 586–594, 2010.

Lijun Zhang, Tianbao Yang, Rong Jin, Yichi Xiao, and Zhi-Hua Zhou. Online stochastic linear optimization under one-bit feedback. In *International Conference on Machine Learning*, pages 392–401. PMLR, 2016.

Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In *International Conference on Machine Learning*, pages 2071–2080. PMLR, 2017.

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

Yelp Inc. and Chris Crawford. Yelp dataset challenge. https://www.kaggle.com/yelp-dataset/yelp-dataset, 2017. Accessed: 2021-05-28.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM transactions on interactive intelligent systems (TIIS)*, 5(4): 1–19, 2015.

Yann LeCun, Corinna Cortes, and Christopher Burges. Mnist handwritten digit database. http://yann.lecun.com/exdb/mnist, 1998. Accessed: 2021-05-28.

Weiwen Liu, Shuai Li, and Shengyu Zhang. Contextual dependent click bandit algorithm for web recommendation. In *International Computing and Combinatorics Conference*, pages 39–50. Springer, 2018b.

Sham M. Kakade and Ambuj Tewari. On the generalization ability of online strongly convex programming algorithms. In *Advances in Neural Information Processing Systems 21*, pages 801–808. Curran Associates, Inc., 2008.

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. A second-order perceptron algorithm. *SIAM J. Comput.*, 34(3):640–668, 2005.

Joel Tropp. Freedman's inequality for matrix martingales. In *arXiv:1101.3039*, 2011.

Nicolò Cesa-Bianchi and Claudio Gentile. Improved risk tail bounds for on-line algorithms. *IEEE Trans. Inf. Theory*, 54(1):386–390, 2008.

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvári. Combinatorial cascading bandits. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 1450–1458, 2015b.

Ruida Zhou, Chao Gan, Jing Yang, and Cong Shen. Cost-aware cascading bandits. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3228–3234, 2018.

ZZ, Qinshi Wang, and Wei Chen. Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications. In *Neural Information Processing Systems. arXiv: 1703.01610. url: http://arxiv.org/ abs/1703.01610*, 2018.

Sho Takemori, Masahiro Sato, Takashi Sonoda, Janmajay Singh, and Tomoko Ohkuma. Submodular bandit problem under multiple constraints. In *Conference on Uncertainty in Artificial Intelligence*, pages 191–200. PMLR, 2020.

Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 169–178, 2011.

Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *International Conference on Machine Learning*, pages 1453–1461. PMLR, 2013.

Nicolò Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Delay and cooperation in nonstochastic bandits. *Journal of Machine Learning Research*, 20 (17):1–38, 2019.

Ciara Pike-Burke, Shipra Agrawal, Csaba Szepesvari, and Steffen Grunewalder. Bandits with delayed, aggregated anonymous feedback. In *International Conference on Machine Learning*, pages 4105–4113. PMLR, 2018.

Zhengyuan Zhou, Renyuan Xu, and Jose Blanchet. Learning in generalized linear contextual bandits with stochastic delays. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Sakshi Arya and Yuhong Yang. Randomized allocation with nonparametric estimation for contextual multi-armed bandits with delayed rewards. *Statistics & Probability Letters*, 164:108818, 2020.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

Supplementary Material: Learning to Plan Variable Length Sequences of Actions with a Cascading Bandit Click Model of User Feedback

A APPENDIX

Algorithm 2 contains a more detailed version of Algorithm 1 of Section 3 where, in particular, we replace the exploration parameter α with an exact expression $\alpha(b, d, T, \delta, D)$ needed for the analysis. Algorithm 2 also includes for technical reasons a projection step at the end. Specifically, the update of vector $w_{c_t+j-1} \to w_{c_t+j}$ is done by first projecting w_{c_t+j-1} onto the set $\{w \in \mathbb{R}^d : |w^\top x_j| \le D\}$ to obtain w'_{c_t+j-1}, and then by computing a standard Newton step. The projection can be efficiently calculated in closed form (see the proof of Lemma 5 below).

The following lemma shows that, in the independent case, the Bayes optimal sequence can be computed efficiently.

Lemma 1. *Let $p_Y(A) = \prod_{j=1}^{|A|} p(x_j)$, and b be the budget length. Then J^* can be computed as follows. Set*

$$s^* = \arg\max_{s=0,1,\ldots,b} E_Y[R(J^*_s, Y)] \ ,$$

where $J^*_s = \langle x_{j^*_1}, x_{j^*_2}, \ldots, x_{j^*_s} \rangle$, $x_{j^*_1}, x_{j^*_2}, \ldots, x_{j^*_s}$ the items associated with the s largest marginal probabilities $p(x_j)$, $x_j \in A$, sorted in non-increasing order. Then $J^* = J^*_{s^*}$, with $J^* = \langle\rangle$ if $s^* = 0$.

Proof. Consider the following argument.

1. Let $J = \langle x_{j_1}, x_{j_2}, \ldots, x_{j_k}, \ldots, j_{k'}, \ldots, x_{j_s} \rangle$, be an *arbitrary* sequence, and let a perturbed sequence $J' = \langle x_{j_1}, x_{j_2}, \ldots, x_{j_{k'}}, \ldots, x_{j_k}, \ldots, x_{j_s} \rangle$ be obtained from J just by swapping x_{j_k} with $x_{j_{k'}}$. Moreover, suppose $p(x_{j_{k'}}) > p(x_{j_k})$. Then considering the difference $\mathbb{E}_Y[R(J', Y)] - \mathbb{E}_Y[R(J, Y)]$ and relying on the fact that rewards r_j are non-decreasing, we want to show that $\mathbb{E}_Y[R(J', Y)] \ge \mathbb{E}_Y[R(J, Y)]$. It suffices to show the claim for the case where x_{j_k} and $x_{j_{k'}}$ are adjacent in J, so that $k' = k + 1$.

   Let us introduce the short-hand notation $p_i = p(x_{j_i})$, and $\Pi = \prod_{i=1}^{k-1}(1 - p_i)$. Our assumption then becomes $p_{k+1} \ge p_k$. Now, since Y's components are independent, $\mathbb{E}_Y[R(J', Y)]$ has the form of function $E(\cdot, \ldots, \cdot)$ defined in Lemma 2. Then, because k and $k+1$ are adjacent positions, one can easily verify that, removing common terms, the difference $\mathbb{E}_Y[R(J', Y)] - \mathbb{E}_Y[R(J, Y)]$ can be written as

   $$\begin{aligned} \mathbb{E}_Y[R(J', Y)] - \mathbb{E}_Y[R(J, Y)] &= \Pi \left[r_k \left(p_{k+1} - p_k\right) + r_{k+1} \Big(p_k(1 - p_{k+1}) - p_{k+1}(1 - p_k) \Big) \right] \\ &= \Pi \left(r_k - r_{k+1}\right)(p_{k+1} - p_k) \end{aligned}$$

   which is non-negative, since $\Pi \ge 0$, $r_k \ge r_{k+1}$ and $p_{k+1} \ge p_k$.

   As the above argument holds for an arbitrary starting sequence J, this shows that, for any given (unordered) set of items contained in J, the best way to sort them in order to maximize $\mathbb{E}_Y[R(J, Y)]$ is to have then in non-increasing order of their marginal probabilities $p(x_j)$.

2. Next, let $J = \langle x_{j_1}, x_{j_2}, \ldots, x_{j_k}, \ldots, x_{j_s} \rangle$, be an *arbitrary* sequence, and let a perturbed sequence $J'' = \langle x_{j_1}, x_{j_2}, \ldots, x_{j_{k''}}, \ldots, x_{j_s} \rangle$ be obtained from J just by *replacing* item x_{j_k} by $x_{j_{k''}}$, where $p(x_{j''_k}) \ge p(x_{j_k})$. Again, we need to show that $\mathbb{E}_Y[R(\langle J'', Y)] \ge \mathbb{E}_Y[R(J, Y)]$. This claim immediately follows from the monotonicity property contained in Lemma 2, thereby showing that, for any given length s, the best assortment of items in J is one that contains those corresponding to the s largest marginal probabilities $p(x_j)$. In turn, combined with the previous item, this implies that J^* has necessarily the form $J^*_s = \langle x_{j^*_1}, x_{j^*_2}, \ldots, x_{j^*_s} \rangle$, for some length $s \in \{1, \ldots, b_t\}$, where $x_{j^*_1}, x_{j^*_2}, \ldots, x_{j^*_s}$ are the items associated with the s largest marginal probabilities $p(x_j)$, sorted in non-increasing order.

Algorithm 2 The contextual bandit algorithm in the independent case with link function $\sigma(x) = \frac{\exp(x)}{1+\exp(x)}$.

Input: Confidence level $\delta \in [0,1]$, width parameter $D > 0$, maximal budget parameter $b > 0$
Init: $M_0 = bI \in \mathbb{R}^{d\times d}$, $w_1 = 0 \in \mathbb{R}^d$, $c_1 = 1$
For $t = 1, 2, \ldots, T$

1. Get:
   - Set of actions $A_t = \{x_{1,t}, \ldots, x_{|A_t|,t}\} \subseteq \{x \in \mathbb{R}^d \,:\, ||x|| \leq 1\}$,
   - budget $b_t \leq b$;
2. For $x_j \in A_t$, set $\hat{\Delta}_{j,t} = x_j^\top w_{c_t}$;
3. Compute J_t :
   - Let $\widehat{J}_{t,s} = \langle x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,s}} \rangle$ be made of the s largest items in A_t in non-increasing order of $\widehat{p}_{j,t}$, where:
     - $\widehat{p}_{j,t} = \sigma(\hat{\Delta}_{j,t} + \epsilon_{j,t})$,
     - $\epsilon_{j,t}^2 = (x_j^\top M_{c_t-1}^{-1} x_j)\ \alpha(b,d,T,\delta)$, with

$$\alpha(b,d,T,\delta,D) = 2bD^2 + \left(\frac{c_\sigma}{c_{\sigma'}}\right)^2 d\log\left(1 + \frac{2}{b}\Big(T\,\frac{c_\sigma}{1-c_\sigma} + 4\log\frac{4(T+1)}{\delta}\Big)\right) + 2\left(12\left(\frac{c_\sigma}{c_{\sigma'}}\right)^2 + \frac{36(1+D)}{c_{\sigma'}}\right)\log\frac{2b(T+4)}{\delta} + 20D^2\log\frac{2bd(T+1)}{\delta}$$

   - Set $\widehat{s}_t = \arg\max_{s=0,1,\ldots,b_t} \widehat{\mathbb{E}}_{Y_t}[R(\widehat{J}_{t,s}, Y_t)]$,
     with

$$\widehat{\mathbb{E}}_{Y_t}[R(\widehat{J}_{t,s}, Y_t)] = \begin{cases} E\left(\hat{\Delta}_{\widehat{j}_{t,1},t} + \epsilon_{\widehat{j}_{t,1},t}, \ldots, \hat{\Delta}_{\widehat{j}_{t,s},t} + \epsilon_{\widehat{j}_{t,s},t}\right) & \text{if } s \geq 1 \\ \ell_{0,t} & \text{otherwise} \end{cases} ,$$

     where

$$E(\Delta_1, \Delta_2, \ldots, \Delta_s) = r_{1,t}\,\sigma(\Delta_1) + \ldots + r_{s,t}\,\sigma(\Delta_s)\prod_{i=1}^{s-1}(1-\sigma(\Delta_i)) + \ell_{s,t}\prod_{i=1}^{s}(1-\sigma(\Delta_i)) \ ;$$

   - Finally, $J_t = \widehat{J}_{t,\widehat{s}_t}$;
4. Observe feedback $Y_t \downarrow J_t = \begin{cases} \langle y_{t,\widehat{j}_{t,1}}, y_{t,\widehat{j}_{t,2}}, \ldots, y_{t,\widehat{j}_{t,\widehat{s}'_t}} \rangle = \langle 0, \ldots, 0, 1 \rangle, & \text{for some } \widehat{s}'_t \leq \widehat{s}_t \quad \text{or} \\ \langle y_{t,\widehat{j}_{t,1}}, y_{t,\widehat{j}_{t,2}}, \ldots, y_{t,\widehat{j}_{t,\widehat{s}_t}} \rangle = \langle 0, \ldots, 0, 0 \rangle \end{cases}$
5. **For** $j = 1, \ldots, \widehat{s}_t$ (in the order of occurrence of items in J_t) update:

$$M_{c_t+j-1} = M_{c_t+j-2} + |s_{j,t}| x_j x_j^\top, \qquad w_{c_t+j} = w'_{c_t+j-1} + \frac{1}{c_{\sigma'}} M_{c_t+j-1}^{-1} \nabla_{j,t} \ ,$$

   where

$$s_{j,t} = \begin{cases} 1 & \text{If } y_{t,j} \text{ is observed and } y_{t,j} = 1 \\ -1 & \text{If } y_{t,j} \text{ is observed and } y_{t,j} = 0 \\ 0 & \text{If } y_{t,j} \text{ is not observed ,} \end{cases}$$

   and $\nabla_{j,t} = \sigma(-s_{j,t}\,\hat{\Delta}'_{j,t})\, s_{j,t}\, x_j$, where $\hat{\Delta}'_{j,t} = x_j^\top w'_{c_t+j-1}$
   with

$$w'_{c_t+j-1} = \arg\min_{w\,:\,-D \leq w^\top x_j \leq D} d_{c_t+j-2}(w, w_{c_t+j-1}) \ ;$$

6. $c_{t+1} \leftarrow c_t + \widehat{s}_t$.

3. What remains is to maximize over length $s \in \{0, 1, \ldots, b\}$. Notice that there is no guarantee that, viewed as a function of s, the quantity $E_Y[R(J_s^*, Y)]$ will have a specific behavior, like unimodality. Hence, we need to try out all allowed values of $s \le b$, including $s = 0$.

This concludes the proof. □

The next lemma is of preliminary importance. It delivers a monotonicity property showing that the upper confidence scheme adopted in Algorithm 2 is properly defined, but it also serves in the proof of subsequent lemmas.

Lemma 2. *For constants $r_1 \ge r_2 \ldots \ge r_s > 0$, $\ell_s < 0$, and a differentiable function $p : \mathbb{R} \to [0,1]$ which is monotonically increasing, the function $E : \mathbb{R}^s \to \mathbb{R}$ defined as*

$$E(\Delta_1, \Delta_2, \ldots, \Delta_s) = r_1\, p(\Delta_1) + r_2\, p(\Delta_2)(1 - p_(\Delta_1)) + \ldots + r_s\, p(\Delta_s) \prod_{i=1}^{s-1}(1 - p(\Delta_i)) + \ell_s \prod_{i=1}^{s}(1 - p(\Delta_i)) \quad (9)$$

enjoys the following properties:

1. *E is non-decreasing in each individual variable Δ_i.*
2. *If, in addition, $r_i \in [0,1]$, for $i = 1, \ldots, s$, $\ell_s \in [-1, 0]$, and $\frac{dp(\Delta)}{d\Delta} \le z$ for all $\Delta \in \mathbb{R}$, then $\frac{\partial E(\Delta_1,\ldots,\Delta_s)}{\partial \Delta_i} \le z(r_i - \ell_s) \le 2z$ holds for all $\Delta_1, \ldots, \Delta_s \in \mathbb{R}$, and i.*
3. *Under the same assumption as in item 2 above,*

$$\frac{\partial E(\Delta_1, \ldots, \Delta_s)}{\partial \Delta_k} \le 2z \prod_{j=1}^{k-1}(1 - p(\Delta_j)) \ .$$

Proof. Define, for $k = 1, \ldots, s$,

$$\begin{aligned} E_k &= E_k(\Delta_k, \Delta_{k+1}, \ldots, \Delta_s) \\ &= r_k\, p(\Delta_k) + r_{k+1}\, p(\Delta_{k+1})(1 - p(\Delta_k)) + \ldots + r_s\, p(\Delta_s) \prod_{i=k}^{s-1}(1 - p(\Delta_i)) + \ell_s \prod_{i=k}^{s}(1 - p(\Delta_i)) \ , \end{aligned}$$

and notice that

$$\begin{aligned} E_k &\le r_k \left(p(\Delta_k) + p(\Delta_{k+1})(1 - p(\Delta_k)) + \ldots + p(\Delta_s) \prod_{i=k}^{s-1}(1 - p(\Delta_i)) \right) + \ell_s \prod_{i=k}^{s}(1 - p(\Delta_i)) \\ &\quad \text{(due to the fact that } r_s \le r_{s-1} \le \ldots \le r_{k+1} \le r_k) \\ &\le r_k \left(p(\Delta_k) + p(\Delta_{k+1})(1 - p(\Delta_k)) + \ldots + p(\Delta_s) \prod_{i=k}^{s-1}(1 - p(\Delta_i)) + \prod_{i=k}^{s}(1 - p(\Delta_i)) \right) \\ &\quad \text{(since } \ell_s \le 0 \le r_k) \\ &= r_k \\ &\quad \text{(since the expression in braces equals 1)} \ . \end{aligned}$$

Then we have, for $k \ge 2$

$$\begin{aligned} E_{k-1} &= \underbrace{(1 - p(\Delta_{k-1}))}_{\ge 0}\, E_k + r_{k-1}\, p(\Delta_{k-1}) \quad (10) \\ &= p(\Delta_{k-1})\, \underbrace{(r_{k-1} - E_k)}_{\ge r_{k-1} - r_k \ge 0} + E_k \ . \quad (11) \end{aligned}$$

From (11) one can see that, viewed solely as a function of Δ_{k-1}, the quantity E_{k-1} can be seen as a positive constant times $p(\Delta_{k-1})$ (since $r_{k-1} - E_k \ge 0$ and E_k only depends on variables $\Delta_k, \ldots, \Delta_s$) plus a constant term independent of Δ_{k-1} (again, because E_k only depends on $\Delta_k, \dots, \Delta_s$). We can now proceed by backward induction on $k = s, s-1, \dots, 1$. For $k = s$ we have $E_s = \ell_s(1 - p(\Delta_s))$ which is non-decreasing in Δ_s since so is $p(\cdot)$, and $\ell_s < 0$. Assuming by induction E_k is non-decreasing in $\Delta_k, \dots, \Delta_s$, we have from (11) that E_{k-1} is non-decreasing in Δ_{k-1}, thanks to the fact that $p(\Delta_{k-1})$ is monotonically increasing in Δ_{k-1}, E_k only depends on $\Delta_k \dots, \Delta_s$, and $r_{k-1} - E_k \geq 0$. Moreover, E_{k-1} is also non-decreasing in $\Delta_k, \dots, \Delta_s$ since, from (10), E_{k-1} is a positive constant (i.e., independent of $\Delta_k, \dots, \Delta_s$) times E_k plus a constant term, again independent of $\Delta_k, \dots, \Delta_s$. Since by induction E_k is non-decreasing in $\Delta_k, \dots, \Delta_s$, so is E_{k-1}.

The above holds for all k, hence it holds in particular for $k = 1$, which concludes the proof of the first part.

As for the second part, we again proceed by backward induction on $k = s, s-1, \dots, 1$. We have $\frac{\partial E_s(\Delta_s)}{\partial \Delta_s} = -\ell_s \frac{\partial p(\Delta_s)}{\Delta_s} \leq z(-\ell_s) \leq z(r_s - \ell_s)$ for all Δ_s. Then assume by the inductive hypothesis that $\frac{\partial E_k(\Delta_k, \dots, \Delta_s)}{\partial \Delta_i} \leq z(r_i - \ell_s)$ for all $\Delta_k, \dots, \Delta_s$, and $i = k, \dots, s$. From (11), we can write

$$\frac{\partial E_{k-1}(\Delta_{k-1}, \dots, \Delta_s)}{\partial \Delta_{k-1}} = \frac{\partial p(\Delta_{k-1})}{\partial \Delta_{k-1}}(r_{k-1} - E_k) \leq z(r_{k-1} - \ell_s) \leq 2z \ , \tag{12}$$

the first inequality deriving from $E_k \geq \ell_s$. On the other hand, from (10) we also have, for $i = k, \dots, s$,

$$\frac{\partial E_{k-1}(\Delta_i, \dots, \Delta_s)}{\partial \Delta_i} = (1 - p(\Delta_{k-1}))\frac{\partial E_k(\Delta_k, \dots, \Delta_s)}{\partial \Delta_i} \leq \frac{\partial E_k(\Delta_k, \dots, \Delta_s)}{\partial \Delta_i} \leq z(r_i - \ell_s) \ ,$$

the inequality following from the inductive hypothesis.

Again, the above holds for all k, hence it holds for $k = 1$, which concludes the proof of the second part.

Finally, as for the third part, we first observe that, for any k,

$$\frac{\partial E(\Delta_1, \dots, \Delta_s)}{\partial \Delta_k} = \prod_{j=1}^{k-1}(1 - p(\Delta_j))\, \frac{\partial E_k(\Delta_k, \dots, \Delta_s)}{\partial \Delta_k} \ ,$$

and then apply the bound $\frac{\partial E_k(\Delta_k, \dots, \Delta_s)}{\partial \Delta_k} \leq 2z$ from (12) to obtain the claimed result. □

The next two lemmas will be the basis for our regret analysis.

Lemma 3. *Let us assume the independence model for outcome Y. Then, for given set of actions A, and budget b, let J^* be the Bayes optimal sequence and $J = \langle x_{j_1}, \dots, x_{j_s} \rangle$ be the sequence computed by Algorithm 2 on A and b, with link function σ such that $\sigma'(\Delta) \leq z$ for all $\Delta \in \mathbb{R}$. Further, let $\Delta_j = u^\top x_j$, and $\hat{\Delta}_j = w^\top x_j$, for all $x_j \in A$, and assume $|\Delta_j - \hat{\Delta}_j| \leq \epsilon_j$ for all j such that $x_j \in A$, where w is the vector used by Algorithm 2 to compute J. Then the one-time regret $\mathbb{E}_Y[R(J^*, Y)] - \mathbb{E}_Y[R(J, Y)]$ can be bounded as follows:*

$$\mathbb{E}_Y[R(J^*, Y)] - \mathbb{E}_Y[R(J, Y)] \leq \begin{cases} 4z \sum_{i=1}^{s} \epsilon_{j_i} \prod_{h=1}^{i-1}(1 - \sigma(\Delta_{j_h})) & \text{if } J \neq \langle\rangle \\ 0 & \text{otherwise .} \end{cases} \tag{13}$$

Proof. Irrespective of whether $J \neq \langle\rangle$ or $J^* \neq \langle\rangle$, we can write

$$\begin{aligned}
\mathbb{E}_Y[R(J^*, Y)] \quad - \quad & \mathbb{E}_Y[R(J, Y)] \\
\leq \quad & \widehat{\mathbb{E}}_Y[R(J^*, Y)] - \mathbb{E}_Y[R(J, Y)] \\
& \text{(using the first part of Lemma 2 combined with the condition } |\Delta_j - \hat{\Delta}_j| \leq \epsilon_j) \\
\leq \quad & \widehat{\mathbb{E}}_Y[R(J, Y)] - \mathbb{E}_Y[R(J, Y)] \\
& \text{(since, by definition of } J,\ \widehat{\mathbb{E}}_Y[R(J^*, Y)] \leq \widehat{\mathbb{E}}_Y[R(J, Y)]. \\
& \text{Notice that this implies that in the case where our algorithm happens to play } J = \langle\rangle \\
& \text{the regret is} \leq 0). \\
= \quad & E(\hat{\Delta}_{j_1} + \epsilon_{j_1}, \dots, \hat{\Delta}_{j_s} + \epsilon_{j_s}) - E(\Delta_{j_1}, \dots, \Delta_{j_s}) \\
& \text{(where } E(\cdot) \text{ is defined in (9))} \\
\leq \quad & E(\Delta_{j_1} + 2\epsilon_{j_1}, \dots, \Delta_{j_s} + 2\epsilon_{j_s}) - E(\Delta_{j_1}, \dots, \Delta_{j_s}) \\
& \text{(using again the first part of Lemma 2 together with } |\Delta_j - \hat{\Delta}_j| \leq \epsilon_j) \ .
\end{aligned}$$

Now, by the mean-value theorem, we can write

$$E(\Delta_{j_1}+2\epsilon_{j_1},\ldots,\Delta_{j_s}+2\epsilon_{j_s})-E(\Delta_{j_1},\ldots,\Delta_{j_s}) = 2\sum_{i=1}^{s}\frac{\partial E(\Delta_{j_1},\ldots,\Delta_{j_s})}{\partial \Delta_{j_i}}\Big|_{\Delta_{j_1}=\xi_{j_s},\ldots,\Delta_{j_s}=\xi_{j_s}}\,\epsilon_{j_i}\ ,$$

where $\xi_{j_i}\in(\Delta_{j_i},\Delta_{j_i}+2\epsilon_{j_i})$, for $i\in[s]$. The third part of Lemma 2 then allows us to write

$$\begin{aligned}\frac{\partial E(\xi_{j_1},\ldots,\xi_{j_s})}{\partial \Delta_{j_i}} &\le 2z\,(1-\sigma(\xi_{j_1}))\ldots(1-\sigma(\xi_{j_{i-1}}))\\ &\le 2z\,(1-\sigma(\Delta_{j_1}))\ldots(1-\sigma(\Delta_{j_{i-1}}))\ ,\end{aligned}$$

the second inequality deriving from the monotonicity of $\sigma(\cdot)$ and the fact that $\xi_{j_i}\in(\Delta_{j_i},\Delta_{j_i}+2\epsilon_{j_i})$. Replacing back, and summing over i yields the claimed bound. □

In order to quantify ϵ_j in Lemma 3, we introduce a suitable surrogate loss function $L(\cdot)$ that determines the dynamics of the algorithm (i.e., the proposed update rule being an online Newton step w.r.t. to this loss function), along with its convergence guarantees. In the proofs that follow we set

$$L(\Delta)=\log(1+e^{-\Delta})\ .$$

Notice that $\sigma(\Delta)=-L'(-\Delta)$.

Lemma 4. *Consider any item $x_{j_i}\in A$, and the random variable $s_{j_i}\in\{-1,0,1\}$ whose value is given in the algorithm's pseudocode. Also, assume x_{j_i} occurs in the i-th position of sequence $J=\langle x_{j_1},x_{j_2},\ldots,x_{j_s}\rangle$. Let c_σ and $c_{\sigma'}$ be two positive constants such that, for all $\Delta\in[-D,D]$ we have $|L'(\Delta)|\le c_\sigma$ and $L''(\Delta)\ge c_{\sigma'}$. Set $\Delta_{j_i}=u^\top x_{j_i}$. Then, for any $\hat\Delta'_{j_i}\in\mathbb{R}$ we have*

$$0\le \textsc{var}[L(s_{j_i}\hat\Delta_{j_i})-L(s_{j_i}\Delta_{j_i})\,|\,J]\le\frac{2(c_\sigma)^2}{c_{\sigma'}}\,\mathbb{E}[L(s_{j_i}\hat\Delta_{j_i})-L(s_{j_i}\Delta_{j_i})\,|\,J]\ .$$

Proof. Let us introduce the shorthands

$$\Delta_j=u_j^\top x,\qquad p_{j_i}=\sigma(\Delta_{j_i}),\qquad \Pi_{i-1}=(1-\sigma(\Delta_{j_1}))\ldots(1-\sigma(\Delta_{j_{i-1}}))\ .$$

We can write

$$\begin{aligned}\mathbb{P}(s_{j_i}=1\,|\,J)&=\Pi_{i-1}\,p_{j_i}\\ \mathbb{P}(s_{j_i}=-1\,|\,J)&=\Pi_{i-1}\,(1-p_{j_i})\\ \mathbb{P}(s_{j_i}=0\,|\,J)&=1-\mathbb{P}(s_{j_i}=1\,|\,J)-\mathbb{P}(s_{j_i}=-1\,|\,J)\ .\end{aligned}$$

Hence, for all $\hat\Delta_{j_i}\in\mathbb{R}$ we have

$$\begin{aligned}\mathbb{E}[L(s_{j_i}\hat\Delta_{j_i}) &- L(s_{j_i}\Delta_{j_i})\,|\,J]\\ &= \Pi_{i-1}\Big(p_{j_i}\Big(L(\hat\Delta_{j_i})-L(\Delta_{j_i})\Big)+(1-p_{j_i})\Big(L(-\hat\Delta_{j_i})-L(-\Delta_{j_i})\Big)\Big)\\ &\ge \Pi_{i-1}\Big(p_{j_i}\Big(L'(\Delta_{j_i})(\hat\Delta_{j_i}-\Delta_{j_i})+\frac{c_{\sigma'}}{2}(\hat\Delta_{j_i}-\Delta_{j_i})^2\Big)\\ &\qquad\qquad +(1-p_{j_i})\Big(L'(-\Delta_{j_i})(\Delta_{j_i}-\hat\Delta_{j_i})+\frac{c_{\sigma'}}{2}(\hat\Delta_{j_i}-\Delta_{j_i})^2\Big)\Big)\\ &\qquad \text{(using } L''(\Delta_{j_i})\ge c_{\sigma'}\text{)}\\ &= \Pi_{i-1}\frac{c_{\sigma'}}{2}(\hat\Delta_{j_i}-\Delta_{j_i})^2\\ &\qquad \text{(since } p_{j_i}=-L'(-\Delta_{j_i})\text{ and } 1-p_{j_i}=-L'(\Delta_{j_i})\ .\end{aligned}$$

Moreover,

$$\begin{aligned}\textsc{var}[L(s_{j_i}\hat\Delta_{j_i})-L(s_{j_i}\Delta_{j_i})\,|\,J] &\le \mathbb{E}[(L(s_{j_i}\hat\Delta_{j_i})-L(s_{j_i}\Delta_{j_i}))^2\,|\,J]\\ &\le \Pi_{i-1}(c_\sigma)^2(\hat\Delta_{j_i}-\Delta_{j_i})^2\\ &\qquad \text{(using } |L'(\Delta_{j_i})|\le c_\sigma\text{)}\ .\end{aligned}$$

Piecing together gives the claimed bound. □

The next lemma helps us define the upper confidence parameters $\epsilon_{j,t}$. To this effect, for $t \in [T]$, let $d_t(u,w)$ be the Mahalanobis distance between vectors u and w as

$$d_{c_t}(u,w) = (u-w)^\top M_{c_t}(u-w) \ ,$$

where M_{c_t} is the matrix maintained by Algorithm 2 at the c_t-th update. The next lemma follows from somewhat standard arguments, and relies on the exp-concavity of $L(\cdot)$.

Lemma 5. *Assume there exists $D > 0$ such that $u^\top x_j \in [-D, D]$ for all $x_j \in A$. Let c_σ and $c_{\sigma'}$ be two positive constants such that, for all $\Delta \in [-D, D]$ we have $0 < 1 - c_\sigma \le \sigma(\Delta) \le c_\sigma < 1$ and $\sigma'(\Delta) \ge c_{\sigma'}$. Then with probability at least $1-\delta$, with $\delta < 1/e$, we have*

$$\begin{aligned} & d_{c_t-1}(u, w'_{c_t}) \\ & \quad \le bD^2 + \left(\frac{c_\sigma}{c_{\sigma'}}\right)^2 d \log\left(1 + \frac{2}{b}\Big(\frac{t\, c_\sigma}{1-c_\sigma} + 4\log\frac{2(t+1)}{\delta}\Big)\right) + \left(12\left(\frac{c_\sigma}{c_{\sigma'}}\right)^2 + \frac{36(1+D)}{c_{\sigma'}}\right)\log\frac{2b(t+4)}{\delta} \end{aligned}$$

uniformly over $c_t \in [bT]$, where $b_t \le b$ for all $t \in [T]$.

Proof. Given items A, the update rules $w'_{c_t+j-1} \to w_{c_t+j} \to w'_{c_t+j}$ combined with the lower bound $L''(\Delta) \ge c_{\sigma'}$ allows us to write for all t (adapted from, e.g., Hazan et al. (2007); Gentile and Orabona (2012))

$$d_{c_t-1}(u, w'_{c_t}) \le bD^2 + \left(\frac{1}{c_{\sigma'}}\right)^2 \cdot \sum_{k=1}^{t-1}\sum_{j=1}^{\hat{s}_k} \nabla_{j,k}^\top M_{c_k+j-1}^{-1}\nabla_{j,k} - \frac{2}{c_{\sigma'}}\sum_{k=1}^{t-1}\sum_{j=1}^{\hat{s}_k}\Big(L(s_{j,k}x_j^\top w'_{c_k+j-1}) - L(s_{j,k}u^\top x_j)\Big) \ , \tag{14}$$

where $c_k = \hat{s}_1 + \hat{s}_2 + \ldots + \hat{s}_{k-1}$.

In particular, notice that the step $w_{c_t+j} \to w'_{c_t+j}$ is a projection of w_{c_t+j} onto the convex set $\{w \in \mathbb{R}^d \, : \, -D \le w^\top x_j \le D\}$ w.r.t. Mahalanobis distance $d_{c_t+j-1}(\cdot,\cdot)$. This projection can be computed in closed form as follows:

$$w'_{c_t+j-1} = \begin{cases} w_{c_t+j-1} & \text{if } |w_{c_t+j-1}^\top x_j| \le D \\ w_{c_t+j-1} - \frac{w_{c_t+j-1}^\top x_j - D}{x_j^\top M_{c_t+j-2}^{-1}x_j} M_{c_t+j-2}^{-1}x_j & \text{if } w_{c_t+j-1}^\top x_j > D \\ w_{c_t+j-1} - \frac{w_{c_t+j-1}^\top x_j + D}{x_j^\top M_{c_t+j-2}^{-1}x_j} M_{c_t+j-2}^{-1}x_j & \text{if } w_{c_t+j-1}^\top x_j < -D \ . \end{cases}$$

Further, we lower bound with high probability $\sum_{k=1}^{t-1}\sum_{j=1}^{\hat{s}_k}\Big(L(s_{j,k}x_j^\top w'_{c_k+j-1}) - L(s_{j,k}u^\top x_j)\Big)$ using the fact that the conditional expectation of the loss difference $L(s_{j,k}x_j^\top w'_{c_k+j-1}) - L(s_{j,k}u^\top x_j)$ is non-negative (Lemma 4).[12] The same lemma also allows for fast rates of convergence, so that we can apply any Freedman-like inequality (see, e.g., Lemma 3 in Kakade and Tewari (2008)) for bounded martingale difference sequences to conclude that

$$\sum_{k=1}^{t-1}\sum_{j=1}^{\hat{s}_k}\Big(L(s_{j,k}x_j^\top w'_{c_k+j-1}) - L(s_{j,k}u^\top x_j)\Big) \ge -\left(\frac{6(c_\sigma)^2}{c_{\sigma'}} + 18L(-D)\right)\log\frac{b(t+4)}{\delta}$$

with $b \ge b_t$ for all t, holds with probability $\ge 1 - \delta/(bt(t+1))$, the boundedness of the difference sequence following from the fact that $|u^\top x_j| \le D$ holds by assumption, and $|x_j^\top w'_{c_k+j-1}| \le D$ holds by the projection

[12] Here, Lemma 4 is applied with expectations conditioned on past history.

steps $w_{c_k+j-1} \to w'_{c_k+j-1}$. We then upper bound $L(-D)$ by $1+D$ and exploit a known upper bound:

$$\begin{aligned}
\sum_{k=1}^{t-1}\sum_{j=1}^{\hat{s}_k} \nabla_{j,k}^\top M_{c_k+j-1}^{-1} \nabla_{j,k} &= \sum_{k=1}^{t-1}\sum_{j=1}^{\hat{s}_k} \sigma^2(-s_{j,k} x_j^\top w'_{c_k+j-1}) |s_{j,k}| \left(x_j^\top M_{c_k+j-1}^{-1} x_j\right) \\
&\le (c_\sigma)^2 \sum_{k=1}^{t-1}\sum_{j=1}^{\hat{s}_k} |s_{j,k}| \left(x_j^\top M_{c_k+j-1}^{-1} x_j\right) \\
&\quad \text{(from the fact that } L'(\Delta) \le c_\sigma \text{ for all } \Delta \in [-D,D]\text{, and } |x_j^\top w'_{c_k+j-1}| \le D) \\
&\le (c_\sigma)^2 d \log\left(1 + \frac{1}{b}\sum_{k=1}^{t-1}\sum_{j=1}^{\hat{s}_k} |s_{j,k}|\right) \qquad (15) \\
&\quad \text{(from a standard inequality, e.g., Cesa-Bianchi et al. (2005))} .
\end{aligned}$$

Since $|s_{j,k}|$ is a Bernoulli random variable which is 1 (that is, the corresponding component of outcome vector Y_k is observed) with (conditional) probability $\Pi_{j-1,k} = \prod_{i=1}^{j-1}(1-\sigma(\Delta_{i,k}))$, where

$$\Delta_{i,k} = u^\top x_i \ , \qquad i = 1, \ldots, \widehat{s}_k \ ,$$

we can apply again the aforementioned Freedman-like inequality from Kakade and Tewari (2008) to conclude that

$$\mathbb{P}\left(\exists t \,:\, \sum_{k=1}^{t-1}\sum_{j=1}^{\hat{s}_k} |s_{j,k}| \le 2\sum_{k=1}^{t-1}\sum_{j=1}^{\hat{s}_k} \Pi_{j-1,k} + 4\log\frac{t(t+1)}{\delta}\right) \ge 1-\delta \ .$$

In turn, since $\Delta_{i,k} \in [-D,D]$, we have $1-\sigma(\Delta_{i,k}) \le c_\sigma$ for all i and k, so that $\sum_{j=1}^{\widehat{s}_t} \Pi_{j-1,k} \le \sum_{j=1}^{\infty} (c_\sigma)^j = \frac{c_\sigma}{1-c_\sigma}$. After some overapproximations, the above implies

$$\mathbb{P}\left(\exists t \,:\, \sum_{k=1}^{t-1}\sum_{j=1}^{\hat{s}_k} |s_{j,k}| \le 2(t-1)\,\frac{c_\sigma}{1-c_\sigma} + 8\log\frac{t+1}{\delta}\right) \ge 1-\delta \ .$$

We plug it back into (15), then back into (14) and replace δ by $\delta/2$ to obtain the claimed result. □

The next lemma takes care of the delayed feedback inherent in the cascading bandit learning protocol.

Lemma 6. *Let M be a $d \times d$ positive definite matrix whose minimal eigenvalue is $\ge b$, for some $b \in \{1,2,\ldots,\}$, and $x_1, x_2, \ldots, x_b \in \{x \in \mathbb{R}^d \,:\, ||x|| \le 1\}$. Then*

$$\sum_{j=1}^{b} x_j^\top M^{-1} x_j \le e \sum_{j=1}^{b} x_j^\top M_j^{-1} x_j \ ,$$

where $M_j = M + x_1 x_1^\top + \ldots x_j x_j^\top$, and e is the base of natural logarithms.

Proof. Consider the quantity $x^\top M_j^{-1} x$, with $M_0 = M$. We first prove that, for any $x \in \mathbb{R}^d$,

$$x^\top M^{-1} x \le \left(1 + \frac{1}{b}\right)^j x^\top M_j^{-1} x \qquad (16)$$

holds for all $j \in [b]$.

By the Sherman-Morrison formula for matrix inversion we have, for an arbitrary $x \in \mathbb{R}^d$, and $j \ge 1$,

$$\begin{aligned}
x^\top M_j^{-1} x &= x^\top (M_{j-1} + x_j x_j^\top)^{-1} x \\
&= x^\top M_{j-1}^{-1} x - \frac{(x^\top M_{j-1}^{-1} x_j)^2}{1 + x_j^\top M_{j-1}^{-1} x_j} \\
&\ge x^\top M_{j-1}^{-1} x - \frac{(x^\top M_{j-1}^{-1} x)(x_j^\top M_{j-1}^{-1} x_j)}{1 + x_j^\top M_{j-1}^{-1} x_j} \\
&\quad \text{(from the Cauchy-Schwarz inequality)}
\end{aligned}$$

so that

$$x^\top M_{j-1}^{-1} x \le x^\top M_j^{-1} x + \frac{(x^\top M_{j-1}^{-1} x)(x_j^\top M_{j-1}^{-1} x_j)}{1 + x_j^\top M_{j-1}^{-1} x_j} \ .$$

Hence, rearranging terms, we can write

$$x^\top M_{j-1}^{-1} x \le x^\top M_j^{-1} x (1 + x_j^\top M_{j-1}^{-1} x_j) \le x^\top M_j^{-1} x \left(1 + \frac{1}{b}\right),$$

the second inequality deriving from the assumption $||x_j|| \le 1$ and the fact that since the smallest eigenvalue of M is at least b, so is the smallest eigenvalue of $M_{j-1} \ge M$. Unwrapping this recurrence over j gives (16).

From (16), since $(1 + 1/b)^j \le e$ when $j \le b$, we have

$$x^\top M^{-1} x \le e \, x^\top M_j^{-1} x \ .$$

Since this holds for a generic x, we instantiate in turn x to $x_1, x_1, \ldots, x_b$, and sum over $j \in [b]$. This yields

$$\sum_{j=1}^{b} x_j^\top M^{-1} x_j \le e \sum_{j=1}^{b} x_j^\top M_j^{-1} x_j \ ,$$

as claimed. $\square$

Proof of Theorem 1. Consider matrix M_{c_t-1} in Lemma 5. If $J_r = \langle x_{\widehat{\jmath}_{r,1}}, \ldots, x_{\widehat{\jmath}_{r,\widehat{s}_r}} \rangle$, for $r = 1, \ldots, t-1$, we can write

$$M_{c_t-1} = bI + \sum_{r=1}^{t-1} \sum_{j=1}^{\widehat{s}_r} |s_{j,r}| \, x_{\widehat{\jmath}_r,j} x_{\widehat{\jmath}_r,j}^\top \ ,$$

where $|s_{j,r}|$ is a Bernoulli random variable which is 1 (that is, the corresponding component of outcome vector Y_r is observed) with probability $\Pi_{j-1,r} = \prod_{i=1}^{j-1}(1 - \sigma(\Delta_{i,r}))$, where

$$\Delta_{i,r} = u^\top x_{\widehat{\jmath}_r,i} \ , \qquad i = 1, \ldots, \widehat{s}_r \ .$$

Let

$$\bar{M}_{c_t-1} = bI + \sum_{r=1}^{t-1} \sum_{j=1}^{\widehat{s}_r} \Pi_{j-1,r} \, x_{\widehat{\jmath}_r,j} x_{\widehat{\jmath}_r,j}^\top \ ,$$

and consider the matrix martingale difference sequence

$$|s_{j,r}| \, x_{\widehat{\jmath}_r,j} x_{\widehat{\jmath}_r,j}^\top - \Pi_{j-1,r} \, x_{\widehat{\jmath}_r,j} x_{\widehat{\jmath}_r,j}^\top \ , \qquad r = 1, \ldots, t-1, \ j = 1, \ldots, \widehat{s}_r \ .$$

By a standard Freedman-style matrix martingale inequality (e.g., Tropp (2011)) adapted to our scenario we have, for positive constants θ and θ',

$$\mathbb{P}\left(\exists t \, : \, \lambda_{\max}\left(M_{c_t-1} - \bar{M}_{c_t-1}\right) \ge \theta, \ ||\bar{M}_{c_t-1}|| \le \theta'\right) \le d \, \exp\left(\frac{-\theta^2/2}{\theta' + \theta/3}\right) \ , \tag{17}$$

where $\lambda_{\max}(\cdot)$ denotes the algebraically largest eigenvalue of the matrix at argument, and $|| \cdot ||$ denotes the spectral norm.

We now proceed according to a standard stratification argument (e.g., Cesa-Bianchi and Gentile (2008)). Setting $A(x, \delta) = 2 \log \frac{xd}{\delta}$ and $f(A, r) = 2A + \sqrt{Ar}$, we can write

$$\begin{aligned}
&\mathbb{P}\left(\exists t \, : \, \lambda_{\max}\left(M_{c_t-1} - \bar{M}_{c_t-1}\right) \ge f(A(||\bar{M}_{c_t-1}||, \delta), ||\bar{M}_{c_t-1}||)\right) \\
&\le \sum_{r=0}^{\infty} \mathbb{P}\left(\exists t \, : \, \lambda_{\max}\left(M_{c_t-1} - \bar{M}_{c_t-1}\right) \ge f(A(||\bar{M}_{c_t-1}||, \delta), ||\bar{M}_{c_t-1}||), \quad 2^r - 1 \le ||\bar{M}_{c_t-1}|| \le 2^{r+1}\right) \\
&\le \sum_{r=0}^{\infty} \mathbb{P}\left(\exists t \, : \, \lambda_{\max}\left(M_{c_t-1} - \bar{M}_{c_t-1}\right) \ge f(A(2^{r+1}, \delta), 2^{r+1}), \quad ||\bar{M}_{c_t-1}|| \le 2^{r+1}\right) \\
&\le \sum_{r=0}^{\infty} d \, \exp\left(\frac{-f^2(A(2^{r+1}, \delta), 2^{r+1})/2}{2^{r+1} + f(A(2^{r+1}, \delta), 2^{r+1})/3}\right) \ ,
\end{aligned}$$

the last inequality deriving from (17).

Since $f(A,r)$ satisfies $f^2(A,r) \geq Ar + A + 2/3f(A,r)A$, the exponent in the last exponential is at least $A(2^{r+1},\delta)/2$, implying

$$\sum_{r=0}^{\infty} \exp\left(-A(2^{r+1},\delta)/2\right) = \sum_{r=0}^{\infty} \frac{\delta}{d\,2^{r+1}} = \delta/d \ ,$$

which in turn implies

$$\mathbb{P}\left(\exists t \,:\, \lambda_{\max}\left(M_{c_t-1} - \bar{M}_{c_t-1}\right) \geq f(A(||\bar{M}_{c_t-1}||,\delta), ||\bar{M}_{c_t-1}||)\right) \leq \delta \ .$$

Plugging back the definitions of $f(A,r)$ and $A(x,\delta)$, noticing that $||\bar{M}_{c_t-1}|| = \lambda_{\max}(\bar{M}_{c_t-1}) \leq b(t+1)$ (due to the fact that $||\bar{M}_{c_t-1}||$ is positive definite and $||x_{\widehat{j}_t,j}|| \leq 1$), and overapproximating gives

$$\mathbb{P}\left(\exists t \,:\, \lambda_{\max}\left(M_{c_t-1} - \bar{M}_{c_t-1}\right) \geq 4\log\frac{bd(t+1)}{\delta} + \sqrt{2\,\lambda_{\max}(\bar{M}_{c_t-1})\,\log\frac{bd(t+1)}{\delta}}\right) \leq \delta \ .$$

Further, we use $\sqrt{ab} \leq a/2 + b/2$ with $a = \lambda_{\max}(\bar{M}_{c_t-1})$ and $b = 2\log\frac{bd(t+1)}{\delta}$. Rearranging gives

$$\mathbb{P}\left(\exists t \,:\, \frac{1}{2}\,\lambda_{\max}\left(\bar{M}_{c_t-1}\right) - \lambda_{\max}(\bar{M}_{c_t-1} - M_{c_t-1}) \leq -5\,\log\frac{bd(t+1)}{\delta}\right) \leq \delta$$

or, equivalently,

$$\mathbb{P}\left(\forall t\ \frac{1}{2}\,\lambda_{\max}\left(\bar{M}_{c_t-1}\right) - \lambda_{\max}(\bar{M}_{c_t-1} - M_{c_t-1}) \geq -5\,\log\frac{bd(t+1)}{\delta}\right) \geq 1-\delta \ .$$

Now, observing that

$$\lambda_{\max}\left(\bar{M}_{c_t-1}\right) - \lambda_{\max}(2\bar{M}_{c_t-1} - 2M_{c_t-1}) \leq \lambda_{\max}\left(2M_{c_t-1} - \bar{M}_{c_t-1}\right)$$

the above implies

$$\mathbb{P}\left(\forall t\ \lambda_{\max}\left(M_{c_t-1} - \frac{1}{2}\bar{M}_{c_t-1}\right) \geq -5\,\log\frac{bd(t+1)}{\delta}\right) \geq 1-\delta \ ,$$

which can be rewritten as

$$\mathbb{P}\left(\forall t\ \forall v \in \mathbb{R}^d \,:\, \frac{v^\top\left(M_{c_t-1} - \frac{1}{2}\bar{M}_{c_t-1}\right)v}{v^\top v} \geq -5\,\log\frac{bd(t+1)}{\delta}\right) \geq 1-\delta \ .$$

If we define

$$\bar{d}_{c_t-1}(u,w) = (u-w)^\top \bar{M}_{c_t-1}(u-w)$$

the above inequality allows us to conclude that

$$d_{c_t-1}(u,w) \geq \frac{1}{2}\,\bar{d}_{c_t-1}(u,w) - 20D^2\log\frac{bd(t+1)}{\delta}$$

holds with probability at least $1-\delta$, uniformly over all $u,w \in \mathbb{R}^d$ such that $||u-w|| \leq 2D$ and all rounds t. Hence, combining with Lemma 5, and upper bounding t by T,

$$\bar{d}_{c_t-1}(u,w'_{c_t}) \leq \alpha(b,d,T,2\delta,D)$$

where

$$\alpha(b,d,T,2\delta,D) = 2bD^2 + \left(\frac{c_\sigma}{c_{\sigma'}}\right)^2 d\log\left(1 + \frac{2}{b}\Big(\frac{T\,c_\sigma}{1-c_\sigma} + 4\log\frac{2(T+1)}{\delta}\Big)\right)$$
$$+ 2\left(12\left(\frac{c_\sigma}{c_{\sigma'}}\right)^2 + \frac{36(1+D)}{c_{\sigma'}}\right)\log\frac{b(T+4)}{\delta} + 20D^2\log\frac{bd(T+1)}{\delta}$$

with probability at least $1 - 2\delta$.

Then Cauchy-Schwarz inequality allows us to write, for all $x \in \mathbb{R}^d$,

$$(u^\top x - x^\top w'_{c_t})^2 \le x^\top \bar{M}_{c_t-1}^{-1} x \, \bar{d}_{c_t-1}(u, w'_{c_t}) \le \left(x^\top \bar{M}_{c_t-1}^{-1} x\right) \alpha(b, d, T, 2\delta) \ .$$

We are therefore in a position to apply Lemma 3 with J therein set to $J_t = \langle x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,\widehat{s}_t}} \rangle$ and ϵ_j set to $\epsilon_{\widehat{j}_t,j} = \sqrt{\left(x_{\widehat{j}_t,j}^\top \bar{M}_{c_t-1}^{-1} x_{\widehat{j}_t,j}\right) \alpha(b, d, T, 2\delta, D)}$, for $j = 1, \ldots, \widehat{s}_t$. Thus we can write

$$\sum_{t=1}^{T} \mathbb{E}_{Y_t}[R(J_t^*, Y_t)] - \mathbb{E}_{Y_t}[R(J_t, Y_t)] \le 4z\sqrt{\alpha(b, d, T, 2\delta, D)} \sum_{t=1}^{T} \sum_{j=1}^{\widehat{s}_t} \sqrt{\left(x_{\widehat{j}_t,j}^\top \bar{M}_{c_t-1}^{-1} x_{\widehat{j}_t,j}\right)} \, \Pi_{j-1,t} \ . \tag{18}$$

Now, for each round t, consider the quantity

$$\sum_{j=1}^{\widehat{s}_t} \left(x_{\widehat{j}_t,j}^\top \bar{M}_{c_t-1}^{-1} x_{\widehat{j}_t,j}\right) \Pi_{j-1,t}$$

Noticing that $\bar{M}_0 = bI$, we invoke Lemma 6 with x_j therein set to $x_{\widehat{j}_t,j}\sqrt{\Pi_{j-1,t}}$ and write

$$\sum_{j=1}^{\widehat{s}_t} \left(x_{\widehat{j}_t,j}^\top \bar{M}_{c_t-1}^{-1} x_{\widehat{j}_t,j}\right) \Pi_{j-1,t} \le e \sum_{j=1}^{\widehat{s}_t} \left(x_{\widehat{j}_t,j}^\top \bar{M}_{c_t-1+j}^{-1} x_{\widehat{j}_t,j}\right) \Pi_{j-1,t} \tag{19}$$

where

$$\bar{M}_{c_t-1+j} = \bar{M}_{c_t-1} + \sum_{i=1}^{j} x_{\widehat{j}_t,i} x_{\widehat{j}_t,i}^\top \Pi_{i-1,t} \ ,$$

with $\Pi_{0,t} = 1$. Thus, for each t,

$$\begin{aligned}
\sum_{j=1}^{\widehat{s}_t} \sqrt{\left(x_{\widehat{j}_t,j}^\top \bar{M}_{c_t-1}^{-1} x_{\widehat{j}_t,j}\right)} \, \Pi_{j-1,t} &= \sum_{j=1}^{\widehat{s}_t} \sqrt{\left(x_{\widehat{j}_t,j}^\top \bar{M}_{c_t-1}^{-1} x_{\widehat{j}_t,j}\right) \Pi_{j-1,t}} \sqrt{\Pi_{j-1,t}} \\
&\le \sqrt{\sum_{j=1}^{\widehat{s}_t} \left(x_{\widehat{j}_t,j}^\top \bar{M}_{c_t-1}^{-1} x_{\widehat{j}_t,j}\right) \Pi_{j-1,t}} \sqrt{\sum_{j=1}^{\widehat{s}_t} \Pi_{j-1,t}} \\
&\quad \text{(from the Cauchy-Schwarz inequality)} \\
&\le \sqrt{\frac{e\, c_\sigma}{1 - c_\sigma} \sum_{j=1}^{\widehat{s}_t} \left(x_{\widehat{j}_t,j}^\top \bar{M}_{c_t-1+j}^{-1} x_{\widehat{j}_t,j}\right) \Pi_{j-1,t}} \\
&\quad \text{(from (19), along with } \Pi_{j-1,t} \le 1 \text{ and } \textstyle\sum_{j=1}^{\widehat{s}_t} \Pi_{j-1,t} \le \frac{c_\sigma}{1-c_\sigma}, \\
&\quad \text{as argued within the proof of Lemma 5)} \ .
\end{aligned}$$

Getting back to (18), combining with the last inequality we have

$$
\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}_{Y_t}[R(J_t^*, Y_t)] - \mathbb{E}_{Y_t}[R(J_t, Y_t)] \\
&\qquad \le 4z\sqrt{\alpha(b,d,T,2\delta,D)} \sum_{t=1}^{T} \sqrt{\frac{e\,c_\sigma}{1-c_\sigma} \sum_{j=1}^{\widehat{s}_t} \left(x_{\widehat{j}_t,j}^\top \bar{M}_{c_t-1+j}^{-1} x_{\widehat{j}_t,j}\right) \Pi_{j-1,t}} \\
&\qquad \le 4z\sqrt{\frac{e\,c_\sigma}{1-c_\sigma}\alpha(b,d,T,\delta,D)T \sum_{t=1}^{T}\sum_{j=1}^{\widehat{s}_t} \left(x_{\widehat{j}_t,j}^\top \bar{M}_{c_t-1+j}^{-1} x_{\widehat{j}_t,j}\right) \Pi_{j-1,t}}\ , \\
&\qquad \text{(again from the Cauchy-Schwarz inequality)} \\
&\qquad \le 4z\sqrt{\frac{e\,c_\sigma}{1-c_\sigma}\alpha(b,d,T,\delta,D)T\, d\, \log\left(1+\frac{1}{b}\sum_{t=1}^{T}\sum_{j=1}^{\widehat{s}_t}\Pi_{j-1,t}\right)} \\
&\qquad \text{(from a standard inequality, e.g., (Cesa-Bianchi et al., 2005),} \\
&\qquad \text{along with } ||x_{\widehat{j}_t,j}|| \le 1 \text{ and } M_0 = bI) \\
&\qquad \le 4z\sqrt{\frac{e\,c_\sigma}{1-c_\sigma}\alpha(b,d,T,\delta,D)T\, d\, \log(1+T)} \\
&\qquad \text{(since } \Pi_{j-1,t} \le 1 \text{ and } \widehat{s}_t \le b)\ .
\end{aligned}
$$

Since the above holds with probability $\ge 1-2\delta$, we replace δ by $\delta/2$ in $\alpha(b,d,T,2\delta,D)$. Then we consider that, since $\sigma(x) = \frac{\exp(x)}{1+\exp(x)}$, we have $c_\sigma = \frac{e^D}{1+e^D}$ (so that $\frac{c_\sigma}{1-c_\sigma} = e^D$), $c_{\sigma'} = e^{-D}/(1+e^{-D})^2 \ge e^{-D}/4$, and $z=1$. Plugging back gives the claimed result.

□

B ALGORITHM FOR THE CASE OF DEPENDENT OUTCOMES

For completeness, we give in Algorithm 3 (see end of the paper) the pseudocode of the greedy algorithm arising from the dependent model of outcomes. All in all, the algorithm performs the same updates as Algorithm 2, but applied to the coverage difference vectors $\bar{c}(x_{j_k} \mid x_{j_1},\ldots,x_{j_{k-1}})$ instead of the original feature vectors x_{j_k}. Moreover, Algorithm 3 replaces the computation of J_t by mimicking the greedy algorithm described in Section 4. In the experiments of Section 5, we replaced Algorithm 3 with a simplified version that removes the projection step and introduces the two parameters α and η as in Algorithm 1.

In the pseudocode of Algorithm 3 we define

$$
\begin{aligned}
\alpha(b,d',T,\delta,D) = 2bD^2 &+ \left(\frac{c_\sigma}{c_{\sigma'}}\right)^2 d' \log\left(1+\frac{2}{b}\Big(\frac{T\,c_\sigma}{1-c_\sigma}+4\log\frac{4(T+1)}{\delta}\Big)\right) \\
&+ 2\left(12\left(\frac{c_\sigma}{c_{\sigma'}}\right)^2 + \frac{36(1+D)}{c_{\sigma'}}\right)\log\frac{2b(T+4)}{\delta} + 20D^2\log\frac{2bd'(T+1)}{\delta}\ .
\end{aligned}
$$

Below we give the derivation for the approximation ratio claimed in the main body of the paper

Lemma 7. *Fix $s \in \{0,1,\ldots,b\}$. Let $J^* = \langle x_{j_1^*},\ldots,x_{j_s^*}\rangle$ be the Bayes optimal sequence under model (5) with unknown vector u. Let $(x_{j_1'}, x_{j_2'},\ldots)$ be the order of items according to Eq.(7) and the unknown vector u and $J' = \langle x_{j_1'},\ldots,x_{j_s'}\rangle$ be the sequence taking first s elements. Suppose $\bar{c}(x_k \mid x_1,\ldots,x_{k-1})^\top u \in [-D,D]$ for all $x_1,\ldots,x_k \in A$. Assume all components of u are non-negative and that*[13] *$\|u\|_1 \le \frac{\sqrt{d'}(z-(1-1/e)c_{\sigma'})}{6(z^2-(1-1/e)c_{\sigma'}^2)}$. Moreover,*

[13]Notice that, since $z=1$ and $c_{\sigma'} = e^{-D}/(1+e^{-D})^2$, this requirement is essentially equivalent to something like $||u||_1 = O(\sqrt{d'})$.

let the reward and loss sequences satisfy[14]

$$s(r_s-\ell_s)\max\left\{\frac{1}{s},1-\frac{s-1}{2}c_\sigma\right\}\left(1-\left(1-\frac{1}{e}\right)\frac{c_{\sigma'}}{z}\right)$$
$$+\,3\ \ell_s\left(1-\max\left\{\frac{1}{s},1-\frac{s-1}{2}c_\sigma\right\}\left(1-\frac{1}{e}\right)\frac{c_{\sigma'}(r_s-\ell_s)}{z(r_1-\ell_s)}\right)\geq 0\,.$$

Let

$$\gamma(s)=\begin{cases}\max\{\frac{1}{s},1-\frac{s-1}{2}c_\sigma\}(1-\frac{1}{e})\frac{c_{\sigma'}(r_s-\ell_s)}{z(r_1-\ell_s)} & s\geq 2\,,\\ 1 & s=0,1\ .\end{cases}$$

Then

$$\mathbb{E}_Y[R(J',Y)]\geq\gamma(s)\ \mathbb{E}_Y[R(J^*,Y)]\,.$$

Proof. It is immediate to see the conclusion holds for $s=0,1$. Now assume $s\geq 2$. Let $J=\langle x_{j_1},\ldots,x_{j_s}\rangle$ be any sequence of length s. Then, setting for brevity $a=2/\sqrt{d'}$ and $a'=-1/\sqrt{d'}(1,\ldots,1)^\top$, we can write

$$\begin{aligned}
\mathbb{E}_Y[R(J,Y)] &= E(\Delta_{j_1},\Delta_{j_1,j_2},\ldots,\Delta_{j_1,j_2,\ldots,j_s})\\
&= r_1p(\Delta_{j_1})+r_2p(\Delta_{j_1,j_2})(1-p(\Delta_{j_1}))+\cdots+r_sp(\Delta_{j_1,\ldots,j_s})\prod_{i=1}^{s-1}(1-p(\Delta_{j_1,\ldots,j_i}))\\
&\qquad +\ell_s(1-\prod_{i=1}^{s}(1-p(\Delta_{j_1,\ldots,j_i})))\\
&= (r_1-\ell_s)p(\Delta_{j_1})+(r_2-\ell_s)p(\Delta_{j_1,j_2})(1-p(\Delta_{j_1}))+\cdots\\
&\qquad +(r_s-\ell_s)p(\Delta_{j_1,\ldots,j_s})\prod_{i=1}^{s-1}(1-p(\Delta_{j_1,\ldots,j_i}))+\ell_s\\
&\leq (r_1-\ell_s)(1-\prod_{i=1}^{s}(1-p(\Delta_{j_1,\ldots,j_i})))+\ell_s\\
&\leq (r_1-\ell_s)\sum_{i=1}^{s}p(\Delta_{j_1,\ldots,j_i})+\ell_s\\
&= (r_1-\ell_s)\sum_{i=1}^{s}\sigma(a\cdot c'(x_{j_i}\mid x_{j_1},\ldots,x_{j_{i-1}})^\top u+a'^\top u)+\ell_s\\
&\leq (r_1-\ell_s)\sum_{i=1}^{s}\left(\sigma(a\cdot c'(x_{j_i}\mid x_{j_1},\ldots,x_{j_{i-1}})^\top u)+c_{\sigma'}a'^\top u\right)+\ell_s\\
&\leq (r_1-\ell_s)\sum_{i=1}^{s}\left(\sigma(0)+z\cdot a\cdot c'(x_{j_i}\mid x_{j_1},\ldots,x_{j_{i-1}})^\top u+c_{\sigma'}a'^\top u\right)+\ell_s\\
&= (r_1-\ell_s)(s/2+s\ c_{\sigma'}\ a'^\top u+z\cdot a\cdot\langle c'(\{x_{j_1},\ldots,x_{j_s}\}),u\rangle)+\ell_s\\
&= (r_1-\ell_s)(s/2+s\ c_{\sigma'}\ a'^\top u+z\cdot a\cdot\langle c'(J),u\rangle)+\ell_s\,,
\end{aligned}$$

where the fourth and third lines from last are both from the properties of the σ function. Also

$$\begin{aligned}
\mathbb{E}_Y[R(J,Y)] &\geq (r_s-\ell_s)(1-\prod_{i=1}^{s}(1-p(\Delta_{j_1,\ldots,j_i})))+\ell_s\\
&\geq (r_s-\ell_s)\max\left\{\frac{1}{s},1-\frac{s-1}{2}c_\sigma\right\}\sum_{i=1}^{s}p(\Delta_{j_1,\ldots,j_i})+\ell_s
\end{aligned}$$

[14]For example, this requirement holds when $r_s\geq 5|\ell_s|$ for all $s\geq 1$, and $\frac{c_{\sigma'}}{z}\leq\frac{1}{2(1-1/e)}$.

$$
\begin{aligned}
&= (r_s - \ell_s) \max\Big\{\frac{1}{s}, 1 - \frac{s-1}{2} c_\sigma\Big\} \sum_{i=1}^{s} \sigma(a \cdot c'(x_{j_i} \mid x_{j_1}, \ldots, x_{j_{i-1}})^\top u + a'^\top u) + \ell_s \\
&\geq (r_s - \ell_s) \max\Big\{\frac{1}{s}, 1 - \frac{s-1}{2} c_\sigma\Big\} \sum_{i=1}^{s} \big(\sigma(a \cdot c'(x_{j_i} \mid x_{j_1}, \ldots, x_{j_{i-1}})^\top u) + z \cdot a'^\top u\big) + \ell_s \\
&\geq (r_s - \ell_s) \max\Big\{\frac{1}{s}, 1 - \frac{s-1}{2} c_\sigma\Big\} \sum_{i=1}^{s} \big(\sigma(0) + c_{\sigma'}\ a \cdot c'(x_{j_i} \mid x_{j_1}, \ldots, x_{j_{i-1}})^\top u + z \cdot a'^\top u\big) + \ell_s \\
&= (r_s - \ell_s) \max\Big\{\frac{1}{s}, 1 - \frac{s-1}{2} c_\sigma\Big\} (s/2 + s\ z\ a'^\top u + c_{\sigma'}\ a\ \langle c'(\{x_{j_1}, \ldots, x_{j_s}\}), u\rangle) + \ell_s \\
&= (r_s - \ell_s) \max\Big\{\frac{1}{s}, 1 - \frac{s-1}{2} c_\sigma\Big\} (s/2 + s\ z\ a'^\top u + c_{\sigma'}\ a\ \langle c'(J), u\rangle) + \ell_s\,,
\end{aligned}
$$

where the second inequality is by Lemma 1 of Hiranandani et al. (2020), and the fourth and fifth lines are from the properties of the σ function. Thus

$$
\begin{aligned}
\mathbb{E}_Y[R(J', Y)] &\geq (r_s - \ell_s) \max\Big\{\frac{1}{s}, 1 - \frac{s-1}{2} c_\sigma\Big\} \big(s/2 + s\ z\ a'^\top u + a\ c_{\sigma'} \langle c'(J'), u\rangle\big) + \ell_s \\
&\geq (r_s - \ell_s) \max\Big\{\frac{1}{s}, 1 - \frac{s-1}{2} c_\sigma\Big\} \left(s/2 + s\ z\ a'^\top u + a\ c_{\sigma'} \Big(1 - \frac{1}{e}\Big) \max_J \langle c'(J), u\rangle\right) + \ell_s \\
&\geq (r_s - \ell_s) \max\Big\{\frac{1}{s}, 1 - \frac{s-1}{2} c_\sigma\Big\} \left(s/2 + s\ z\ a'^\top u + a\ c_{\sigma'} \Big(1 - \frac{1}{e}\Big) \langle c'(J^*), u\rangle\right) + \ell_s \\
&\geq (r_s - \ell_s) \max\Big\{\frac{1}{s}, 1 - \frac{s-1}{2} c_\sigma\Big\} \\
&\quad \times \left(s/2 + s\ z\ a'^\top u + a\ c_{\sigma'} \Big(1 - \frac{1}{e}\Big) \left(\frac{\mathbb{E}_Y[R(J^*, Y)] - \ell_s}{a\ z\ (r_1 - \ell_s)} - \frac{s}{2az} - \frac{s\ c_{\sigma'}\ a'^\top u}{az}\right)\right) + \ell_s \\
&\geq \max\Big\{\frac{1}{s}, 1 - \frac{s-1}{2} c_\sigma\Big\} \Big(1 - \frac{1}{e}\Big) \frac{c_{\sigma'} (r_s - \ell_s)}{z (r_1 - \ell_s)} \mathbb{E}_Y[R(J^*, Y)] \\
&\quad + (r_s - \ell_s) \max\Big\{\frac{1}{s}, 1 - \frac{s-1}{2} c_\sigma\Big\} s \left(\frac{1}{2}(1 - \Big(1 - \frac{1}{e}\Big) \frac{c_{\sigma'}}{z}) + a'^\top u (z - \Big(1 - \frac{1}{e}\Big) \frac{c_{\sigma'}^2}{z})\right) \\
&\quad + \ell_s \left(1 - \max\Big\{\frac{1}{s}, 1 - \frac{s-1}{2} c_\sigma\Big\} \Big(1 - \frac{1}{e}\Big) \frac{c_{\sigma'} (r_s - \ell_s)}{z (r_1 - \ell_s)}\right) \\
&\geq \max\Big\{\frac{1}{s}, 1 - \frac{s-1}{2} c_\sigma\Big\} \Big(1 - \frac{1}{e}\Big) \frac{c_{\sigma'} (r_s - \ell_s)}{z (r_1 - \ell_s)} \mathbb{E}_Y[R(J^*, Y)]\ .
\end{aligned}
$$

In the above, the second inequality is based on the fact that the selection of J' is equivalent to running GREEDY on maximizing $\langle c(J), u\rangle$ over J, along with the typical approximation ratio of monotone and sub-modular set function optimization. The third inequality is by $\max_J \langle c'(J), u\rangle \geq \langle c'(J^*), u\rangle$. The fourth inequality is by the lower bound of $\mathbb{E}_Y[R(J^*, Y)]$ in terms of $\langle c'(J^*), u\rangle$. The last inequality is by the definition of a', and the assumptions on r_s, ℓ_s. ☐

The next lemma is the dependent outcome counterpart to Lemma 3.

Lemma 8. *Let us assume the dependent model (5) for outcome vector Y. Then, for given set of actions A, and budget b, let J^* be the Bayes optimal sequence and $J = \langle x_{j_1}, \ldots, x_{j_s}\rangle$ be the sequence computed by Algorithm 3 on A and b, with link function σ such that $\sigma'(\Delta) \leq z$ for all $\Delta \in \mathbb{R}$. Further, let $\Delta_{j_1,\ldots,j_k} = u^\top \bar{c}(x_{j_k} \mid x_{j_1}, \ldots, x_{j_{k-1}})$, and $\hat{\Delta}_{j_1,\ldots,j_k} = w^\top \bar{c}(x_{j_k} \mid x_{j_1}, \ldots, x_{j_{k-1}})$, for all conditional vectors computed from A, and assume $|\Delta_{j_1,\ldots,j_k} - \hat{\Delta}_{j_1,\ldots,j_k}| \leq \epsilon_{j_1,\ldots,j_k}$ for all j sequence, where w is the vector used by Algorithm 3 to compute J. Suppose[15] $\Delta_{j_1,\ldots,k_k} + 2\epsilon_{j_1,\ldots,j_k} \in [-D, D]$ for all $x_{j_1}, \ldots, x_{j_k} \in A$. Then the scaled one-time regret (8) can be bounded as follows:*

$$
\mathbb{E}_Y[\gamma(s_t^*) R(J^*, Y)] - \mathbb{E}_Y[R(J, Y)] \leq \begin{cases} 4z \sum_{i=1}^{s} \epsilon_{j_1,\ldots,j_i} \prod_{h=1}^{i-1} (1 - \sigma(\Delta_{j_1,\ldots,j_h})) & \text{if } J \neq \langle\rangle \\ 0 & \text{otherwise}\ . \end{cases}
$$

[15]This requirement is controllable since $\epsilon_{j_1,\ldots,j_k}$ is reasonably small after $O(\log T)$ rounds.

Proof. Irrespective of whether $J \neq \langle\rangle$ or $J^* \neq \langle\rangle$, we can write

$$\begin{aligned}
&\mathbb{E}_Y[\gamma(s_t^*)R(J^*,Y)] - \mathbb{E}_Y[R(J,Y)] \\
&\quad \le \widehat{\mathbb{E}}_Y[\gamma(s_t^*)R(J^*,Y)] - \mathbb{E}_Y[R(J,Y)] \\
&\quad \le \widehat{\mathbb{E}}_Y[R(J',Y)] - \mathbb{E}_Y[R(J,Y)] \\
&\quad \le \widehat{\mathbb{E}}_Y[R(J,Y)] - \mathbb{E}_Y[R(J,Y)] \\
&\quad = E(\hat{\Delta}_{j_1} + \epsilon_{j_1}, \hat{\Delta}_{j_1,j_2} + \epsilon_{j_1,j_2}, \ldots, \hat{\Delta}_{j_1,j_2,\ldots,j_s} + \epsilon_{j_1,j_2,\ldots,j_s}) - E(\Delta_{j_1}, \Delta_{j_1,j_2} \ldots, \Delta_{j_1,j_2,\ldots,j_s}),
\end{aligned}$$

where $\widehat{\mathbb{E}}_Y$ is defined as in Algorithm 2 by using $\hat{\Delta}_{j_1,\ldots,k_k} + \epsilon_{j_1,\ldots,j_k}$. Here J' is computed similarly in Lemma 7 but under $\hat{\Delta}_{j_1,\ldots,k_k} + \epsilon_{j_1,\ldots,j_k}$ and length s_t^*. The $\gamma(s_t^*)$-approximation still holds according to Lemma 7. The list J' is just $\hat{J}_{s_t^*}$ in Algorithm 3 and is no better than J under $\widehat{\mathbb{E}}_Y$ according to the computation of s.

Similar to the proof of Lemma 3, by the mean-value theorem, we can write

$$\begin{aligned}
E(\Delta_{j_1} + 2\epsilon_{j_1}, &\Delta_{j_1,j_2} + 2\epsilon_{j_1,j_2}, \ldots, \Delta_{j_1,j_2,\ldots,j_s} + 2\epsilon_{j_1,j_2,\ldots,j_s}) - E(\Delta_{j_1}, \Delta_{j_1,j_2} \ldots, \Delta_{j_1,j_2,\ldots,j_s}) \\
&= 2\sum_{i=1}^{s} \frac{\partial E(\Delta_{j_1}, \Delta_{j_1,j_2} \ldots, \Delta_{j_1,j_2,\ldots,j_s})}{\partial \Delta_{j_1,j_2,\ldots,j_i}}\Big|_{\Delta_{j_1}=\xi_{j_1},\ldots,\Delta_{j_1,\ldots,j_s}=\xi_{j_1,\ldots,j_s}} \epsilon_{j_1,\ldots,j_i},
\end{aligned}$$

where $\xi_{j_1,\ldots,j_i} \in (\Delta_{j_1,\ldots,j_i}, \Delta_{j_1,\ldots,j_i} + 2\epsilon_{j_1,\ldots,j_i})$. The third part of Lemma 2 then allows us to write

$$\begin{aligned}
\frac{\partial E(\Delta_{j_1}, \Delta_{j_1,j_2} \ldots, \Delta_{j_1,j_2,\ldots,j_s})}{\partial \Delta_{j_1,j_2,\ldots,j_i}}&\Big|_{\Delta_{j_1}=\xi_{j_1},\ldots,\Delta_{j_1,\ldots,j_s}=\xi_{j_1,\ldots,j_s}} \\
&\le 2z\,(1-\sigma(\xi_{j_1}))\cdots(1-\sigma(\xi_{j_1,\ldots,j_{i-1}})) \\
&\le 2z\,(1-\sigma(\Delta_{j_1}))\cdots(1-\sigma(\Delta_{j_1,\ldots,j_{i-1}})),
\end{aligned}$$

the second inequality deriving from the monotonicity of $\sigma(\cdot)$ and the fact that $\xi_{j_1,\ldots,j_i} \in (\Delta_{j_1,\ldots,j_i}, \Delta_{j_1,\ldots,j_i} + 2\epsilon_{j_1,\ldots,j_i})$. Replacing back, and summing over i yields the claimed bound. □

Based on this lemma, we combine with the corresponding remaining parts in the proof for the independent case. This gives us a *scaled* regret bound which coincides with the one for the dependent case.

Yet, it is worth stressing that, despite the two regret *bounds* look alike, the two underlying notions of regret are widely different, both because we have now a scaled regret, and because of the different assumptions on the process generating the outcomes as compared to the independent case.

C FURTHER RELATED WORK

Kveton et al. (2015b) studies a variant of cascading bandits where the feedback stops when a 0 outcome is observed, as opposed to a 1 outcome of the standard cascading bandit model. This reward is equivalent to a Boolean AND function on the sequence, and the available sequences are defined by combinatorial constraints of the problem. Zhou et al. (2018) also studies a variant of cascading bandits where each arm has an extra (unknown) cost when displayed. The length of the recommended sequences can also change, but in their setting this is due to the trade-off between the attractiveness and the cost of an item, while in our setting this is due to the trade-off between attractiveness of items and both reward and loss values. The combinatorial semi-bandit setting with probabilistically triggered arms ZZ et al. (2018) is a generalization of the cascading bandit setting that also encompasses, for instance, influence maximization problems. The authors are able to remove the inconvenient dependence on $1/p^*$ alluded to at the end of Section 3, but their comprehensive analysis only applies to non-contextual bandit scenarios.

Besides cascading bandits, relevant works investigate bandits with submodular reward functions to account for diversity in the item assortment (e.g., Yue and Guestrin (2011); Takemori et al. (2020)). In particular, Takemori et al. (2020) show a regret bound of the form $\sqrt{bT}$ in a submodular bandits scenario with rewards on items similar to our setting, yet relying on a feedback which is more informative than ours. For instance, in the independent case, their setting is equivalent to a (constrained) combinatorial bandits scenario with *semi-bandit* feedback with linear rewards.

Regarding the generative model for outcome vectors, following previous work Li and Zhang (2018), we assumed the probability that an item is successful is ruled by a generalized linear model (GLM), Such a model is more convenient than a purely linear model, since the sigmoidal link function would always map values to $(0, 1)$ which we need here to encode probabilities and compute the Bayes optimal sequence. The bandit problem under GLM assumptions is first studied in Filippi et al. (2010), whose regret bound can be improved by the finer self-concordant analysis of Faury et al. (2020). The online Newton step analysis presented here is inspired by the GLM-based bandit analysis contained in Gentile and Orabona (2012). See also Zhang et al. (2016) for similar results. Li et al. (2017) gives an optimal solution for this model up to a constant coefficient.

Finally, the update method that deals with long sequences in our paper also often appears in the study of bandit algorithms with delayed feedback. There is indeed some kind of similarity between a cascading model and a delayed feedback model in bandits: both share the need for a bandit algorithm to deal with signals that are received somehow later than the time the algorithm commit to actions. Relevant works in bandits with delayed feedback include Dudik et al. (2011); Joulani et al. (2013); Cesa-Bianchi et al. (2019); Pike-Burke et al. (2018); Zhou et al. (2019); Arya and Yang (2020). Yet, we are not aware of a way to reduce the delayed bandit model to the cascading bandit model, or vice versa.

D FURTHER EXPERIMENTAL RESULTS

This section contains details on our experimental setting and further results that have been omitted from the main paper.

D.1 Dataset Preprocessing

We report here the pre-processing steps we followed for the Million Songs, Yelp, and MNIST datasets.

- **Million Songs:** The Million Songs Dataset (MSD) is a repository of audio features and metadata of a million contemporary pop songs. We consider the Echo Nest Taste Profile Subset of MSD that contains the play-counts of some of these songs by real users. We pick $100,000$ users that have played the highest number of songs and $50,000$ songs with the highest number of users. We sample $10,000$ songs at random and calculate the singular value decomposition (SVD) of the corresponding $100,000 \times 10,000$ ratings matrix into 10 principal components. The projection matrices from the SVD are used to compute embeddings of dimension $d = 10$ for the remaining $40,000$ songs for training the bandit algorithms. The embeddings are normalized to unit L_2-norm and the dataset is shuffled randomly. In every round of bandit learning, the algorithm is presented with a non-overlapping chunk of movies as arms (A_t). The chunk size is 100. The outcome of an arm is decided by the mean rating received by the corresponding movie in the dataset. If this mean rating is greater than its median value in the dataset, the outcome is a success, else is a failure. As mentioned in Section 2.2, for the dependent algorithm the 40,000 SVD-projected d-dimensional vectors have been used to compute coverage vectors through a Gaussian Mixture Model (GMM) with d' centroids.

- **Yelp:** The Yelp Dataset Challenge is a library of restaurants (and related businesses) and their reviews from customers. We pick $200,000$ users that have reviewed the highest number of businesses and $50,000$ businesses with the highest number of reviews. We sample $10,000$ businesses at random and calculate the singular value decomposition (SVD) of the corresponding $200,000 \times 10,000$ ratings matrix into 10 principal components. The projection matrices from the SVD are used to compute embeddings of dimension $d = 10$ for the remaining $40,000$ businesses for training the bandit algorithms. The embeddings are normalized to unit L_2-norm and the dataset is shuffled randomly. In every round of bandit learning, the algorithm is presented with a non-overlapping chunk of movies as arms (A_t). The chunk size is 100. The outcome of an arm is decided by the mean rating received by the corresponding movie in the dataset. If this mean rating is greater than its median value in the dataset, the outcome is a success, else is a failure. As mentioned in Section 2.2, for the dependent algorithm the $40,000$ SVD-projected d-dimensional vectors have been used to compute coverage vectors through a GMM with d' centroids.

- **MNIST:** The MNIST dataset consists of $60,000$ training samples and $10,000$ test samples. We draw $19,800$ samples at random from the training split for constructing a $d = 10$-dimensional embedding space using Principal Component Analysis (PCA) and combine the remaining training samples with the test samples and randomly shuffle it to create a dataset of $50,200$ samples for training the bandit algorithm. As mentioned in Section 2.2, for the dependent algorithm the $50,200$ SVD-projected 10-dimensional vectors are used to compute coverage vectors through a GMM with d' centroids. All observed vectors (embeddings and coverage vectors) are scaled to unit L_2-norm.

MNIST has 10 output classes. For each of these output classes, we define a sub-task that considers that class as the "pivot-class". At every round of bandit learning, we present the agent with a non-overlapping chunk of examples as arms. The agent observes success only if it chooses an arm whose output class matches the pivot class. We choose the pivot class at the beginning of each experiment and keep it constant throughout.

D.2 Results

Our additional experimental results are summarized in Tables 1 through 7, and Figure 4.

Tables $1-5$ contain the final cumulative reward CR achieved by the tested algorithms on the three datasets MSD, Yelp, and MovieLens at the end of bandit training, as we vary the budget parameter b_t across the values $1, 5, 10, 50, 100$. Table 7 has a similar content on the MNIST dataset, but in aggregate form over the 10 pivot classes. Finally, Figure 4 is simply the Dep counterpart to Figure 2 in the main body of the paper.

Observe that since vanilla rewards do not distinguish between early and late successes in the sequence, for larger values of b_t the performances of all algorithms become indistinguishable from one another. Besides, when $b_t = 50$ or $b_t = 100$ also Rand performs as well as all other algorithms. This is made evident in all tables. This is not the case for the exponential scenario where, given the decreasing values of rewards and losses, early successes (or early give ups) are always more profitable than late successes (or late give ups).

In most cases, Dep turns out to be the best performer, especially in the exponential scenario. In the MNIST dataset, the Inv versions of Ind and Dep tend to be competitive only in the vanilla scenario.

In the vanilla scenario, GL-CDCM turns out to be a good competitor, often at par with Ind or even superior to it, but still worse than Dep. Besides, it should be emphasized that the MLE estimation contained in GL-CDCM makes its running time far higher than that of Ind and Dep. On the other hand, C-UCB tends to be worse than all other algorithms (Excluding Rand and Eps).

Finally, a quick comparison between Figure 4 and Figure 2 reveals that Dep tends to produce longer sequences than Ind, but also to achieve slightly earlier successes.

To summarize, from these experiments, the following trends emerge.

1. In a vanilla scenario that emphasizes early success (b_t small), the baseline algorithms (Eps, C-UCB, GL-CDCM) are rarely the winner. In most cases, the winner is the proposed dependent (Dep) algorithm. On the other hand, as the budget b_t grows the algorithms tend to be indistinguishable. This has to be expected, as when b_t is large even the random policy (Rand) becomes competitive in the vanilla scenario.

2. In the exponential scenario, Dep generally outperforms Ind, with the exception of a few pivot classes in the MNIST dataset and on MovieLens with $b_t \geq 5$.

3. Ind and Dep are clearly outperforming their corresponding "Inv" variants Ind-Inv and Dep-Inv, with a few exceptions on the MNIST dataset. This finding seems to contradict the experimental results reported in Kveton et al. (2015a).

4. C-UCB is always inferior to most of their competitors, while GL-CDCM is sometimes superior to Ind, but still worse than Dep.

5. As for a deeper understanding of the behavior of Ind and Dep in the exponential scenario, our experiments reveal that: (i) neither Ind nor Dep do saturate their budget length, and (ii) Ind produces shorter sequences than Dep, though on these datasets Dep tends to achieve success slightly earlier in the list.

Table 1: Comparison of cumulative reward at the end of training on the Million Songs Dataset for the Vanilla reward scenario. "Rand" refers to the random policy, "Eps" is the ϵ-greedy version of our Algorithm 1, "C-UCB" is the cascading bandit algorithm of Zong et al. (2016), while "GL-CDCM" is the one from Liu et al. (2018b). Moreover, "Ind" and "Dep" are abbreviations for the Independent (Algorithm 1) and Dependent (Algorithm 3) algorithms proposed in this paper. Finally, "Ind-Inv" and "Dep-Inv" are the versions of "Ind" and "Dep" where the list is presented in reverse order (as suggested by Kveton et al. (2015a); Combes et al. (2015)). Standard deviations for the randomized algorithms (Rand and Eps) over 100 repetitions are in braces. For each value of b_t, we emphasize in bold the best performance.

	Rand	**Eps**	**C-UCB**	**GL-CDCM**	**Ind-Inv**	**Dep-Inv**	**Ind**	**Dep**
$b_t = 1$	199.1(9.7)	309.0(0.1)	286.0	314.0	328.0	**355.0**	328.0	**355.0**
$b_t = 5$	386.4(3.5)	373.6(5.0)	396.0	**399.0**	**399.0**	**399.0**	**399.0**	**399.0**
$b_t = 10$	**398.6(0.6)**	392.7(2.1)	**399.0**	**399.0**	**399.0**	**399.0**	**399.0**	**399.0**
$b_t = 50$	**399.0(0.0)**	**399.0(0.0)**	**399.0**	**399.0**	**399.0**	**399.0**	**399.0**	**399.0**
$b_t = 100$	**399.0(0.0)**	**399.0(0.0)**	**399.0**	**399.0**	**399.0**	**399.0**	**399.0**	**399.0**

Table 2: Same as in Table 1 with the exponential reward scenario. Notice that scenario does not include the baselines "Eps", "C-UCB" and "GL-CDCM" since those baselines are defined to work only in the vanilla scenario.

	Rand	**Ind-Inv**	**Dep-Inv**	**Ind**	**Dep**
$b_t = 1$	81.8(15.8)	285.4	**328.6**	285.4	**328.6**
$b_t = 5$	253.9(8.6)	337.2	360.4	356.4	**376.4**
$b_t = 10$	265.8(6.8)	329.2	354.1	356.4	**376.4**
$b_t = 50$	266.5(6.8)	329.2	347.4	356.4	**376.4**
$b_t = 100$	266.5(7.2)	329.2	347.4	356.4	**376.4**

Table 3: Same as in Table 1 for the Yelp dataset.

	Rand	**Eps**	**C-UCB**	**GL-CDCM**	**Ind-Inv**	**Dep-Inv**	**Ind**	**Dep**
$b_t = 1$	199.4(9.9)	251.2(2.5)	246.0	291.0	275.0	**330.0**	275.0	**330.0**
$b_t = 5$	386.9(3.4)	361.4(6.1)	396.0	398.0	398.0	**399.0**	**399.0**	**399.0**
$b_t = 10$	**398.6(0.6)**	389.3(3.2)	**399.0**	**399.0**	**399.0**	**399.0**	**399.0**	**399.0**
$b_t = 50$	**399.0(0.0)**	**399.0(0.0)**	**399.0**	**399.0**	**399.0**	**399.0**	**399.0**	**399.0**
$b_t = 100$	**399.0(0.0)**	**399.0(0.0)**	**399.0**	**399.0**	**399.0**	**399.0**	**399.0**	**399.0**

Table 4: Same as in Table 2 for the Yelp dataset.

	Rand	**Ind-Inv**	**Dep-Inv**	**Ind**	**Dep**
$b_t = 1$	79.9(16.1)	200.6	**288.6**	200.6	**288.6**
$b_t = 5$	253.9(9.7)	310.7	340.1	325.8	**358.7**
$b_t = 10$	265.7(7.4)	290.2	322.5	325.8	**361.3**
$b_t = 50$	265.6(7.3)	301.8	314.2	326.0	**361.3**
$b_t = 100$	265.4(7.1)	301.4	314.2	326.0	**361.3**

Table 5: Same as in Table 1 for the Movielens dataset.

	Rand	Eps	C-UCB	GL-CDCM	Ind-Inv	Dep-Inv	Ind	Dep
$b_t = 1$	244.7(11.2)	431.0(0.2)	406.0	456.0	453.0	**461.0**	453.0	**461.0**
$b_t = 5$	474.7(3.7)	464.9(16.2)	489.0	**490.0**	**490.0**	**490.0**	**490.0**	**490.0**
$b_t = 10$	**489.5(0.7)**	483.9(4.7)	**490.0**	**490.0**	**490.0**	**490.0**	**490.0**	**490.0**
$b_t = 50$	**490.0(0.0)**	**490.0(0.0)**	**490.0**	**490.0**	**490.0**	**490.0**	**490.0**	**490.0**
$b_t = 100$	**490.0(0.0)**	**490.0(0.0)**	**490.0**	**490.0**	**490.0**	**490.0**	**490.0**	**490.0**

Table 6: Same as in Table 2 for the Movielens dataset.

	Rand	Ind-Inv	Dep-Inv	Ind	Dep
$b_t = 1$	97.3(17.7)	430.8	**442.0**	430.8	**442.0**
$b_t = 5$	311.9(9.6)	429.9	446.4	**475.5**	472.6
$b_t = 10$	326.2(7.8)	420.8	442.7	**476.0**	472.6
$b_t = 50$	326.8(7.9)	404.8	440.3	**476.0**	472.6
$b_t = 100$	326.6(7.9)	390.2	440.3	**476.0**	472.6

Table 7: Number of wins of each algorithm out of the 10 sub-problems of the MNIST dataset. Ties are broken by splitting the score equally among the winners. E.g., in the exponential scenario with $b_t = 10$, Dep at score 7.5 means that Dep turned out to be the winner in 7 out of the 10 sub-problems, and tied with Dep-Inv in one of the remaining 3. For each of the two scenarios and each value of b_t, we emphasize in bold the best performance.

	Vanilla Reward scenario								Exponential Reward scenario				
	Rand	Eps	C-UCB	GL-CDCM	Ind-Inv	Dep-Inv	Ind	Dep	Rand	Ind-Inv	Dep-Inv	Ind	Dep
$b_t = 1$	0	1	**2**	0	1	**2**	**2**	**2**	0	**3**	2	**3**	2
$b_t = 5$	0	0	1.5	1.25	2.5	**2.75**	0.25	1.75	0	0	2	1	**7**
$b_t = 10$	0	0	0.5	1.33	1	**3.58**	1.25	2.33	0	0	2.5	0	**7.5**
$b_t = 50$	0	0	**1.67**	**1.67**	**1.67**	**1.67**	**1.67**	**1.67**	0	0	0	**5**	**5**
$b_t = 100$	**1.43**	0	**1.43**	**1.43**	**1.43**	**1.43**	**1.43**	**1.43**	0	0	0	**5**	**5**

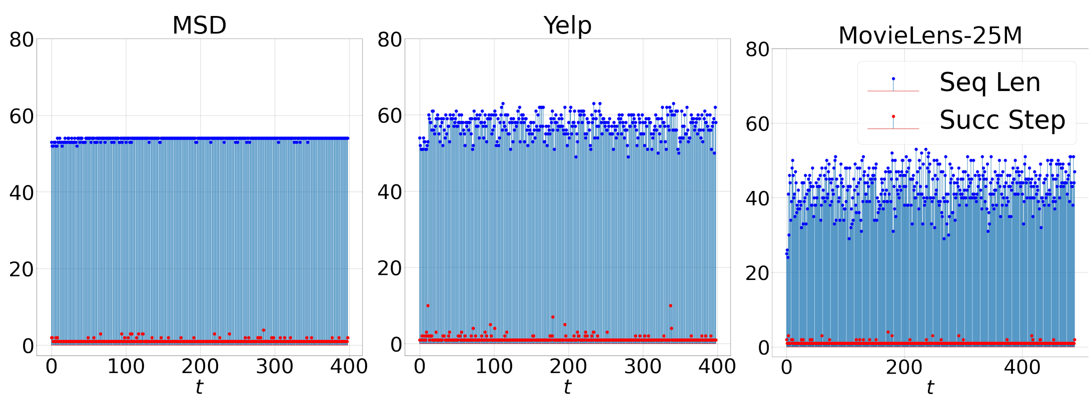

Figure 4: Dep operating on the three datasets MSD (left), Yelp (middle), and MovieLens (right) in the exponential scenario with $b_t = 100$. The plots report, for each chunk of the datasets (x-axis), the length $\widehat{s}_t$ of the sequence J_t computed by Ind ("Seq Len") along with the position where the first success is observed ("Succ Step"), that is, value $\widehat{s}'_t$ for J_t (y-axis) – please recall the notation in Algorithm 1. Chunks where success is not observed are excluded. The algorithm never saturates budget b_t, while achieving success within the first few items. In particular, for all t where success is achieved, we have $\widehat{s}'_t \leq 4$ on MSD, $\widehat{s}'_t \leq 10$ on Yelp, and $\widehat{s}'_t \leq 4$ on MovieLens-25M.

E CO2 EMISSION RELATED TO EXPERIMENTS

Experiments were conducted using Google Cloud Platform in region europe-west1, which has a carbon efficiency of 0.27 kgCO_2eq/kWh. A cumulative of 5000 hours of computation was performed on hardware of type Intel Xeon E5-2699 (TDP of 145W).

Total emissions are estimated to be 195.75 kgCO_2eq of which 100 percents were directly offset by the cloud provider. Estimations were conducted using the MachineLearning Impact calculator presented in Lacoste et al. (2019).

Algorithm 3 The contextual bandit algorithm in the dependent case with link function $\sigma(x) = \frac{\exp(x)}{1+\exp(x)}$.

Input: Confidence level $\delta \in [0, 1]$, width parameter $D > 0$, maximal budget parameter $b > 0$;
Init: $M_0 = bI \in \mathbb{R}^{d' \times d'}$, $w_1 = 0 \in \mathbb{R}^{d'}$, $c_1 = 1$
For $t = 1, 2, \ldots, T$:

1. Get:
   - Set of actions $A_t = \{x_{1,t}, \ldots, x_{|A_t|,t}\} \subseteq \{x \in \mathbb{R}^{d'} : ||x|| \le 1\}$,
   - budget $b_t \le b$;
2. Compute J_t :
   - **For** $k = 1, \ldots, \min\{b_t, |A_t|\}$:
   $$x_{\widehat{j}_{t,k}} = \arg \max_{x \in A_t \setminus \{x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}}\}} \sigma\Big(\bar{c}(x \mid x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}})^\top w_{c_t} + \epsilon_t(x \mid x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}})\Big) ,$$
   where $\epsilon_t^2(x \mid x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}}) = \bar{c}(x \mid x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}})^\top M_{c_t-1}^{-1} \bar{c}(x \mid x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}})\, \alpha(b, d', T, \delta, D)$
   - Let $\widehat{J}_{t,s} = \langle x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,s}} \rangle$ for any $s \le b_t$;
   - Set $\widehat{s}_t = \arg \max_{s=0,1,\ldots,b_t} \widehat{\mathbb{E}}_{Y_t}[R(\widehat{J}_{t,s}, Y_t)]$, with
   $$\begin{aligned}
   \hat{\Delta}_{\widehat{j}_{t,k},t} &= \bar{c}(x_{\widehat{j}_{t,k}} \mid x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}})^\top w_{c_t} \\
   \epsilon^2_{\widehat{j}_{t,k},t} &= \bar{c}(x_{\widehat{j}_{t,k}} \mid x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}})^\top M_{c_t-1}^{-1}\, \bar{c}(x_{\widehat{j}_{t,k}} \mid x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}})\ \alpha(b, d', T, \delta, D) \\
   \widehat{\mathbb{E}}_{Y_t}[R(\widehat{J}_{t,s}, Y_t)] &= \begin{cases} E\left(\hat{\Delta}_{\widehat{j}_{t,1},t} + \epsilon_{\widehat{j}_{t,1},t}, \ldots, \hat{\Delta}_{\widehat{j}_{t,s},t} + \epsilon_{\widehat{j}_{t,s},t}\right) & \text{if } s \ge 1 \\ \ell_{0,t} & \text{otherwise ,} \end{cases}
   \end{aligned}$$
   where function $E(\cdot, \ldots, \cdot)$ is as (6) in Algorithm 1;
   - Finally, $J_t = \widehat{J}_{t,\widehat{s}_t}$;
3. Observe feedback $Y_t \downarrow J_t = \begin{cases} \langle y_{t,\widehat{j}_{t,1}}, y_{t,\widehat{j}_{t,2}}, \ldots, y_{t,\widehat{j}_{t,\widehat{s}'_t}} \rangle = \langle 0, \ldots, 0, 1 \rangle, & \text{for some } \widehat{s}'_t \le \widehat{s}_t \qquad \text{or} \\ \langle y_{t,\widehat{j}_{t,1}}, y_{t,\widehat{j}_{t,2}}, \ldots, y_{t,\widehat{j}_{t,\widehat{s}_t}} \rangle = \langle 0, \ldots, 0, 0 \rangle \end{cases}$
4. **For** $k = 1, \ldots, \widehat{s}_t$ (in the order of occurrence of items in J_t) update :
   $$M_{c_t+k-1} = M_{c_t+k-2} + |s_{k,t}|\ \bar{c}(x_{\widehat{j}_{t,k}} \mid x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}})\ \bar{c}(x_{\widehat{j}_{t,k}} \mid x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}})^\top ,$$
   $$w_{c_t+k} = w'_{c_t+k-1} + \frac{1}{c_{\sigma'}} M_{c_t+k-1}^{-1} \nabla_{k,t} ,$$
   where
   $$s_{k,t} = \begin{cases} 1 & \text{If } y_{t,k} \text{ is observed and } y_{t,k} = 1 \\ -1 & \text{If } y_{t,k} \text{ is observed and } y_{t,k} = 0 \\ 0 & \text{If } y_{t,k} \text{ is not observed ,} \end{cases}$$
   and $\nabla_{k,t} = \sigma(-s_{k,t}\, \hat{\Delta}'_{k,t})\, s_{k,t}\, \bar{c}(x_{\widehat{j}_{t,k}} \mid x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}})$, where $\hat{\Delta}'_{k,t} = \bar{c}(x_{\widehat{j}_{t,k}} \mid x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}})^\top w'_{c_t+k-1}$ with
   $$w'_{c_t+k-1} = \arg \min_{w\,:\, -D \le w^\top \bar{c}(x_{\widehat{j}_{t,k}} \mid x_{\widehat{j}_{t,1}}, \ldots, x_{\widehat{j}_{t,k-1}}) \le D} d_{c_t+j-2}(w, w_{c_t+k-1}) ;$$
5. $c_{t+1} \leftarrow c_t + \widehat{s}_t$.