Journal of Machine Learning Research 14 (2013) 399-436 Submitted 10/10; Revised 9/12; Published 1/13

Ranked Bandits in Metric Spaces: Learning Diverse Rankings over Large Document Collections*

Aleksandrs Slivkins SLIVKINS@MICROSOFT.COM
Microsoft Research Silicon Valley
1065 La Avenida
Mountain View, CA 94043, USA

Filip Radlinski FILIPRAD@MICROSOFT.COM
Microsoft Research Cambridge
7 J.J. Thomson Ave.
Cambridge UK

Sreenivas Gollapudi SREENIG@MICROSOFT.COM
Microsoft Research Silicon Valley
1065 La Avenida
Mountain View, CA 94043, USA

Editor: Nicolo Cesa-Bianchi

Abstract

Most learning to rank research has assumed that the utility of different documents is independent, which results in learned ranking functions that return redundant results. The few approaches that avoid this have rather unsatisfyingly lacked theoretical foundations, or do not scale. We present a learning-to-rank formulation that optimizes the fraction of satisfied users, with several scalable algorithms that explicitly takes document similarity and ranking context into account. Our formulation is a non-trivial common generalization of two multi-armed bandit models from the literature: *ranked bandits* (Radlinski et al., 2008) and *Lipschitz bandits* (Kleinberg et al., 2008b). We present theoretical justifications for this approach, as well as a near-optimal algorithm. Our evaluation adds optimizations that improve empirical performance, and shows that our algorithms learn orders of magnitude more quickly than previous approaches.

Keywords: online learning, clickthrough data, diversity, multi-armed bandits, contextual bandits, regret, metric spaces

1. Introduction

Identifying the most relevant results to a query is a central problem in web search, hence learning ranking functions has received a lot of attention (e.g., Joachims, 2002; Burges et al., 2005; Chu and Ghahramani, 2005; Taylor et al., 2008). One increasingly important goal is to learn from user interactions with search engines, such as clicks. We address the task of learning a ranking function that minimizes the likelihood of *query abandonment*: the event that the user does not click on any of the search results for a given query. This objective is particularly interesting as query abandonment

*. Preliminary versions of this paper has been published as a conference paper in *ICML 2010* and as a technical report at `arxiv.org/abs/1005.5197` (May 2010). Compared to the conference version, this paper contains full proofs and a significantly revised presentation.

is a major challenge in today's search engines, and is also sensitive to the diversity and redundancy among documents presented.

We consider the Multi-Armed Bandit (MAB) setting (e.g., Cesa-Bianchi and Lugosi, 2006), which captures many online learning problems wherein an algorithm chooses sequentially among a fixed set of alternatives, traditionally called "arms". In each round an algorithm chooses an arm and collects the corresponding reward. Crucially, the algorithm receives limited feedback—only for the arm it has chosen, which gives rise to the tradeoff between *exploration* (acquiring new information) and *exploitation* (taking advantage of the information available so far).

While most of the literature on MAB corresponds to learning a single best alternative, MAB algorithms can also be extended to learning a ranking of documents that minimizes query abandonment (Radlinski et al., 2008; Streeter and Golovin, 2008). In this setting, called *Ranked Bandits*, in each round an algorithm chooses an *ordered list* of k documents from some fixed collection of documents, and receives clicks on some of the chosen documents. Crucially, the click probability for a given document may depend on the documents shown above: a user scrolls the list top-down and may leave as soon as she clicked on the first document. The goal is to minimize query abandonment.

Radlinski et al. (2008) and Streeter and Golovin (2008) propose a simple but effective approach: for each position in the ranking there is a separate instance bandit algorithm which is responsible for choosing a document for this position. However, the specific algorithms they considered are impractical at WWW scales.

Prior work on MAB algorithms has considered exploiting structure in the space of arms to improve convergence rates. One particular approach, articulated by Kleinberg et al. (2008b) is well suited to our scenario: when the arms form a metric space and the payoff function satisfies a Lipschitz condition with respect to this metric space. The metric space provides information about similarity between arms, which allows the algorithm to make inferences about similar arms without exploring them. Further, they propose a "zooming algorithm" which partitions the metric space into regions (and treats each region as a "meta-arm") so that the partition is adaptively refined over time and becomes finer in regions with higher payoffs.

In web search, a metric space directly models similarity between documents. (It is worth noting that most offline learning-to-rank approaches also rely on similarity between documents, at least implicitly.)

Our contributions. This paper initiates the study of bandit learning-to-rank with side information on similarity between documents. We adopt the Ranked bandits setup: a user scrolls the results top-down and may leave after a single click, the goal is to minimize query abandonment. The similarity information is expressed as a metric space.

In this paper we consider a "perfect world" scenario: there exists an informative distance function which meaningfully describes similarity between documents in a ranked setting, and an algorithm has access to such function. We focus on two high-level questions: How to represent the knowledge of document similarity, and how to use it algorithmically in a bandit setting. We believe that studying such "perfect world" scenario is useful, and perhaps necessary, to inform and guide the corresponding data-driven work.

We propose a simple bandit model which combines *Ranked bandits* (Radlinski et al., 2008) and *Lipschitz bandits* (Kleinberg et al., 2008b), and admits efficient bandit algorithms that, unlike those in prior work on bandit learning-to-rank, scale to large document collections. Our model is based on the new notion of "conditional Lipschitz continuity" which asserts that similar documents have similar click probabilities even conditional on the event that all documents in a given set

of documents are skipped (i.e., not clicked on) by the current user. We study this model both theoretically and empirically.

First, we validate the expressiveness of our model by providing an explicit construction for a wide family of plausible user distributions which provably fit the model. The analysis of this construction is perhaps the most technical contribution of this paper. We also use this construction in simulations.

Second, we put forth a battery of algorithms for our model. Some of these algorithms are straightforward combinations of ideas from prior work on Ranked bandits and Lipschitz bandits, and some are new.

A crucial insight in the new algorithms is that for each position i in the ranking there is a *context* that we can use, namely the set of documents chosen for the above positions in the same round. Indeed, since our objective is non-abandonment we only care about position i if all documents shown above i have been skipped in the present round. So the algorithm responsible for position i can simply *assume* that these documents have been skipped.

This interpretation of contexts allows us to cast the position-i problem as a *contextual bandit* problem. Moreover, we derive a Lipschitz condition on contexts (with respect to a suitably defined metric), which allows us to use the contextual Lipschitz MAB machinery from Slivkins (2009). We also exploit correlations between clicks: if a given document is included in the context—that is, if this document is skipped by the current user—then similar documents are likely to be skipped, too. More specifically, we propose two algorithms that use contexts: a "heavy-weight" algorithm which uses both the metric on contexts and correlated clicks, and a "light-weight" algorithm which uses correlated clicks but not the metric on contexts.

Third, we provide scalability guarantees for the heavy-weight contextual algorithm, proving that the convergence rate depends only on the dimensionality of the metric space but not on the number of documents. However, we argue that our provable guarantees do not fully reflect the power of the algorithm, and outline some directions for the follow-up theoretical work. In particular, we identify a stronger benchmark and discuss convergence to this benchmark. We provide an initial result: we prove, without any guarantees on the convergence rate, that the heavy-weight contextual algorithm indeed converges to this stonger benchmark. This theoretical discussion is one of the contributions.

Finally, we empirically study the performance of our algorithms. We run a large-scale simulation using the above-mentioned construction with realistic parameters. The main goal is to compare the convergence rates of the various approaches. In particular, we confirm that metric-aware algorithms significantly outperform the metric-oblivious ones, and that taking the context into account improves the convergence rate. Somewhat surprisingly, our light-weight contextual algorithm performs better than the heavy-weight one.

A secondary, smaller-scale experiment studies the limit behaviour of the algorithms, that is, the query abandonment probability that the algorithms converge to. Following the theoretical discussion mentioned above, we design a principled example on which different algorithms exhibit very different limit behaviour. Interestingly, the heavy-weight contextual algorithm is the only algorithm that achieves the optimal limit behaviour in this experiment.

Map of the paper. We start with a brief survey of related work (Section 2). We define our model in Section 3, and validate its expressiveness in Section 4. In-depth discussion of relevant approaches from prior work is in Section 5. Our new approach, ranked contextual bandits in metric spaces, is presented in Section 6. Scalability guarantees are discussed in Section 7. We present our simulations in Section 8.

To keep the flow of the paper, the lengthy proofs for the theoretical results in Section 4 are presented in Section A and Section B. Moreover, the background on instance-dependent regret bounds for UCB1-style algorithms is discussed in Appendix C.

2. Related Work on Multi-Armed Bandits

Multi-armed bandits have been studied for many decades as a simple yet expressive model for understanding exploration-exploitation tradeoffs. A thorough discussion of the literature on bandit problems is beyond the scope of this paper. For background, a reader can refer to a book (Cesa-Bianchi and Lugosi, 2006) and a recent survey (Bubeck and Cesa-Bianchi, 2012) on regret-minimizing bandits.[1] A somewhat different, Bayesian perspective can be found in surveys (Sundaram, 2005; Bergemann and Välimäki, 2006).

On a very high level, there is a crucial distinction between regret-minimizing formulations and Bayesian/MDP formulations (see the surveys mentioned above); this paper follows the former. Among regret-minimizing formulations, an important distinction is between stochastic rewards (Lai and Robbins, 1985; Auer et al., 2002a) and adversarial rewards (Auer et al., 2002b).

Below we survey several directions that are directly relevant to this paper.

Ranked bandits. A bandit model in which an algorithm learns a ranking of documents with a goal to minimize query abandonment has been introduced in Radlinski et al. (2008) under the name *ranked bandits*. A crucial feature in this setting is that the click probability for a given document may depend not only on the document and the position in which it is shown, but also the documents shown above. In particular, documents shown above can "steal" clicks from the documents shown below, in the sense that a user scrolls the list top-down and may leave as soon as she clicked on the first document.

Independently, Streeter and Golovin (2008) considered a more general model where the goal is to minimize an arbitrary (known) submodular set function, rather than query abandonment. A further generalization to submodular functions on ordered assignements (rather than on sets) was considered in (Golovin et al., 2009). The contributions of the three papers essentially coincide for the special case of ranked bandits.

Uchiya et al. (2010)[2] and Kale et al. (2010)[2] considered a related bandit model in which an algorithm selects a ranking of documents in each round, but the click probabilities for a given document do not depend on which other documents are shown to the same user.

Bandits with structure. Numerous papers enriched the basic MAB setting by assuming some structure on arms, typically in order to handle settings where the number of arms is very large or infinite. Most relevant to this paper is the model where arms lie in a metric space and their expected rewards satisfy the Lipschitz condition with respect to this metric space (see Section 3 for details). This model, for a general metric space, has been introduced in Kleinberg et al. (2008b) under the name *Lipschitz MAB*; the special case of unit interval has been studied in (Agrawal, 1995; Kleinberg, 2004; Auer et al., 2007) under the name *continuum-armed bandits*. Subsequent work on Lipschtz MAB includes Bubeck et al. (2011), Kleinberg and Slivkins (2010), Maillard and Munos (2010), Slivkins (2009) and Slivkins (2011). A closely related model posits that arms corresponds to leaves

1. Regret of an algorithm in T rounds, typically denoted $R(T)$, is the expected payoff of the benchmark in T rounds minus that of the algorithm. A standard benchmark is the best arm in hindsight.
2. This is either concurrent or subsequent work with respect to the conference publication of this paper.

on a tree, but no metric space is revealed to the algorithm (Kocsis and Szepesvari, 2006; Pandey et al., 2007; Munos and Coquelin, 2007; Slivkins, 2011).

Another commonly assumed structure is linear or convex payoffs (e.g., Awerbuch and Kleinberg, 2008; Flaxman et al., 2005; Dani et al., 2007; Abernethy et al., 2008; Hazan and Kale, 2009). Linear/convex payoffs is a much stronger assumption than similarity, essentially because it allows to make strong inferences about far-away arms. Other structural assumptions have been considered, for example, Wang et al. (2008) and Bubeck and Munos (2010) and Srinivas et al. (2010)[2].

The distinction between the various possible structural assumptions is orthogonal to the distinction between stochastic and adversarial rewards. With a few exceptions, papers on MAB with linear/convex payoffs allow adversarial payoffs, whereas papers on MAB with similarity information focus on stochastic payoffs

Contextual bandits. Here in each round the algorithm receives a *context*, chooses an arm, and the reward depends both on the arm and the context. The term "contextual bandits" was coined in Langford and Zhang (2007). The setting, with a number of different modifications, has been introduced independently in several papers; a possibly incomplete list is Woodroofe (1979), Auer et al. (2002b), Auer (2002), Wang et al. (2005), Langford and Zhang (2007), Hazan and Megiddo (2007) and Pandey et al. (2007).

There are several models for how contexts are related to rewards: rewards are linear in the context (e.g., Auer, 2002; Langford and Zhang, 2007) and Chu et al. (2011)[2], the context is a random variable correlated with rewards (Woodroofe, 1979; Wang et al., 2005; Rigollet and Zeevi, 2010); rewards are Lipschitz with respect to a metric space on contexts (Hazan and Megiddo, 2007; Slivkins, 2009) and Lu et al. (2010)[2].

Most work on contextual bandits has been theoretical in nature; experimental work on contextual MAB includes Pandey et al. (2007) and Li et al. (2010, 2011)[2].

3. Problem Formalization: Ranked Bandits in Metric Spaces

Let us introduce the online learning-to-rank problem that we study in this paper.

Ranked bandits. Following Radlinski et al. (2008), we are interested in learning an optimally diverse ranking of documents for a given query. We model it as a *ranked bandit* problem as follows. Let X be a set of documents ("arms"). Each 'user' is represented by a binary *relevance vector*: a function $\pi : X \to \{0,1\}$. A document $x \in X$ is called "relevant" to the user if and only if $\pi(x) = 1$. Let $\mathcal{F}_X$ be the set of all possible relevance vectors. Users come from a distribution $\mathcal{P}$ on $\mathcal{F}_X$ that is fixed but not revealed to an algorithm.[3] This $\mathcal{P}$ will henceforth be called the *user distribution*.

In each round, the following happens: a user arrives, sampled independently from $\mathcal{P}$; an algorithm outputs a list of k documents; the user scans this list top-down, and clicks on the first relevant document. The goal is to maximize the expected fraction of *satisfied users*: users who click on at least one document. Note that in contrast with prior work on diversifying existing rankings (e.g., Carbonell and Goldstein, 1998), the algorithm needs to directly learn a diverse ranking.

Since we count satisfied users rather than the clicks themselves, we can assume w.l.o.g. that a user leaves once she clicks once. (Alternatively, the algorithm does not record any subsequent clicks.) A user is satisfied or not satisfied independently of the order in which she scans the results. However, the assumption of the top-down scan determines the feedback received by the algorithm, that is, which document gets clicked.

3. This also models users for whom documents are probabilistically relevant (Radlinski et al., 2008).

We will say that there are *k slots* to be filled in each round, so that when the algorithm outputs the list of k documents, the i-th document in this list appears in slot i. Note that the standard model of MAB with stochastic rewards (e.g., Auer et al., 2002a) is a special case with a single slot ($k = 1$).

Click probabilities. Recall that $\mathcal{P}$ is a distribution over relevance vectors. The *pointwise mean* of $\mathcal{P}$ is a function $\mu : X \to [0,1]$ such that $\mu(x) \triangleq \mathbb{E}_{\pi\sim\mathcal{P}}[\pi(x)]$. Thus, $\mu(x)$ is the click probability for document x if it appears in the top slot.

Each slot $i > 1$ is examined by the user only in the event that all documents in the higher slots are not clicked, so the relevant click probabilities for this slot are conditional on this event. Formally, fix a subset of documents $S \subset X$ and let $Z_S \triangleq \{\pi(\cdot) = 0 \text{ on } S\}$ be the event that all documents in S are not relevant to the user. Let $(\mathcal{P}|Z_S)$ be the distribution of users obtained by conditioning $\mathcal{P}$ on this event, and let $\mu(\cdot\,|Z_S)$ be its pointwise mean. Then $\mu(x\,|Z_S)$ is the click probability for document x if S is the set of documents shown above x in the same round.

Metric spaces. Throughout the paper, let (X,D) be a *metric space*.That is, X is a set and D is a symmetric function on $X \times X \to [0,\infty]$ such that $D(x,y) = 0 \iff x = y$, and $D(x,y) + D(y,z) \geq D(x,z)$ (triangle inequality).

A function $\nu : X \to \mathbb{R}$ is said to be *Lipschitz-continuous* with respect to (X,D) if

$$|\nu(x) - \nu(y)| \leq D(x,y) \qquad \text{for all } x,y \in X. \tag{1}$$

Throughout the paper, we will write *L-continuous* for brevity.

A user distribution $\mathcal{P}$ is called L-continuous with respect to (X,D) if its pointwise mean μ is L-continuous with respect to (X,D).

Document similarity. To allow us to incorporate information about similarity between documents, we start with the model, called *Lipschitz MAB*, proposed by Kleinberg et al. (2008b) for the standard (single-slot) bandits. In this model, an algorithm is given a metric space (X,D) with respect to which the pointwise mean μ is L-continuous.[4]

While this model suffices for learning the document at the top slot (see Kleinberg et al., 2008b for details), it is not sufficiently informative for lower slots. This is because the relevant click probabilities $\mu(\cdot\,|Z_S)$ are conditional and therefore are not directly constrained by L-continuity. To enable efficient learning in all k slots, we will assume a stronger property called *conditional L-continuity*:

Definition 1 *$\mathcal{P}$ is called* conditionally *L-continuous w.r.t. (X,D) if the conditional pointwise mean $\mu(\cdot|Z_S)$ is L-continuous for all $S \subset X$.*

Now, a document x in slot $i > 1$ is examined only if event Z_S happens, where S is the set of documents in the higher slots: that is, if all documents in the higher slots are not relevant to the user. The document x has a conditional click probability $\mu(x|Z_S)$. The function $\mu(\cdot\,|Z_S)$ satisfies the Lipschitz condition (1), which will allow us to use the machinery from MAB problems on metric spaces.

Formally, we define the *k-slot Lipschitz MAB problem*, an instance of which consists of a triple $(X,D,\mathcal{P})$, where (X,D) is a metric space that is known to an algorithm, and $\mathcal{P}$ is a latent user distribution which is conditionally L-continuous w.r.t. (X,D).

4. One only needs to assume that similarity between any two documents x,y is summarized by a number $\delta_{x,y}$ such that $|\mu(x) - \mu(y)| \leq \delta_{x,y}$. Then one obtains a metric space by taking the shortest paths closure.

Note that the k-slot Lipschitz MAB problem subsumes the "metric-free" ranked bandit problem from Radlinski et al. (2008) (as a special case with a trivial metric space in which all distances are equal to 1) and the Lipschitz MAB problem from Kleinberg et al. (2008b) (as a special case with a single slot).

3.1 Metric Space: A Running Example

Web documents are often classified into hierarchies, where closer pairs are more similar.[5] For evaluation, we assume the documents X fall in such a tree, with each document $x \in X$ a leaf in the tree. On this tree, we consider a very natural metric: the distance between any two tree nodes u, v is exponential in the height (i.e., the hop-count distance to the root) of their least common ancestor:

$$D(u,v) = c \times \varepsilon^{\texttt{height}(\texttt{LCA}(u,v))},$$

for some constant c and base $\varepsilon \in (0,1)$. We call this the ε-*exponential tree metric* (with constant c). However, our algorithms and analyses extend to arbitrary metric spaces.

3.2 Alternative Notion of Document Similarity

An alternative notion of document similarity focuses on *correlated relevance*: correlation between the relevance of two documents to a given user. We express "similarity" by bounding the probability of the "discorrelation event" $\{\pi(x) \neq \pi(y)\}$. Specifically, we consider *conditional L-correlation*, defined as follows:

Definition 2 *Call* $\mathcal{P}$ L-correlated *w.r.t.* (X,D) *if*

$$\Pr_{\pi\sim\mathcal{P}}[\pi(x) \neq \pi(y)] \leq D(x,y) \quad \forall x,y \in X. \tag{2}$$

Call $\mathcal{P}$ conditionally L-correlated *w.r.t.* (X,D) *if (2) holds conditional on* Z_S *for any* $S \subset X$*, that is,*

$$\Pr_{\pi\sim(\mathcal{P}|Z_S)}[\pi(x) \neq \pi(y)] \leq D(x,y) \quad \forall x,y \in X, S \subset X.$$

It is easy to see that conditional L-correlation implies conditional L-continuity. In fact, we show that the two notions are essentially equivalent. Namely, we prove that conditional L-continuity w.r.t. (X,D) implies conditional L-correlation w.r.t. $(X,2D)$.

Lemma 3 *Consider an instance* $(X,D,\mathcal{P})$ *of the k-slot Lipschitz MAB problem. Then the user distribution* $\mathcal{P}$ *is conditionally L-correlated w.r.t.* $(X,2D)$.

Proof Fix documents $x,y \in X$ and a subset $S \subset X$. For brevity, write "$x=1$" to mean "$\pi(x)=1$", etc. We claim that

$$\Pr[x=1 \wedge y=0 \,|Z_S] \leq D(x,y). \tag{3}$$

Indeed, consider the event $Z = Z_{S+\{y\}}$. Applying the Bayes theorem to $(\mathcal{P}|Z_S)$, we obtain that

$$\begin{aligned} \mu(x|Z) &= \Pr[x=1 \,|\, \{y=0\} \wedge Z_S] \\ &= \frac{\Pr[x=1 \wedge y=0 \,|Z_S]}{\Pr[y=0 \,|Z_S]}. \end{aligned} \tag{4}$$

5. One example of such hierarchical classification is the Open Directory Project (`http://dmoz.org`).

On the other hand, since $\mu(y|Z) = 0$, by conditional L-continuity it holds that

$$\mu(x|Z) = |\mu(x|Z) - \mu(y|Z)| \leq D(x,y), \tag{5}$$

so claim (3) follows from Equation (4) and Equation (5).

Likewise, $\Pr[x = 0 \wedge y = 1 \,|Z_S] \leq D(x,y)$. Since

$$\{\pi(x) \neq \pi(y)\} = \{x = 1 \wedge y = 0\} \cup \{x = 0 \wedge y = 1\},$$

it follows that $\Pr[\pi(x) \neq \pi(y) \,|Z_S] \leq 2D(x,y)$. ■

4. Expressiveness of the Model

Our approach relies on the conditional L-continuity (equivalently, conditional L-correlation) of the user distribution. How "expressive" is this assumption, that is, how rich and "interesting" is the collection of problem instances that satisfy it? While the unconditional L-continuity assumption is usually considered reasonable from the expressiveness point of view, even the unconditional L-correlation (let alone the conditional L-correlation) is a very non-trivial property about correlated relevance, and thus potentially problematic. A related concern is how to generate a suitable collection of problem instances for simulation experiments.

We address both concerns by defining a natural (albeit highly stylized) generative model for the user distribution, which we then use in the experiments in Section 8. We start with a tree metric space (X,D) and the desired pointwise mean $\mu : X \to (0, \frac{1}{2}]$ that is L-continuous w.r.t. (X,D). The generative model provides a rich family of user distributions that are conditionally L-continuous w.r.t. (X, cD), for some small c. This result is a key theoretical contribution of this paper (and by far the most technical one).

We develop the generative model in Section 4.1. We extend this result to arbitrary metric spaces in Section 4.2, and to distributions over conditionally L-continuous user distributions in Section 4.3. To keep the flow of the paper, the detailed analysis is deferred to Section A and Section B.

4.1 Bayesian Tree Network

The generative model is a tree-shaped Bayesian network with 0-1 "relevance values" $\pi(\cdot)$ on nodes, where leaves correspond to documents. The tree is essentially a topical taxonomy on documents: subtopics correspond to subtrees. The relevance value on each sub-topic is obtained from that on the parent topic via a low-probability mutation.

The mutation probabilities need to be chosen so as to guarantee conditional L-continuity and the desired pointwise mean μ. It is fairly easy to derive a necessary and sufficient condition for the pointwise mean, and a necessary condition for conditional L-continuity. The latter condition states that the mutation probabilities need to be bounded in terms of the distance between the child and the parent. The hard part is to prove that this condition is *sufficient*.

Let us describe our Bayesian tree network in detail. The network inputs a tree metric space (X,D) and the desired pointwise mean μ, and outputs a relevance vector $\pi : X \to \{0,1\}$. Specifically, we assume that documents are leaves of a finite rooted edge-weighted tree, which we denote τ_{d}, with node set V and leaf set $X \subset V$, so that D is a (weighted) shortest-paths metric on V.

Algorithm 1 User distribution for tree metrics

Input: Tree (root r, node set V); $\mu(r) \in [0,1]$
mutation probabilities $q_0, q_1 : V \to [0,1]$
Output: relevance vector $\pi : V \to \{0,1\}$

function AssignClicks(tree node v)
$b \leftarrow \pi(v)$
for each child u of v **do**
$$\pi(u) \leftarrow \begin{cases} 1-b & \text{w/prob } q_b(u) \\ b & \text{otherwise} \end{cases}$$
AssignClicks(u)

Pick $\pi(r) \in \{0,1\}$ at random with expectation $\mu(r)$
AssignClicks(r)

Recall that μ is L-continuous w.r.t. (X,D). We assume that μ takes values in the interval $[\alpha, \frac{1}{2}]$, for some constant parameter $\alpha > 0$. We show that μ can be extended from X to V preserving the range and L-continuity (see Section A for the proof).

Lemma 4 *μ can be extended to V so that $\mu : V \to [\alpha, \frac{1}{2}]$ is L-continuous w.r.t. (V,D).*

In what follows, by a slight abuse of notation we will assume that the domain of μ is V, with the same range $[\alpha, \frac{1}{2}]$, and that μ is L-continuous w.r.t. (V,D). Also, we redefine the relevance vectors to be functions $V \to \{0,1\}$ rather than $X \to \{0,1\}$.

The Bayesian network itself is very intuitive. We pick $\pi(\text{root}) \in \{0,1\}$ at random with a suitable expectation $\mu(\text{root})$, and then proceed top-down so that the child's bit is obtained from the parent's bit via a low-probability mutation. The mutation is parameterized by functions $q_0, q_1 : V \to [0,1]$, as described in Algorithm 1: for each node u, if the parent's bit is set to b then the mutation $\{\pi(u) = 1-b\}$ happens with probability $q_b(u)$. These parameters let us vary the degree of independence between each child and its parent, resulting in a rich family of user distributions.

To complete the construction, it remains to define the mutation probabilities q_0, q_1. Let $\mathcal{P}$ be the resulting user distribution. It is easy to see that μ is the pointwise mean of $\mathcal{P}$ on V if and only if

$$\mu(u) = (1-\mu(v))\, q_0(u) + \mu(v)(1-q_1(u)) \tag{6}$$

whenever u is a child of v. (For sufficiency, use induction on the tree.) Further, letting $q_b = q_b(u)$ for each bit $b \in \{0,1\}$, note that

$$\begin{aligned} \Pr[\pi(u) \neq \pi(v)] &= \mu(v)\, q_1 + (1-\mu(v))\, q_0 \\ &= \mu(v)(q_0+q_1) + (1-2\mu(v))\, q_0 \\ &\geq \mu(v)(q_0+q_1). \end{aligned}$$

Thus, if $\mathcal{P}$ is L-correlated w.r.t. (X,D) then

$$q_0(u) + q_1(u) \leq D(u,v)/\mu(v). \tag{7}$$

We show that (6-7) suffices to guarantee conditional L-continuity.

For a concrete example, one could define

$$(q_0(u), q_1(u)) = \begin{cases} \left(0, \frac{\mu(v)-\mu(u)}{\mu(v)}\right) & \text{if } \mu(v) \geq \mu(u) \\ \left(\frac{\mu(u)-\mu(v)}{1-\mu(v)}, 0\right) & \text{otherwise.} \end{cases} \tag{8}$$

The q_0, q_1 defined as above satisfy (6-7) for any μ that is L-continuous on (V, D).

The provable properties of Algorithm 1 are summarized in the theorem below. It is technically more convenient to state this theorem in terms of L-correlation rather than L-continuity.

Theorem 5 *Let D be the shortest-paths metric of an edge-weighted rooted tree with a finite leaf set X. Let $\mu : X \to [\alpha, \frac{1}{2}]$, $\alpha > 0$ be L-continuous w.r.t. (X, D). Suppose $q_0, q_1 : V \to [0,1]$ satisfy (6-7).*

Let $\mathcal{P}$ be the user distribution constructed by Algorithm 1. Then $\mathcal{P}$ has pointwise mean μ and is conditionally L-correlated w.r.t. $(X, 3D_\mu)$ where

$$D_\mu(x,y) \triangleq D(x,y) \min\left(\frac{1}{\alpha}, \frac{3}{\mu(x)+\mu(y)}\right).$$

Remark. The theorem can be strengthened by replacing D_μ with the shortest-paths metric induced by D_μ.

Below we provide a proof sketch. The detailed proof is presented in Section B.

Proof Sketch As we noted above, the statement about the pointwise mean trivially follows from Equation (6) using induction on the tree. In what follows we focus on conditional L-correlation.

Fix leaves $x, y \in X$ and a subset $S \subset X$. Let z be the least common ancestor of x, y. Recall that in Algorithm 1 the bit $\pi(\cdot)$ at each node is a random mutation of that of its parent. We focus on the event $\mathcal{E}$ that no mutation happened on the $z \to x$ and $z \to y$ paths. Note that $\mathcal{E}$ implies $\pi(x) = \pi(y) = \pi(z)$. Therefore

$$\Pr[\pi(x) \neq \pi(y) \,|Z_S] \leq \Pr[\bar{\mathcal{E}} \,|Z_S],$$

where $\bar{\mathcal{E}}$ is the negation of $\mathcal{E}$. Intuitively, $\bar{\mathcal{E}}$ is a low-probability "failure event". The rest of the proof is concerned with showing that $\Pr[\bar{\mathcal{E}} \,|Z_S] \leq 3D_\mu(x,y)$.

First we handle the unconditional case. We claim that

$$\Pr[\bar{\mathcal{E}}] \leq D_\mu(x,y). \tag{9}$$

Note that Equation (9) immediately implies that $\mathcal{P}$ is L-correlated w.r.t. (X, D_μ). This claim is not very difficult to prove, essentially since the condition in Equation (7) is specifically engineered to satisfy the unconditional L-correlation property. We provide the proof in detail.

Let $w \in \operatorname{argmin}_{u \in P_{xy}} \mu(u)$, where P_{xy} is the $x \to y$ path. Let $(z = x_0, x_1, \ldots, x_n = x)$ be the $z \to x$ path. For each $i \geq 1$ by Equation (7) the probability of having a mutation at x_i is at most $D(x_i, x_{i-1})/\mu(w)$, so the probability of having a mutation on the $z \to x$ path is at most $D(x,z)/\mu(w)$. Likewise for the $z \to y$ path. So $\Pr[\bar{\mathcal{E}}] \leq D(x,y)/\mu(w) \leq D(x,y)/\alpha$.

It remains to prove that

$$\Pr[\bar{\mathcal{E}}] \leq D(x,y) \frac{3}{\mu(x)+\mu(y)}. \tag{10}$$

Indeed, by L-continuity it holds that

$$\mu(w) \geq \mu(x) - D(x,w),$$
$$\mu(w) \geq \mu(y) - D(y,w).$$

Since $D(x,y) = D(x,w) + D(y,w)$, it follows that

$$\mu(w) \geq \tfrac{\mu(x)+\mu(y)-D(x,y)}{2}. \tag{11}$$

Now, either the right-hand side of Equation (11) is at least $\frac{\mu(x)+\mu(y)}{3}$, or the right-hand side of Equation (10) is at least 1. In both cases Equation (10) holds. This completes the proof of the claim (9).

The conditional case is much more difficult. We handle it by showing that

$$\Pr[\bar{\mathcal{E}} \,|\, Z_S] \leq 3 \Pr[\bar{\mathcal{E}}]. \tag{12}$$

In fact, Equation (12) holds even if Equation (7) is replaced with a much weaker bound: $\max(q_0(u), q_1(u)) \leq \frac{1}{2}$ for each u.

The mathematically subtle proof of Equation (12) can be found in Section B. The crux in this proof is that event Z_S is more likely if document z is not relevant to the user:

$$\Pr[Z_S \,|\, z = 0] \geq \Pr[Z_S \,|\, z = 1].$$

■

4.2 Arbitrary Metric Spaces

We can extend Theorem 3.1 to arbitrary metric spaces using prior work on *metric embeddings*. Fix an N-point metric space (X,D) and a function $\mu : X \to [\alpha, \frac{1}{2}]$ that is L-continuous on (X,D). It is known (Bartal, 1996; Fakcharoenphol et al., 2004) that there exists a distribution $\mathcal{P}_{\text{tree}}$ over tree metric spaces $(X,\mathcal{T})$ such that $D(x,y) \leq \mathcal{T}(x,y)$ and

$$\mathbb{E}_{\mathcal{T}\sim\mathcal{P}_{\text{tree}}}[\mathcal{T}(x,y)] \leq c\, D(x,y) \quad \forall x,y \in X,$$

where $c = O(\log N)$.[6]

Our construction (Algorithm 2) is simple: first sample a tree metric space $(X,\mathcal{T})$ from $\mathcal{P}_{\text{tree}}$, then independently generate a user distribution $\mathcal{P}_{\mathcal{T}}$ for $(X,\mathcal{T})$ as per Algorithm 1.

Theorem 6 *The user distribution $\mathcal{P}$ produced by Algorithm 2 has pointwise mean μ and is conditionally L-correlated w.r.t. $(X, 3c\,D_\mu)$, where D_μ is given by*

$$D_\mu(x,y) = D(x,y) \min\left(\tfrac{1}{\alpha}, \tfrac{3}{\mu(x)+\mu(y)}\right).$$

6. This is the main result in Fakcharoenphol et al. (2004), which improves on an earlier result in Bartal (1996) with $c = O(\log^2 N)$. For point sets in a d-dimensional Euclidean space one could take $c = O(d \log \frac{1}{\varepsilon})$, where ε is the minimal distance. In fact, this result extends to a much more general family of metric spaces—those of doubling dimension d (Gupta et al., 2003). Doubling dimension, the smallest d such that any ball can be covered by 2^d balls of half the radius, has been introduced to the theoretical computer science literature in Gupta et al. (2003), and has been a well-studied concept since then.

Algorithm 2 User distribution for arbitrary metric spaces

Input: metric space (X,D); function $\mu : X \to [\alpha, \frac{1}{2}]$ that is L-continuous on (X,D).
Output: relevance vector $\pi : X \to \{0,1\}$

1. Sample a tree metric space $(X,\mathcal{T})$ from $\mathcal{P}_{\text{tree}}$,
2. Run Algorithm 1 for $(X,\mathcal{T})$, output the resulting π.

Proof The function μ is L-continuous w.r.t. each tree metric space $(X,\mathcal{T})$, so by Theorem 3.1 user distribution $\mathcal{P}_{\mathcal{T}}$ has pointwise mean μ and is conditionally L-correlated w.r.t. $(X, 3\mathcal{T}_\mu)$. It follows that the aggregate user distribution $\mathcal{P}$ has pointwise mean μ, and moreover for any $x,y \in X$ and $S \subset X$ we have

$$\begin{aligned}
&\Pr_{\pi\sim\mathcal{P}}[\pi(x) \neq \pi(y) \,|Z_S] \\
&\quad \leq \mathbb{E}_{\mathcal{T}\sim\mathcal{P}_{\text{tree}}}\left[\Pr_{\pi\sim\mathcal{P}_{\mathcal{T}}}[\pi(x) \neq \pi(y) \,|Z_S]\right] \\
&\quad \leq \mathbb{E}_{\mathcal{T}\sim\mathcal{P}_{\text{tree}}}\left[3\,\mathcal{T}_\mu(x,y)\right] \\
&\quad \leq 3c\,D_\mu(x,y).
\end{aligned}$$

■

4.3 Distributions over User Distributions

Let us verify that conditional L-continuity is *robust*, in the sense that any distribution over conditionally L-continuous user distributions is itself conditionally L-continuous. This result considerably extends the family of user distributions for which we have conditional L-continuity guarantees.

Lemma 7 *Let $\mathcal{P}$ be a distribution over countably many user distributions $\mathcal{P}_i$ that are conditionally L-continuous w.r.t. a metric space (X,D). Then $\mathcal{P}$ is conditionally L-continuous w.r.t. (X,D).*

Proof Let μ and μ_i be the (conditional) pointwise means of $\mathcal{P}$ and $\mathcal{P}_i$, respectively. Formally, let us treat each $\mathcal{P}_i$ as a measure, so that $\mathcal{P}_i(E)$ is the probability of event E under $\mathcal{P}_i$. Let $\mathcal{P} = \sum_i q_i \mathcal{P}_i$, where $\{q_i\}$ are positive coefficients that sum up to 1. Fix documents $x,y \in X$ and a subset $S \subset X$. Then

$$\begin{aligned}
\mu(x|S) = \mathcal{P}(x=1\,|Z_S) &= \frac{\mathcal{P}(x=1 \wedge Z_S)}{\mathcal{P}(Z_S)} \\
&= \frac{\sum_i q_i\, \mathcal{P}_i(x=1 \wedge Z_S)}{\mathcal{P}(Z_S)} \\
&= \frac{\sum_i q_i\, \mathcal{P}_i(Z_S)\, \mu_i(x|Z_S)}{\mathcal{P}(Z_S)}.
\end{aligned}$$

It follows that

$$\begin{aligned}
&|\mu(x|S) - \mu(y|S)| \\
&\quad = \frac{\sum_i q_i \, \mathcal{P}_i(Z_S) \, (\mu_i(x|Z_S) - \mu_i(y|Z_S))}{\mathcal{P}(Z_S)} \\
&\quad \leq \frac{\sum_i q_i \, \mathcal{P}_i(Z_S) \, D(x,y)}{\mathcal{P}(Z_S)} \\
&\quad \leq D(x,y).
\end{aligned}$$

■

5. Algorithms from Prior Work

Let us discuss some algorithmic ideas from prior work that can be adapted to our setting. Interestingly, one can combine these algorithms in a *modular* way, which we make particularly transparent by putting forward a suitable naming scheme. Throughout this section, we let `Bandit` be some algorithm for the MAB problem.

5.1 Ranked Bandits

Given some bandit algorithm `Bandit`, the "ranked" algorithm `RankBandit` for the multi-slot MAB problem is defined as follows (Radlinski et al., 2008). We have k slots (i.e., ranks) for which we wish to find the best documents to present. In each slot i, a separate instance $\mathcal{A}_i$ of `Bandit` is created. In each round these instances select the documents to show independently of one another. If a user clicks on slot i, then this slot receives a reward of 1, and all higher (i.e., skipped) slots $j < i$ receive a reward of 0. For slots $j > i$, the state is rolled back as if this round had never happened (as if the user never considered these documents). If no slot is clicked, then all slots receive a reward of 0.

Let us emphasize that the above approach can be applied to *any* algorithm `Bandit`. In Radlinski et al. (2008), this approach gives rise to algorithms `RankUCB1` and `RankEXP3`, based on MAB algorithms `UCB1` and `EXP3`(Auer et al., 2002a,b). `EXP3` is designed for the *adversarial* setting with no assumptions on how the clicks are generated, which translates into concrete provable guarantees for `RankEXP3`. `UCB1` is geared towards the *stochastic* setting with i.i.d. rewards on each arm, although the per-slot i.i.d. assumption breaks for slots $i > 1$ because of the influence of the higher slots. Nevertheless, in small-scale experiments `RankUCB1` performs much better than `RankEXP3`(Radlinski et al., 2008).

Provable guarantees. Letting T be the number of rounds and `OPT` be the probability of clicking on the optimal ranking, algorithm `RankBandit` achieves

$$\mathbb{E}[\texttt{\#clicks}] \geq (1 - \tfrac{1}{e})\, T \times \texttt{OPT} - k R(T), \tag{13}$$

where $R(T)$ is any upper bound on regret for `Bandit` in each slot (Radlinski et al., 2008; Streeter and Golovin, 2008).

In the multi-slot setting, *performance* of an algorithm up to time T is defined as the time-averaged expected total number of clicks. We will consider performance as a function of T. Assuming $R(T) = o(T)$ in Equation (13), performance of `RankBandit` converges to or exceeds $(1 - \frac{1}{e})$`OPT`.

Convergence to $(1-\frac{1}{e})\texttt{OPT}$ is proved to be worst-case optimal. Thus, as long as $R(T)$ scales well with time, for the document collection sizes that are typical for the application at hand, Radlinski et al. (2008) interpret Equation (13) as a proof of an algorithm's scalability in the multi-slot MAB setting.

`RankBandit` is presented in Radlinski et al. (2008) as the online version of the *greedy algorithm*: an offline fully informed algorithm that selects documents greedily slot by slot from top to bottom. The performance of this algorithm is called the *greedy optimum*,[7] which is equal to $(1-\frac{1}{e})\,\texttt{OPT}$ in the worst case, but for "benign" problem instances it can be as good as `OPT`. The greedy optimum is a more natural benchmark for `RankBandit` than $(1-\frac{1}{e})\texttt{OPT}$. However, results w.r.t. this benchmark are absent in the literature.[8]

5.2 Lipschitz Bandits

Both `UCB1` and `EXP3` are impractical when there are too many documents to explore them all. To alleviate this issue, one can use the similarity information provided by the metric space and the Lipschitz assumption; this setting is called *Lipschitz MAB*.

Below we describe two "metric-aware" algorithms from Kleinberg (2004) and Kleinberg et al. (2008b). Both are well-defined for arbitrary metric spaces, but for simplicity we present them for a special case in which documents are leaves in a *document tree* (denoted τ_{d}) with an ε-exponential tree metric. In both algorithms, a *subtree* is chosen in each round, then a document in this subtree is sampled at random, choosing uniformly at each branch.

Given some bandit algorithm `Bandit`, Kleinberg (2004) define algorithm `GridBandit` for the Lipschitz MAB setting. This algorithm proceeds in phases: in phase i, the depth-i subtrees are treated as "arms", and a fresh copy of `Bandit` is run on these arms.[9] Phase i lasts for $k\varepsilon^{-2i}$ rounds, where k is the number of depth-i subtrees. This meta-algorithm, coupled with an adversarial MAB algorithm such as `EXP3`, is the only algorithm in the literature that takes advantage of the metric space in the adversarial setting. Following Radlinski et al. (2008), we expect `GridEXP3` to be overly pessimistic for our problem, trumped by the corresponding stochastic MAB approaches such as `GridUCB1`.

The "zooming algorithm" (Kleinberg et al., 2008b, Algorithm 3) is a more efficient version of `GridUCB1`: instead of iteratively reducing the grid size in the entire metric space, it *adaptively* refines the grid in promising areas. It maintains a set $\mathcal{A}$ of *active subtrees* which collectively partition the leaf set. In each round the active subtree with the maximal *index* is chosen. The index of a subtree is (assuming stochastic rewards) the best available upper confidence bound on the click probabilities in this subtree. It is defined via the *confidence radius*[10] given (letting T be the time horizon) by

$$\texttt{rad}(\cdot) \triangleq \sqrt{4\log(T)/(1+\#\texttt{samples}(\cdot))}. \tag{14}$$

The algorithm "zooms in" on a given active subtree u (de-activates u and activates all its children) when $\texttt{rad}(u)$ becomes smaller than its *width* $\mathtt{W}(u) \triangleq \varepsilon^{\mathrm{depth}(u)} = \max_{x,x'\in u} D(x,x')$.

7. If due to ties there are multiple "greedy rankings", define the greedy optimum via the *worst* of them.
8. Following the conference publication of this paper, Streeter and Golovin claimed that the techniques in Streeter and Golovin (2008) can be used to extend Equation (13) to the greedy optimum benchmark. If so, then it may be possible to use the same approach to improve our guarantees.
9. As an empirical optimization, previous events can also be replayed to better initialize later phases.
10. The meaning of $\texttt{rad}(\cdot)$ is that w.h.p. the sample average is within $\pm\texttt{rad}(\cdot)$ from the true mean.

Algorithm 3 "Zooming algorithm" in trees

initialize (document tree τ_d)**:**
$\mathcal{A} \leftarrow \emptyset$; activate($\texttt{root}(\tau_d)$)

activate($u \in \texttt{nodes}(\tau_d)$)**:**
$\mathcal{A} \leftarrow \mathcal{A} \cup \{u\}$; $n(u) \leftarrow 0$; $r(u) \leftarrow 0$

Main loop:
$u \leftarrow \text{argmax}_{u \in \mathcal{A}}\, \texttt{index}(u)$,
where $\texttt{index}(u) = \frac{r(u)}{n(u)} + 2\,\texttt{rad}(u)$
"Play" a random document from $\texttt{subtree}(u)$
$r(u) \leftarrow r(u) + \{\text{reward}\}$; $n(u) \leftarrow n(u) + 1$
if $\texttt{rad}(u) < \texttt{W}(u)$ **then**
deactivate u: remove u from $\mathcal{A}$
activate all children of u

Provable guarantees. Regret guarantees for the two algorithms above are independent of the number of arms (which, in particular, can be infinite). Instead, they depend on the covering properties of the metric space (X,D). A crucial notion here is the *covering number* $N_r(X)$, defined as the minimal number of balls of radius r sufficient to cover X. It is often useful to summarize the covering numbers $N_r(X)$, $r > 0$ with a single number called the *covering dimension*:

$$\texttt{CovDim}(X,D) \triangleq \inf\{d \geq 0 : N_r(X) \leq \alpha r^{-d} \quad \forall r > 0\}. \tag{15}$$

(Here $\alpha > 0$ is a constant which we will keep implicit in the notation.) In particular, for an arbitrary point set in $\mathbb{R}^d$ under the standard (ℓ_2) distance, the covering dimension is d, for some $\alpha = O(1)$. For an ε-exponential tree metric with maximal branching factor b, the covering dimension is $d = \log_{1/\varepsilon}(b)$, with $\alpha = 1$.

Against an oblivious adversary, GridEXP3 has regret

$$R(T) = \tilde{O}(\alpha T^{(d+1)/(d+2)}), \tag{16}$$

where d is the covering dimension of (X,D).

For the stochastic setting, GridUCB1 and the zooming algorithm enjoy strong instance-dependent regret guarantees. These guarantees reduce to Equation (16) in the worst case, but are much better for "nice" problem instances. Informally, regret guarantees improve for problem instances in which the set of near-optimal arms has smaller covering numbers than the set of all arms. Regret guarantees for the zooming algorithm are (typically) much stronger than for GridUCB1. In particular, one can derive a version of Equation (16) with a different d called the *zooming dimension*, which is equal to the covering dimension in the worst case but can be much smaller, even $d = 0$. These issues are further discussed in Appendix C.

5.3 Anytime Guarantees and the Doubling Trick

While the zooming algorithm, and also the contextual zooming algorithm from Section 5.5, are defined for a fixed time horizon, one can obtain the corresponding *anytime* versions using a simple *doubling trick*: in each phase $i \in \mathbb{N}$, run a fresh instance of the algorithm for 2^i rounds. These

versions are run indefinitely and enjoy the same asymptotic upper bounds on regret as the original algorithms (but now these bounds hold for each round).

5.4 Ranked Bandits in Metric Spaces

Using and combining the algorithms in the previous two subsections, we obtain the following battery of algorithms for k-slot Lipschitz MAB problem:

- metric-oblivious algorithms: `RankUCB1` and `RankEXP3`.
- simple metric-aware algorithms: `RankGridUCB1` and `RankGridEXP3` (ranked versions of `GridUCB1` and `GridEXP3`, respectively).
- `RankZoom`: the ranked version of the zooming algorithm.

In theory, `RankGridEXP3` scales to large document collections, in the sense that it achieves Equation (13) with $R(T)$ that does not degenerate with #documents:

Theorem 8 *Consider the k-slot Lipschitz MAB problem on a metric space with covering dimension d (as defined in Equation (15), with constant α). Then after T rounds* `RankGridEXP3` *achieves*

$$\frac{\mathbb{E}[\texttt{\#clicks}]}{T} \geq (1-\tfrac{1}{e})\,\texttt{OPT} - \tilde{O}\left(\frac{\alpha k}{T^{1/(d+2)}}\right).$$

The theorem follows from the respective regret bounds for `GridEXP3` (Equation (16)) and `RankBandit` (Equation (13)). We do not have any provable guarantees for other algorithms because the corresponding regret bounds for the single-slot setting do not directly plug into Equation (13). However, the strong instance-dependent guarantees for `GridUCB1` and especially for the zooming algorithm (even though they do not directly apply to the ranked bandit setting) suggest that `RankGridUCB1` and `RankZoom` are promising. We shall see that these two algorithms perform much better than `RankGridEXP3` in the experiments.

5.5 Contextual Lipschitz Bandits

We also leverage prior work on contextual bandits. The relevant contextual MAB setting, called contextual Lipschitz MAB, is as follows. In each round nature reveals a *context* h, an algorithm chooses a document x, and the resulting reward is an independent $\{0,1\}$ sample with expectation $\mu(x|h)$. Further, one is given *similarity information*: metrics D and D_{c} on documents and contexts, respectively, such that for any two documents x,x' and any two contexts h,h' we have

$$|\mu(x|h)-\mu(x'|h')| \leq D(x,x')+D_{\mathrm{c}}(h,h').$$

Let X_{c} be the set of contexts, and $X_{\mathrm{dc}} = X \times X_{\mathrm{c}}$ be the set of all (document, context) pairs. Abstractly, one considers the metric space $(X_{\mathrm{dc}}, D_{\mathrm{dc}})$, henceforth the *DC-space*, where the metric is

$$D_{\mathrm{dc}}((x,h),(x',h')) = D(x,x')+D_{\mathrm{c}}(h,h').$$

We will use the "contextual zooming algorithm" (`ContextZoom`) from Slivkins (2009). This algorithm is well-defined for arbitrary D_{dc}, but for simplicity we will state it for the case when D and D_{c} are ε-exponential tree metrics.

Let us assume that documents and contexts are leaves in a document tree τ_{d} and context tree τ_{c}, respectively. The algorithm (see Algorithm 4 for pseudocode) maintains a set $\mathcal{A}$ of *active strategies*

Algorithm 4 `ContextZoom` in trees

initialize (document tree τ_d, context tree τ_c)**:**
$\mathcal{A} \leftarrow \emptyset$; activate($\texttt{root}(\tau_d)$, $\texttt{root}(\tau_c)$)

activate ($u \in \texttt{nodes}(\tau_d)$, $u_c \in \texttt{nodes}(\tau_c)$)**:**
$\mathcal{A} \leftarrow \mathcal{A} \cup \{(u,u_c)\}$; $n(u,u_c) \leftarrow 0$; $r(u,u_c) \leftarrow 0$

Main loop:
Input a context $h \in \texttt{nodes}(\tau_c)$
$(u,u_c) \leftarrow \operatorname{argmax}_{(u,u_c)\in\mathcal{A}:\, h\in u_c} \texttt{index}(u,u_c)$,
where $\texttt{index}(u,u_c) = \texttt{W}(u\times u_c) + \frac{r(u,u_c)}{n(u,u_c)} + \texttt{rad}(u,u_c)$
"Play" a random document from $\texttt{subtree}(u)$
$r(u,u_c) \leftarrow r(u,u_c) + \{\text{reward}\}$; $n(u,u_c) \leftarrow n(u,u_c)+1$
if $\texttt{rad}(u,u_c) < \texttt{W}(u,u_c)$ **then**
deactivate (u,u_c): remove (u,u_c) from $\mathcal{A}$
activate all pairs (child(u), child(u_c))

of the form (u,u_c), where u is a subtree in τ_d and u_c is a subtree in τ_c. At any given time the active strategies partition X_{dc}. In each round, a context h arrives, and one of the active strategies (u,u_c) with $h \in u_c$ is chosen: namely the one with the maximal *index*, and then a document $x \in u$ is picked uniformly at random. The index of (u,u_c) is, essentially, the best available upper confidence bound on expected rewards from choosing a document $x \in u$ given a context $h \in u_c$. The index is defined via sample average, confidence radius (14), and "width" $\texttt{W}(u\times u_c)$. The latter can be any upper bound on the diameter of the product set $u \times u_c$ in the DC-space:

$$\texttt{W}(u,u_c) \geq \max_{x,x'\in u,\ h,h'\in u_c} D(x,x') + D_c(h,h'). \tag{17}$$

The (de)activation rule ensures that the active strategies form a finer partition in the regions of the DC-space that correspond to higher rewards and more frequently occurring contexts.

Provable guarantees. The provable guarantees for the contextual MAB problem are in terms of *contextual regret*, which is regret is with respect to a much stronger benchmark: the best arm in hindsight *for every given context*.

Regret guarantees for `ContextZoom` focus on the *DC-space* (X_{dc}, D_{dc}). A very pessimistic regret bound is Equation (16) with $d = \texttt{CovDim}(X_{dc}, D_{dc})$. However, as for the zooming algorithm, much better instance-dependent bounds are possible. See Appendix C for further discussion.

6. New Approach: Ranked Contextual Bandits

We now present a new approach in which the upper slot selections are taken into account as a *context* in the contextual MAB setting.

The slot algorithms in the `RankBandit` setting can make their selections sequentially. Then without loss of generality each slot algorithm $\mathcal{A}_i$ knows the set S of documents in the upper slots. We propose to treat S as a "context" to $\mathcal{A}_i$. Specifically, $\mathcal{A}_i$ will assume that none of the documents in S is clicked, that is, event Z_S happens (else the i-th slot is ignored by the user). For each such round, the click probabilities for $\mathcal{A}_i$ are given by $\mu(\cdot\,|Z_S)$, which is an L-continuous function on (X,D).

6.1 `RankCorrZoom`: "Light-weight" Ranked Contextual Algorithm

We first propose a simple modification to `RankZoom`, called `RankCorrZoom`, which uses the contexts as discussed above.

Recall that in the zooming algorithm, the index of an active subtree u is defined so that, assuming stochastic rewards, it is an upper confidence bound on the click probability of any document x in this subtree:

$$\texttt{w.h.p. } \texttt{index}(u) \geq \max_{x \in u} \mu(x). \tag{18}$$

Moreover, it follows from the analysis in (Kleinberg et al., 2008b) that performance of the algorithm improves if the index is decreased as long as Equation (18) holds.

Now consider `RankZoom`, and let $\mathcal{A}_i$ be the instance of the zooming algorithm in slot $i \geq 2$. While for $\mathcal{A}_i$ the rewards are no longer stochastic, our intuition for why `RankZoom` may be a good algorithm is still based on Equation (18). In other words, we *wish* that for each context $S \subset X$ we have

$$\texttt{w.h.p. } \texttt{index}(u) \geq \max_{x \in u} \mu(x|Z_S), \tag{19}$$

and our intuition is that it is desirable to decrease the index as long as Equation (19) holds.

We will derive an upper bound on $\max_{x \in u} \mu(x|Z_S)$ using correlation between u and S, and we will cap the index of u at this quantity. Since $\mu(y|Z_S) = 0$ for any $y \in S$, we have

$$\begin{aligned} &\mu(x|Z_S) = |\mu(x|Z_S) - \mu(y|Z_S)| \leq D(x,y), \quad \forall y \in S \\ &\mu(x|Z_S) \leq D(x,S) \triangleq \min_{y \in S} D(x,y). \end{aligned} \tag{20}$$

In other words, if document x is close to some document in S, the event Z_S limits the conditional probability $\mu(x|Z_S)$. Therefore we can cap the index of u at $\max_{x \in u} D(x,S)$:

$$\texttt{index}(u) \leftarrow \min\left(\texttt{index}(u),\ \max_{x \in u} D(x,S)\right).$$

The version of `RankZoom` with the above "correlation rule" will be called `RankCorrZoom`.

To simplify the computation of $\max_{x \in u} D(x,S)$ in an ε-exponential tree metric, we note that it is equal to $D(\texttt{root}(u), S)$ if u is disjoint with S, and in general it is equal to $D(\texttt{root}(v), S)$, where v is the largest subtree of u that is disjoint with S.

6.2 Contextual *Lipschitz* MAB Interpretation

Let us cast each slot algorithm $\mathcal{A}_i$ as a contextual algorithm in the contextual *Lipschitz* MAB setting (as defined in Section 5.5). We need to specify a metric D_c on contexts $S \subset X$ which can be computed by the algorithm and satisfies the Lipschitz condition:

$$|\mu(x|Z_S) - \mu(x|Z_{S'})| \leq D_c(S,S') \quad \text{for all } x \in X \text{ and } S, S' \subset X. \tag{21}$$

Lemma 9 *Consider the k-slot Lipschitz MAB problem. For any $S, S' \subset X$, define*

$$D_c(S,S') \triangleq 4 \inf \textstyle\sum_{j=1}^{n} D(x_j, x'_j), \tag{22}$$

where the infimum is taken over all $n \in \mathbb{N}$ and over all n-element sequences $\{x_j\}$ and $\{x'_j\}$ that enumerate, possibly with repetitions, all documents in S and S'. Then D_c satisfies Equation (21).

Proof For shorthand, let us write

$$\sigma(x|S) \triangleq 1-\mu(x|Z_S),$$
$$\sigma(x|S,y) \triangleq \sigma(x|S\cup\{y\}).$$

First, we claim that for any $y \in X$ and $y' \in S$

$$|\sigma(x|S,y)-\sigma(x|S,y')| \leq 4D(y,y'). \tag{23}$$

Indeed, noting that $\sigma(x|S,y) = \sigma(y|S,x)\ \frac{\sigma(x|S)}{\sigma(y|S)}$, we can re-write the left-hand side of Equation (23) as

$$\begin{aligned}\text{LHS}(23) &= \sigma(x,S)\left|\frac{\sigma(y|S,x)}{\sigma(y|S)}-\frac{\sigma(y'|S,x)}{\sigma(y'|S)}\right| \\ &\leq \sigma(x,S)\ D(y,y')\ \frac{\sigma(y|S)+\sigma(y|S,x)}{\sigma(y|S)\,\sigma(y'|S)} \\ &= D(y,y')\ \frac{\sigma(x|S)+\sigma(x|S,y)}{\sigma(y'|S)} \leq 2D(y,y').\end{aligned} \tag{24}$$

In Equation (24), we have used the L-continuity of $\sigma(\cdot|S)$ and $\sigma(\cdot|S,x)$. To achieve the constant of 2, it was crucial that $y' \in S$, so that $\sigma(y'|S) = 1$. This completes the proof of Equation (23).

Fix some $n \in \mathbb{N}$ and some n-element sequences $\{x_i\}$ and $\{x'_i\}$ that enumerate, possibly with repetitions, all values in S and S', respectively. Consider sets

$$S_i = \{x'_1, \ldots, x'_i\} \cup \{x_{i+1}, \ldots, x_n\},\ \ 1 \leq i \leq n-1,$$

and let $S_0 = S$ and $S_{n+1} = S'$. To prove the lemma, it suffices to show that

$$|\sigma(x|S_i)-\sigma(x|S_{i+1})| \leq 4D(x_{i+1}, x'_{i+1}) \tag{25}$$

for each $i \leq n$. To prove Equation (25), fix i and let $y = x_{i+1}$ and $y' = x'_{i+1}$. Note that $S_i \cup \{y'\} = S_{i+1} \cup \{y\}$, call this set S^*. Then using Equation (23) (note, $y \in S_i$ and $y' \in S'_i$) we obtain

$$\begin{aligned}|\sigma(x|S_i)-\sigma(x|S^*)| &= |\sigma(x|S_i,y)-\sigma(x|S_i,y')| \\ &\leq 2D(y,y'), \\ |\sigma(x|S_{i+1})-\sigma(x|S^*)| &= |\sigma(x|S_{i+1},y')-\sigma(x|S_{i+1},y)| \\ &\leq 2D(y,y'),\end{aligned}$$

which implies Equation (25). ■

6.3 `RankContextZoom`: "Full-blown" Ranked Contextual Algorithm

Now we can take any algorithm for the contextual Lipschitz MAB problem (with metric D_c on contexts given by Equation (22)), and use it as a slot algorithm. We will use `ContextZoom`, augmented by the "correlation rule" similar to the one in Section 6.1. The resulting "ranked" algorithm will be called `RankContextZoom`.

The implementation details are not difficult. Suppose the metric space on documents is the ε-exponential tree metric, and let τ_d be the document tree. Consider slot $(i+1)$-th slot, $i \geq 1$.[11] Then the contexts are unordered i-tuples of documents. Let us define *context tree* τ_c as follows. Depth-ℓ nodes of τ_c are unordered i-tuples of depth-ℓ nodes from τ_d, and leaves are contexts. The root of τ_c is $(r \dots r)$, where $r = \texttt{root}(\tau_d)$. For each internal node $u_c = (u_1 \dots u_i)$ of τ_c, its children are all unordered tuples $(v_1 \dots v_i)$ such that each v_j is a child of u_j in τ_d. This completes the definition of τ_c. Letting u and u_c be level-ℓ subtrees of τ_d and τ_c, respectively, it follows from the definition of D_c in Equation (22) that $D_c(S,S') \leq 4i\varepsilon^\ell$ for any contexts $S, S' \in u_c$. Thus setting $\mathbb{W}(u \times u_c) \triangleq \varepsilon^\ell(4i+1)$ satisfies Equation (17).

We define the "correlation rule" as follows. Let (u, u_c) be an active strategy in the execution of `ContextZoom`, where u is a subtree of the document tree τ_d, and u_c is a subtree of the context tree τ_c. It follows from the analysis in (Slivkins, 2009) that decreasing the index of (u, u_c) improves performance, as long it holds that

$$\texttt{index}(u, u_c) \geq \mu(x|Z_S), \quad \forall x \in u, S \in u_c.$$

Recall that $\mu(x|Z_S) \leq D(x,S)$ by Equation (20), so we can cap $\texttt{index}(u, u_c)$ at $\max_{x \in u} D(x|S)$:

$$\texttt{index}(u,S) \leftarrow \min\left(\texttt{index}(u,S),\ \max_{x \in u} D(x|S)\right).$$

This completes the description of `RankContextZoom`.

7. Provable Scalability Guarantees and Discussion

Noting that for each slot $i \geq k$ the covering dimension of the DC-space is at most k times the covering dimension of (X,D), it follows that a (very pessimistic) upper bound on contextual regret of `RankContextZoom` is $R(T) = \tilde{O}(\alpha T^{1-1/(kd+2)})$. Plugging this into Equation (13), we obtain:

Theorem 10 *Consider the k-slot Lipschitz MAB problem on a metric space with covering dimension d (as defined in Equation (15), with constant α). Then after T rounds algorithm* `RankContextZoom` *achieves*

$$\frac{\mathbb{E}[\texttt{\#clicks}]}{T} \geq (1 - \tfrac{1}{e})\,\texttt{OPT} - \tilde{O}\left(\frac{\alpha k}{T^{1/(kd+2)}}\right).$$

This is just a basic scalability guarantee which does not degenerate with the number of documents. (Note that it is *worse* than the one for `RankGridEXP3`.) We believe that this guarantee is very pessimistic, as it builds on a very pessimistic version of the result for `ContextZoom`. In particular, we ignore the intuition that for a given slot, contexts $S \subset X$ may gradually converge over time to the greedy optimum, which effectively results in a much smaller set of possible contexts.[12] We believe this effect is very important to the performance `RankContextZoom`. In particular, it causes `RankContextZoom` to perform much better than `RankGridEXP3` in simulations.

11. For slot 1, contexts are empty, so `ContextZoom` reduces to Algorithm 3.
12. It is also wasteful (but perhaps less so) that we use a slot-k bound for each slot $i < k$.

7.1 A Better Benchmark

Recall that while the bound in Equation (13) uses $(1-\frac{1}{e})\,\mathrm{OPT}$ as a benchmark, a more natural benchmark would be the greedy optimum. We provide a preliminary convergence result for RankContextZoom, without any specific regret bounds.

Such result is more elegantly formulated in terms of a version of RankContextZoom, henceforth called anytime-RankContextZoom, which uses the anytime version of ContextZoom (see Section 5.3).

Theorem 11 *Fix an instance of the k-slot MAB problem. The performance of anytime-*RankContextZoom *up to any given time t is equal to the greedy optimum minus $f(t)$ such that $f(t) \to 0$.*

Proof Sketch It suffices to prove that with high probability, anytime-RankContextZoom outputs a greedy ranking in all but $f_k(t)$ rounds among the first t rounds, where $f_k(t) \to 0$.

We prove this claim by induction on k, the number of slots. Suppose it holds for some $k-1$ slots, and focus on the k-th slot. Consider all rounds in which a greedy ranking is chosen for the upper slots but not for the k-th slot. In each such round, the k-th slot replica of anytime-ContextZoom incurs contextual regret at least δ_k, for some instance-specific constant $\delta_k > 0$. Thus, with high probability there can be at most $R_k(t)/\delta_k$ such rounds, where $R_k(t) = o(t)$ is an upper bound on contextual regret for slot k. Thus, one can take $f_k(t) = f_{k-1}(t) + R_k(t)/\delta_k$. ■

Theorem 11 is about the "metric-less" setting from Radlinski et al. (2008). It easily extends to the "ranked" version of any bandit algorithm whose contextual regret is sublinear with high probability.

It is an open question whether (and under which assumptions) Theorem 11 can be extended to the "ranked" versions of non-contextual bandit algorithms such as RankUCB1. One assumption that appears essential is the uniqueness of the greedy ranking. To see that multiple greedy rankings may cause problems for ranked non-contextual algorithms, consider a simple example:

- There are two slots and three documents x_1, x_2, x_3 such that $\mu = (\frac{1}{2}, \frac{1}{2}, \frac{1}{3})$ and the relevance of each arm is independent of that of the other arms.[13]

An optimal ranking for this example is a greedy ranking that puts x_1 and x_2 in the two slots, achieving aggregate click probability $\frac{3}{4}$. According to our intuition, a "reasonable" ranked non-contextual algorithm will behave as follows. The slot 1 algorithm will alternate between x_1 and x_2, each with frequency $\to \frac{1}{2}$. Since the slot-2 algorithm is oblivious to the slot 1 selection, it will observe averages that converge over time to $(\frac{1}{4}, \frac{1}{4}, \frac{1}{3})$,[14] so it will select document x_3 with frequency $\to 1$. Therefore frequency $\to 1$ the ranked algorithm will alternate between (x,z) or (y,z), each of which has aggregate click probability $\frac{2}{3}$.

13. Here documents x_1, x_2, x_3 can stand for disjoint *subsets* of documents with highly correlated payoffs. Documents within a given subset can lie far from one another in the metric space.

14. Suppose x_j, $j \in \{1,2\}$ is chosen in slot 1. Then, letting $S = \{x_j\}$, $\mu(x_1|Z_S)$ equals 0 if $j = 1$ and $\frac{1}{2}$ otherwise (which averages to $\frac{1}{4}$), whereas $\mu(x_3|Z_S) = \frac{1}{3}$.

RankUCB1 RankEXP3	metric-oblivious algorithms: ranked versions of UCB1 and EXP3	Section 5.1
RankGridUCB1 RankGridEXP3	simple metric-aware algorithms: ranked versions of GridUCB1 and GridEXP3	Section 5.4
RankZoom	the ranked version of the zooming algorithm	Section 5.4
	contextual algorithms:	
RankCorrZoom	"light-weight"(based on the zooming algorithm)	Section 6.1
RankContextZoom	"full-blown" (based on ContextZoom).	Section 6.3

Table 1: Algorithms for the k-slot Lipschitz MAB problem.

7.2 Desiderata

We believe that the above guarantees do not reflect the full power of our algorithms, and more generally the full power of conditional L-continuity. The "ideal" performance guarantee for RankBandit in our setting would use the greedy optimum as a benchmark, and would have a bound on regret that is free from the inefficiencies outlined in the discussion after Theorem 10. Furthermore, this guarantee would only rely on some general property of Bandit such as a bound on regret or contextual regret. We conjecture that such guarantee is possible for RankContextZoom, and, perhaps under some assumptions, also for RankCorrZoom and RankZoom.

Further, one would like to study the relative benefits of the new "contextual" algorithms (RankContextZoom and RankCorrZoom) and the prior work such as RankZoom. The discussion Section 7.1 suggests that the difference can be particularly pronounced when the pointwise mean has multiple peaks of similar value. In fact, we confirm this experimentally in Section 8.4.

8. Evaluation

Let us evaluate the performance of the algorithms presented in Section 5 and Section 6. We summarize these algorithms in Table 8.

In all UCB1-based algorithms in Table 8, including all extensions of the zooming algorithm, one can damp exploration by replacing the $4\log(T)$ factor in Equation (14) with 1. Such change effectively makes the algorithm more *optimistic*; it was found beneficial for RankUCB1 by Radlinski et al. (2008). We find (see Section 8.3) that this change greatly improves the average performance in our experiments. So, by a slight abuse of notation, we will assume this change from now on.

8.1 Experimental Setup

Using the generative model from Section 4 (Algorithm 1 with Equation (8)), we created a document collection with $|X| = 2^{15} \approx 32,000$ documents[15] in a binary ε-exponential tree metric space with $\varepsilon = 0.837$ (and constant $c = 1$, see Section 3.1). The value for ε was chosen so that the most dissimilar documents in the collection still have a non-trivial similarity, as may be expected for web documents. Each document's expected relevance $\mu(x)$ was set by first identifying a small number

15. This is a realistic number of documents that may be considered in detail for a typical web search query after pruning very unlikely documents.

of "peaks" $y_i \in X$, choosing $\mu(\cdot)$ for these documents, and then defining the relevance of other documents as the minimum allowed while obeying L-continuity and a background relevance rate μ_0:

$$\mu(x) \triangleq \max(\mu_0,\ \tfrac{1}{2} - \min_i D(x, y_i)). \tag{26}$$

For internal nodes in the tree, μ is defined bottom-up (from leaves to the root) as the mean value of all children nodes. As a result, we obtain a set of documents X where each document $x \in X$ has an expected click probability $\mu(x)$ that obeys L-continuity.

Our simulation was run over a 5-slot ranked bandit setting, learning the best 5 documents. We evaluated over 300,000 user visits sampled from $\mathcal{P}$ per Algorithm 1. Performance within 50,000 impressions, typical for the number of times relatively frequent queries are seen by commercial search engines in a month, is essential for any practical applicability of this approach. However, we also measure performance for a longer time period to obtain a deeper understanding of the convergence properties of the algorithms.

We consider two models for $\mu(\cdot)$ in Equation (26). In the first model, two "peaks" $\{y_1, y_2\}$ are selected at random with $\mu(\cdot) = \frac{1}{2}$, and μ_0 set to 0.05. The second model is less "rigid" (and thus more realistic): the relevant documents y_i and their expected relevance rates $\mu(\cdot)$ are selected according to a Chinese Restaurant Process (Aldous, 1985) with parameters $n = 20$ and $\theta = 2$, and setting $\mu_0 = 0.01$. The Chinese Restaurant Process is inspired by customers coming in to a restaurant with an infinite number of tables, each with infinite capacity. At time t, a customer arrives and can choose to sit at a new table with probability $\theta/(t-1+\theta)$, and otherwise sits at an already occupied table with probability proportional to the number of customers already sitting at that table. By considering each table as equivalent to a peak in the distrubtion, this leads to a set of peaks with expected relevance rates distributed accoring to a power law. Following Radlinski et al. (2008), we assign users to one of the peaks, then select relevant documents so as to obey the expected relevance rate $\mu(x)$ for each document x.

As baselines we use an algorithm ranking the documents at random, and the (offline) greedy algorithm discussed in Section 5.1.

8.2 Main Experimental Results

Our experimental results are summarized in Figure 1 and Figure 2.

`RankEXP3` and `RankUCB1` perform as poorly as picking documents randomly: the three curves are indistinguishable. This is due to the large number of available documents and slow convergence rates of these algorithms. Other algorithms that explore all strategies (such as REC Radlinski et al., 2008) would perform just as poorly. This result is consistent with results reported by Radlinski et al. (2008) on just 50 documents. On the other hand, algorithms that progressively refine the space of strategies explored perform much better.

`RankCorrZoom` achieves the best empirical performance, converging rapidly to near-optimal rankings. `RankZoom` is a close second. The theoretically preferred `RankContextZoom` comes third, with a significant gap. This appears to be due to the much larger branching factor in the strategies activated by `RankContextZoom` slowing down the convergence. (However, as we investigate in Section 8.4, `RankContextZoom` may significantly outperform the other algorithms if μ has multiple peaks with similar values.)

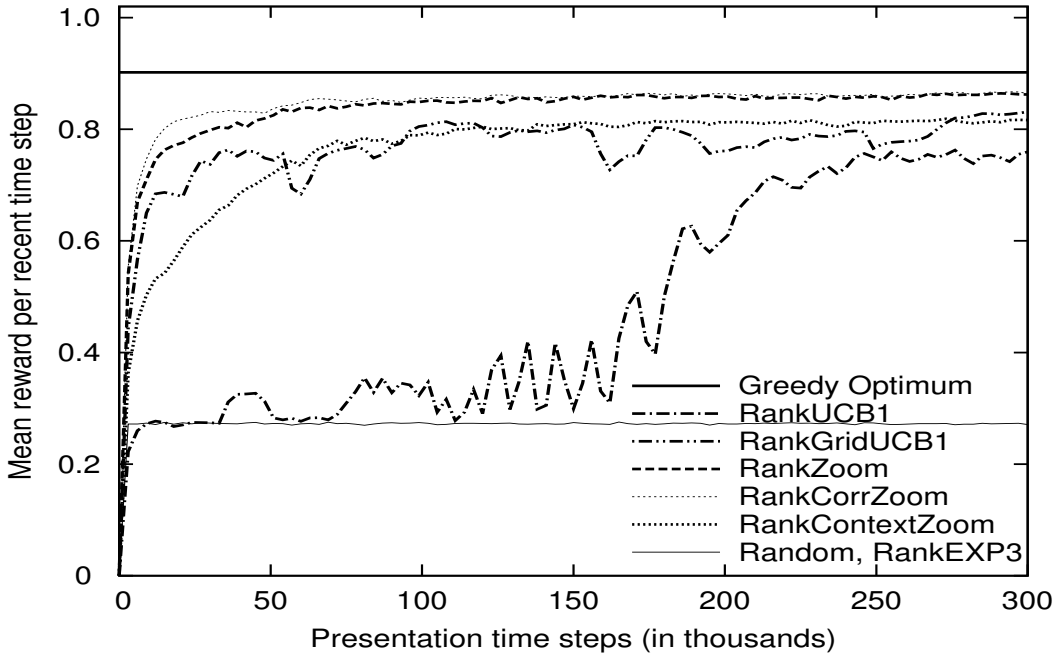

Figure 1: The learning algorithms on 5-slot problem instances with two relevance peaks.

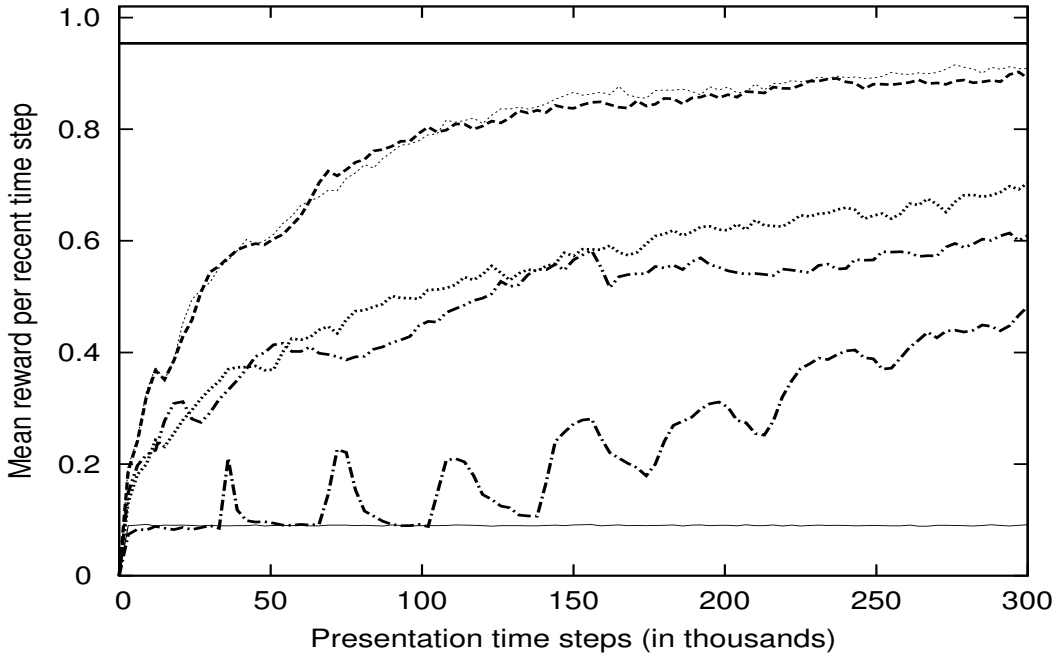

Figure 2: The learning algorithms on 5-slot problem instances with random relevance rates $\mu(\cdot)$ selected according to the Chinese Restaurant Process.

8.3 "Optimistic" vs. "Pessimistic" UCB1-style Algorithms

We find that the "optimistic" UCB1-style algorithms (obtained by replacing the $4\log(T)$ factor in Equation (14) with 1) perform dramatically better than their "pessimistic" counterparts. In Figure 3 and Figure 4 we compare RankUCB1 and RankZoom with their respective "pessimistic" versions (which are marked with a "- -" after the algorithm name). We saw a similar increase in performance for other UCB1-style algorithms, too.

8.4 Secondary Experiment

As discussed in Section 7.1, some RankBandit-style algorithms may converge to a suboptimal ranking if μ has multiple peaks with similar values. To investigate this, we designed a small-scale experiment presented in Figure 5. We generated a small collection of 128 documents using the same setup with two "peaks", and assumed 2 slots. Each peak corresponds to a half of the user population, with peak value $\mu = \frac{1}{2}$ and background value $\mu_0 = 0.05$.

We see that RankContextZoom converges more slowly than the other zooming variants, but eventually outperforms them. This confirms our intuition, and suggests that RankContextZoom may eventually outperform the other algorithms on a larger collection, such as that used for Figures 1 and 2.

9. Further Directions

This paper initiates the study of bandit learning-to-rank with side information on similarity between documents, focusing on an idealized model of document similarity based on the new notion of "conditional Lipschitz-continuity". As discussed in Section 7, we conjecture that provable performance guarantees can be improved significantly. On the experimental side, future work will include evaluating the model on web search data, and designing sufficiently memory- and time-efficient implementations to allow experiments on real users. An interesting challenge in such an endeavor would be to come up with effective similarity measures. A natural next step would be to also exploit the similarity between search queries.

Appendix A. Proof of Lemma 4 (Extending μ from Leaves to Tree Nodes)

Recall that Lemma 4 is needed to define the generative model in Section 4. We will prove a slightly more general statement:

Lemma 12 *Let D be the shortest-paths metric of an edge-weighted rooted tree with node set V and leaf set X. Let $\mu : X \to [a,b]$ be an L-continuous function on (X,D). Then μ can be extended to V so that $\mu : V \to [a,b]$ is L-continuous w.r.t. (V,D).*

Proof For each $x \in V$, let $\mathcal{L}(x)$ be the set of all leaves in the subtree rooted at x. For each $z \in \mathcal{L}(y)$ the assignment $\mu(x)$ should satisfy

$$\mu(z) - D(x,z) \leq \mu(x) \leq \mu(z) + D(x,z)$$

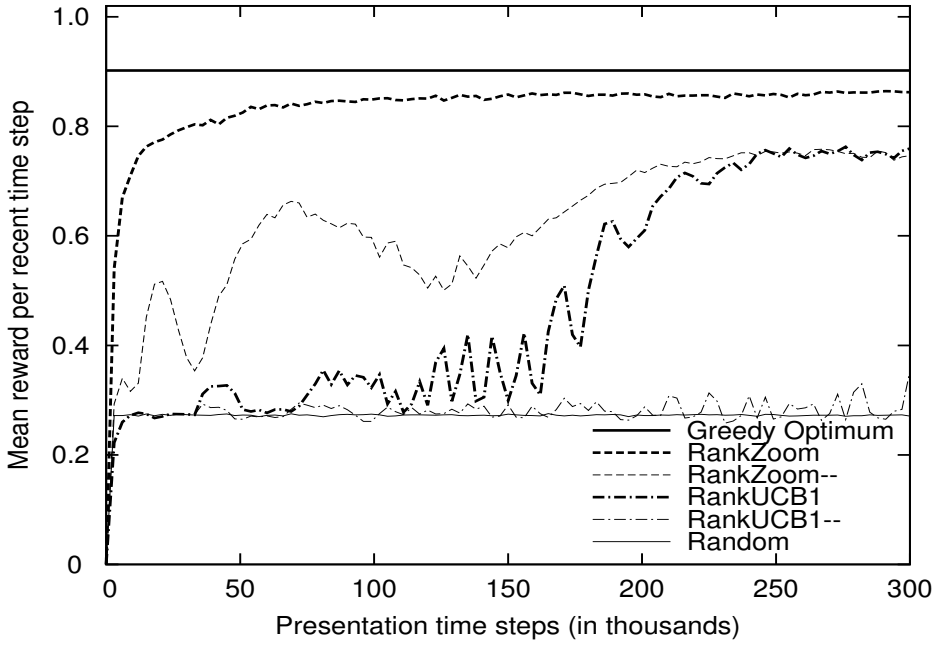

Figure 3: "Optimistic" vs. "pessimistic" UCB1-style algorithms:
The learning algorithms on 5-slot problem instances with two relevance peaks.

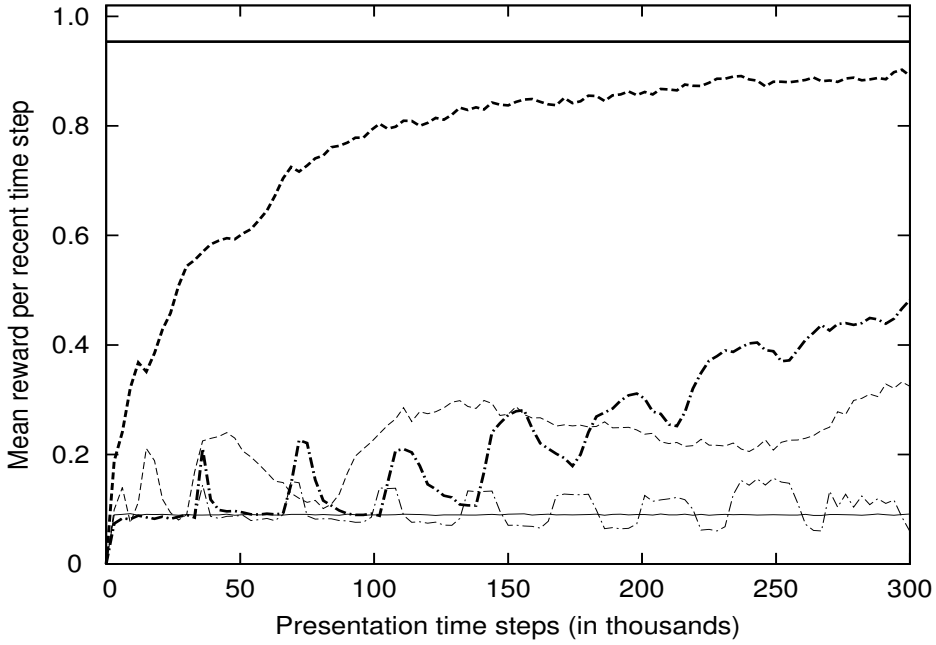

Figure 4: "Optimistic" vs. "pessimistic" UCB1-style algorithms:
The learning algorithms on 5-slot problem instances with random relevance rates $\mu(\cdot)$ selected according to the Chinese Restaurant Process.

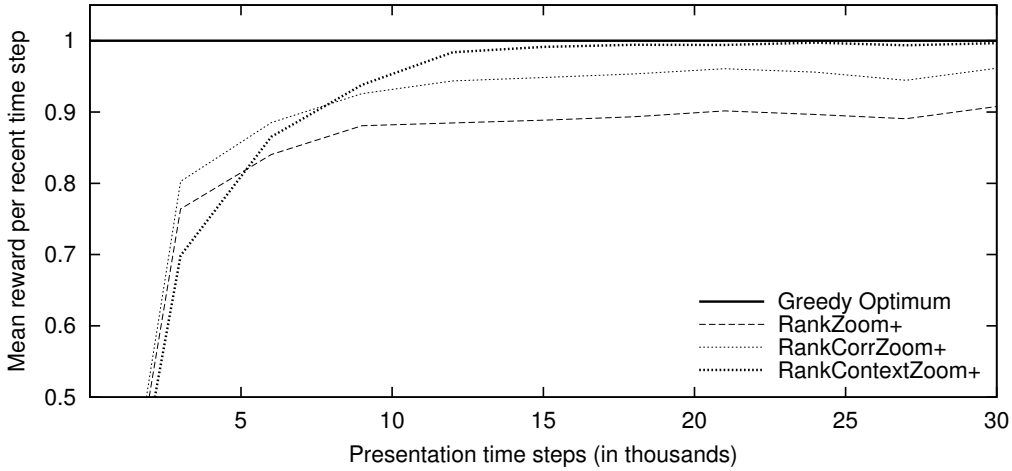

Figure 5: Zooming-style algorithms in a two-slot setting over a small document collection.

Thus $\mu(x)$ should lie in the interval $I(x) \triangleq [\mu^-(x), \mu^+(x)]$, where

$$\mu^-(x) \triangleq \sup_{z \in \mathcal{L}(x)} \mu(z) - D(x,z),$$
$$\mu^+(x) \triangleq \inf_{z \in \mathcal{L}(x)} \mu(z) + D(x,z).$$

This interval is always well-defined, that is, $\mu^-(x) \leq \mu^+(x)$. Indeed, if not then for some $z, z' \in \mathcal{L}(x)$

$$\mu(z) - D(x,z) > \mu(z') + D(x,z')$$
$$\mu(z) - \mu(z') > D(x,z) + D(x,z') \geq D(z,z'),$$

contradiction, claim proved. Note that $\mu^+(x) \geq a$ and $\mu^-(x) \leq b$, so the intervals $I(x)$ and $[a,b]$ overlap.

Using induction on the tree, we will construct values $\mu(x)$, $x \in V$ such that the Lipschitz condition

$$|\mu(x) - \mu(y)| \leq D(x,y) \qquad \text{for all } x, y \in X$$

holds whenever x is a parent of y. For the root x_0, let $\mu(x_0)$ be an arbitrary value in the interval $I(x_0) \cap [a,b]$. For the induction step, suppose for some x we have chosen $\mu(x) \in I(x) \cap [a,b]$ and y is a child of x. We need to choose $\mu(y) \in I(y) \cap [a,b]$ so that $|\mu(x) - \mu(y)| \leq D(x,y)$. Note that

$$\begin{aligned}
\mu(x) \geq \mu^-(x) &\geq \sup_{z \in \mathcal{L}(y)} \left[\mu(z) - D(x,y) - D(y,z)\right] \\
&= \mu^-(y) - D(x,y), \\
\mu(x) \leq \mu^+(x) &\leq \inf_{z \in \mathcal{L}(y)} \left[\mu(z) + D(x,y) + D(y,z)\right] \\
&= \mu^+(y) + D(x,y).
\end{aligned}$$

It follows that $I(y)$ and $[\mu(x) - D(x,y), \mu(x) + D(x,y)]$ have a non-empty intersection. Therefore, both intervals have a non-empty intersection with $[a,b]$. So we can choose $\mu(y)$ as required. This completes the construction of $\mu()$ on V.

To check that μ is Lipschitz-continuous on V, fix $x,y \in V$, let P be the $x \to y$ path in the tree, and note that

$$\begin{aligned}|\mu(x)-\mu(y)| &\leq \textstyle\sum_{(u,v)\in P} |\mu(u)-\mu(v)| \\ &\leq \textstyle\sum_{(u,v)\in P} D(u,v) = D(x,y).\end{aligned}$$

■

Appendix B. Proof of Theorem 5 (Expressiveness of the Model)

Recall that a proof sketch for Theorem 5 was given in Section 4. In this section we complete this proof sketch by proving Equation (12).

Notation. Let us introduce the notation (some of it is from the proof sketch).

For a tree node u, let $\mathcal{T}_u$ be the node set of the subtree rooted at u. For convenience (and by a slight abuse of notation) we will write $u = b$, $b \in \{0,1\}$ to mean $\pi(u) = b$.

Fix documents $x,y \in X$. We focus on the key event, denoted $\mathcal{E}$, that no mutation happened on the $x \to y$ path. Recall that in Algorithm 1, for each tree node u with parent v we assign $\pi(u) \leftarrow M_u(\pi(v))$, where $M_u : \{0,1\} \to \{0,1\}$ is a random mutation which flips the input bit b with probability $q_b(u)$. If M_u is the identity function, then we say that no mutation happened at u. We say that no mutation happened on the $x \to y$ path if no mutation happened at each node in N_{xy}, the set of all nodes on the $x \to y$ path except z. This event is denoted $\mathcal{E}$; note that it implies $\pi(x) = \pi(y) = \pi(z)$. Its complement $\bar{\mathcal{E}}$ is, intuitively, a low-probability "failure event".

Fix a subset of documents $S \subset X$. Recall that Z_S denotes the event that all documents in S are irrelevant, that is, $\pi(x) = 0$ for all $x \in S$.

What we need to prove. We need to prove Equation (12), which states that

$$\Pr[\bar{\mathcal{E}} \,|\, Z_S] \leq 3 \Pr[\bar{\mathcal{E}}].$$

It suffices to prove the following lemma:

Lemma 13 $\Pr[\bar{\mathcal{E}} \,|\, Z_S] \leq \Pr[\bar{\mathcal{E}}] \times (2/\Pr[\mathcal{E}])$.

(Indeed, letting $p = \Pr[\bar{\mathcal{E}}]$ it holds that $\Pr[\bar{\mathcal{E}} \,|\, Z_S] \leq \min\left(1, \frac{2p}{1-p}\right) \leq 3p$.)

Remark. Lemma 13 inherits assumptions (6-7) on the mutation probabilities. Specifically for this Lemma, the upper bound (6) on mutation probabilities can be replaced with a much weaker upper bound:

$$\max(q_0(u), q_1(u)) \leq \tfrac{1}{2} \quad \text{for each tree node } u. \tag{27}$$

Our goal is to prove Lemma 13. In a sequence on claims, we will establish that

$$\Pr[Z_S \,|\, z = 0] \geq \Pr[Z_S \,|\, z = 1]. \tag{28}$$

Intuitively, (28) means that the low-probability mutations are more likely to zero out a given subset of the leaves if the value at some fixed internal node is zero (rather than one).

B.1 Using Equation (28) to Prove Lemma 13

Let us extend the notion of mutation from a single node to the $x \to y$ path. Recall that N_{xy} denotes the set of all nodes on this path except z. Then the individual node mutations $\{M_u : u \in N_{xy}\}$ collectively provide a mutation on N_{xy}, which we define simply as a function $M : N_{xy} \times \{0,1\} \to \{0,1\}$ such that $\pi(\cdot) = M(\cdot, \pi(z))$. Crucially, M is chosen independently of $\pi(z)$ (and of all other mutations). Let $\mathcal{M}$ be the set of all possible mutations of N_{xy}. By a slight abuse of notation, we treat the event $\mathcal{E}$ as the identity mutation.

Claim 14 *Fix $M \in \mathcal{M}$ and $b \in \{0,1\}$. Then*

$$\Pr[Z_S \,|\, M, \pi(z) = b] \leq \Pr[Z_S \,|\, \mathcal{E}, \pi(z) = 0].$$

Proof For each tree node u, let $S_u = S \cap \mathcal{T}_u$ be the subset of S that lies in the subtree $\mathcal{T}_u$. Then by (28)

$$\begin{aligned}\Pr[Z_S \,|\, M,\, \pi(z) = b] &= \textstyle\prod_u \Pr[Z_{S_u} \,|\, \pi(u) = M(u,b)] \\ &\leq \textstyle\prod_u \Pr[Z_{S_u} \,|\, \pi(u) = 0] \\ &= \Pr[Z_S \,|\, \mathcal{E},\, \pi(z) = 0],\end{aligned}$$

where the product is over all tree nodes $u \in N_{xy}$ such that the intersection S_u is non-empty. ■

Proof [Proof of Lemma 13] On one hand, by Claim 14

$$\begin{aligned}\Pr[Z_S \cap \bar{\mathcal{E}}] &= \textstyle\sum_{b,M} \Pr[M]\ \Pr[z = b]\ \Pr[Z_S \,|\, M, z = b] \\ &\leq \textstyle\sum_{b,M} \Pr[M]\ \Pr[z = b]\ \Pr[Z_S \,|\, \mathcal{E}, z = 0] \\ &= \Pr[\bar{\mathcal{E}}] \times \Pr[Z_S \,|\, \mathcal{E}, z = 0],\end{aligned}$$

where the sums are over bits $b \in \{0,1\}$ and all mutations $M \in \mathcal{M} \setminus \{\mathcal{E}\}$. On the other hand,

$$\Pr[Z_S] = \textstyle\sum_{b,M} \Pr[M]\ \Pr[z = b]\ \Pr[Z_S \,|\, M, z = b]$$

(where the sum is over $b \in \{0,1\}$ and $M \in \mathcal{M}$)

$$\geq \Pr[\mathcal{E}]\ \Pr[z = 0]\ \Pr[Z_S \,|\, \mathcal{E}, z = 0].$$

Since $\Pr[z = 0] \geq \frac{1}{2}$, it follows that

$$\begin{aligned}\Pr[\bar{\mathcal{E}} \,|\, Z_S] &= \Pr[Z_S \cap \bar{\mathcal{E}}] \,/\, \Pr[Z_S] \\ &\leq 2\,\Pr[\bar{\mathcal{E}}] / \Pr[\mathcal{E}].\end{aligned}$$

■

B.2 Proof of Equation (28)

First we prove (28) for the case $S \subset \mathcal{T}_z$, then we build on it to prove the (similar, but considerably more technical) case $S \cap T_z = \emptyset$. The general case follows since the events $Z_{S \cap \mathcal{T}_z}$ and $Z_{S \setminus \mathcal{T}_z}$ are conditionally independent given $\pi(z)$.

Claim 15 *If $S \subset \mathcal{T}_z$ then (28) holds.*

Proof Let us use induction the depth of z. For the base case, the case $x = y = z$. Then $S = \{z\}$ is the only possibility, and the claim is trivial.

For the induction step, consider children u_i of z such that the intersection $S_i \triangleq S \cap \mathcal{T}_{u_i}$ is non-empty. Let $u_1, \ldots, u_k$ be all such children. For brevity, denote $Z_i \triangleq Z_{S_i}$, and

$$\nu_i(a|b) \triangleq \Pr[u_i = a \,|\, z = b], \quad a, b \in \{0,1\}.$$

Note that $v_i(1,0) = q_0(x_i)$ and $v_i(0,1) = q_1(x_i)$.

Then for each $b \in \{0,1\}$ we have

$$\Pr[Z_S \,|\, z = b] = \textstyle\prod_{i=1}^{k} \Pr[Z_i \,|\, z = b] \tag{29}$$

$$\Pr[Z_i \,|\, z = b] = \textstyle\sum_{a \in \{0,1\}} \nu_i(a|b) \, \Pr[Z_i \,|\, u_i = a]. \tag{30}$$

By (29), to prove the claim it suffices to show that

$$\Pr[Z_i \,|\, z = 0] \geq \Pr[Z_i \,|\, z = 1]$$

holds for each i. By the induction hypothesis we have

$$\Pr[Z_i \,|\, u_i = 0] \geq \Pr[Z_i \,|\, u_i = 1]. \tag{31}$$

Combining (31) and (27), and noting that by (30) we have $\nu_i(0|0) \geq \nu_i(0|1)$, it follows that

$$\begin{aligned}
&\Pr[Z_i \,|\, z = 0] - \Pr[Z_i \,|\, z = 1] \\
&\quad = \textstyle\sum_{a \in \{0,1\}} \Pr[Z_i \,|\, u_i = a] \; (\nu_i(a|0) - \nu_i(a|1)) \\
&\quad \geq \Pr[Z_i \,|\, u_i = 1] \quad \textstyle\sum_{a \in \{0,1\}} (\nu_i(a|0) - \nu_i(a|1)) \\
&\quad = 0
\end{aligned}$$

because $\nu_i(0|0) + \nu_i(1|0) = \nu_i(0|1) + \nu_i(1|1) = 1$. ■

Corollary 16 *Consider tree nodes r, v, w such that r is an ancestor of v which in turn is an ancestor of w. Then for any $c \in \{0,1\}$*

$$\Pr[u = 0 \,|\, w = 0, r = c] \geq \Pr[u = 0 \,|\, w = 1, r = c].$$

Proof We claim that for each $b \in \{0,1\}$

$$\Pr[w = b \,|\, u = b] \geq \Pr[w = b \,|\, u = 1 - b]. \tag{32}$$

Indeed, truncating the subtree $\mathcal{T}_w$ to a single node w and specializing Lemma 15 to a singleton set $S = \{w\}$ (with $z = u$) we obtain (32) for $b = 0$. The case $b = 1$ is symmetric.

Now, for brevity we will omit conditioning on $\{r = c\}$ in the remainder of the proof. (Formally, we will work on in the probability space obtained by conditioning on this event.) Then for each $b \in \{0,1\}$

$$\begin{aligned}
&\Pr[u = 0 \,|\, w = b] \\
&\quad = \frac{\Pr[u = 0 \wedge w = b]}{\Pr[u = 0 \wedge w = b] \cup \Pr[u = 1 \wedge w = b]} \\
&\quad = \frac{1}{1 + \Phi(b)},
\end{aligned}$$

where

$$\begin{aligned}
\Phi(b) &\triangleq \frac{\Pr[u = 1 \wedge w = b]}{\Pr[u = 0 \wedge w = b]} \\
&= \frac{\Pr[w = b \,|\, u = 1] \Pr[u = 1]}{\Pr[w = b \,|\, u = 0] \Pr[u = 0]}
\end{aligned}$$

is decreasing in b by (32). ■

We will also need a stronger, *conditional*, version of Lemma 15 whose proof is essentially identical (and omitted).

Claim 17 *Suppose $S \subset \mathcal{T}_z$ and $u \neq z$ is a tree node such that $\mathcal{T}_u$ is disjoint with S. Then*

$$\Pr[Z_S \,|\, z = 0, u = 1] \geq \Pr[Z_S \,|\, z = 1, u = 1].$$

We will use Corollary 16 and Lemma 17 to prove (28) for the case $S \cap T_z = \emptyset$.

Claim 18 *If S is disjoint with $\mathcal{T}_z$ then (28) holds.*

Proof Suppose S is disjoint with $\mathcal{T}_z$, and let r be the root of the tree. We will use induction on the tree to prove the following: for each $c \in \{0,1\}$,

$$\Pr[Z_S \,|\, r = c, z = 0] \geq \Pr[Z_S \,|\, r = c, z = 1] \tag{33}$$

For the induction base, consider a tree of depth 2, consisting of the root r and the leaves. Then $z \notin S$ is a leaf, so Z_S is independent of $\pi(z)$ given $\pi(r)$, so (33) holds with equality.

For the induction step, fix $c \in \{0,1\}$. Let us set up the notation similarly to the proof of Claim 15. Consider children u_i of r such that the intersection $S_i \triangleq S \cap \mathcal{T}_{u_i}$ is non-empty. Let $u_1, \ldots, u_k$ be all such children. Assume $z \in \mathcal{T}_{u_i}$ for some i (else, Z_S is independent from $\pi(z)$ given $\pi(r)$, so (33) holds with equality); without loss of generality, assume this happens for $i = 1$. For brevity, for $a, b \in \{0,1\}$ denote

$$\begin{aligned}
f_i(a,b) &\triangleq \Pr[Z_{S_i} \quad | \, u_i = a, z = b] \\
\nu_i(a|b) &\triangleq \Pr[u_i = a \,|\, r = c, \;\; z = b].
\end{aligned}$$

Note that $f_i(a,b)$ and $\nu_i(a|b)$ do not depend on b for $i > 1$.

Then for each $b \in \{0,1\}$

$$\begin{aligned}
&\Pr[Z_S \,|\, r = c, z = b] \\
&\quad = \sum_{a_i \in \{0,1\},\, i \geq 1} \prod_{i \geq 1} f_i(a_i,b)\, \nu_i(a_i|b) \\
&\quad = \Phi \times \textstyle\sum_{a \in \{0,1\}} f_1(a,b)\, \nu_1(a|b),
\end{aligned}$$

where

$$\Phi \triangleq \sum_{a_i \in \{0,1\},\, i \geq 2} \prod_{i \geq 2} f_i(a_i,b)\, \nu_i(a_i|b)$$

does not depend on of b. Therefore:

$$\begin{aligned}
&\Pr[Z_S \,|\, r = c, z = 1] - \Pr[Z_S \,|\, r = c, z = 1] \\
&\quad = \Phi \times \textstyle\sum_{a \in \{0,1\}} \\
&\qquad\qquad [\, f_1(a,0)\, \nu_1(a|0) - f_1(a,1)\, \nu_1(a|1)\,] && (34) \\
&\quad \geq \Phi \times \textstyle\sum_{a \in \{0,1\}} f_1(a,1)\, [\, \nu_1(a|0) - \nu_1(a|1)\,] && (35) \\
&\quad \geq \Phi \times f_1(1,1) \textstyle\sum_{a \in \{0,1\}} [\, \nu_1(a|0) - \nu_1(a|1)\,] && (36) \\
&\quad = 0. && (37)
\end{aligned}$$

The above transitions hold for the following reasons:

(34 $\to$ 35) By Induction Hypothesis, $f_1(a,0) \geq f_1(a,1)$

(35 $\to$ 36) By Lemma 17 $f_1(0,1) \geq f_1(1,1)$, and moreover we have $\nu_1(0|0) \geq \nu_1(0|1)$ by Corollary 16.

(36 $\to$ 37) Since $\nu_i(0|0) + \nu_i(1|0) = \nu_i(0|1) + \nu_i(1|1) = 1$

This completes the proof of the inductive step. ■

Appendix C. Instance-Dependent Regret Bounds from Prior Work

In this section we discuss instance-dependent regret bounds from prior work on UCB1-style algorithms for the single-slot setting. The purpose is to put forward a concrete mathematical evidence which suggests that RankGridUCB1, RankZoom and RankCorrZoom are likely to satisfy strong upper bounds on regret in the k-slot setting (perhaps under some additional assumptions), even if such bounds are beyond the reach of our current techniques. Similarly, we believe that the regret bound for RankContextZoom that we have been able to prove (Theorem 10) is overly pessimistic. A secondary purpose is to provide more intuition for when these algorithms are likely to excel.

Our story begins with the comparison between the guarantees for EXP3 and UCB1 in the standard (single-slot, metric-free) bandit setting, and then progresses to Lipschtz MAB and contextual Lispchitz MAB.

In what follows, we let μ denote the vector of expected rewards in the stochastic reward setting, so that $\mu(x)$ is the expected reward of arm x. Let $\Delta(x) \triangleq \max \mu(\cdot) - \mu(x)$ denote the "badness" of arm x compared to the optimum.

C.1 Standard Bandits: UCB1 vs. EXP3

Algorithm EXP3(Auer et al., 2002b) achieves regret $R(T) = \tilde{O}(\sqrt{nT})$ against an oblivious adversary. In the stochastic setting, UCB1(Auer et al., 2002a) performs much better, with *logarithmic* regret for every fixed μ. More specifically, each arm $x \in X$ contributes only $O(\log T)/\Delta(x)$ to regret. Noting that the total regret from playing arms with $\Delta(\cdot) \leq \delta$ can be a priori upper-bounded by δT, we bound regret of UCB1 as:

$$R(T) = \min_{\delta>0} \left(\delta T + \sum_{x \in X: \Delta(x) > \delta} \frac{O(\log T)}{\Delta(x)} \right). \tag{38}$$

Note that Equation (38) depends on μ. In particular, if $\Delta(\cdot) \geq \delta$ then $R(T) = O(\frac{n}{\delta} \log T)$.

However, for any given T there exists a "worst-case" pointwise mean μ_T such that $R(T) = \tilde{\Theta}(\sqrt{nT})$ in Equation (38), matching EXP3. The above regret guarantees for EXP3 and UCB1 are optimal up to constant factors (Auer et al., 2002b; Kleinberg et al., 2008a).

C.2 Bandits in Metric Spaces

Let (X, D) denote the metric space. Recall that the *covering number* $N_r(X)$ is the minimal number of balls of radius r sufficient to cover X, and the *covering dimension* is defined as

$$\texttt{CovDim}(X, D) \triangleq \inf\{d \geq 0 : N_r(X) \leq \alpha r^{-d} \quad \forall r > 0\}.$$

(Here $\alpha > 0$ is a constant which we will keep implicit in the notation.)

Against an oblivious adversary, GridEXP3 has regret

$$R(T) = \tilde{O}(\alpha T^{(d+1)/(d+2)}), \tag{39}$$

where d is the covering dimension of (X, D).

For the stochastic setting, GridUCB1 and the zooming algorithm have better μ-specific regret guarantees in terms of the covering numbers. These guarantees are similar to Equation (38) for UCB1. In fact, it is possible, and instructive, to state the guarantees for all three algorithms in a common form.

Consider reward scales $\mathcal{S} = \{2^i : i \in \mathbb{N}\}$, and for each scale $r \in \mathcal{S}$ define

$$X_r = \{x \in X : r < \Delta(x) \leq 2r\}.$$

Then regret (38) of UCB1 can be restated as

$$R(T) = \min_{\delta>0} \left(\delta T + \sum_{r \in \mathcal{S}: r \geq \delta} N_{(\delta,r)} \frac{O(\log T)}{r} \right), \tag{40}$$

where $N_{(\delta,r)} = |X_r|$. Further, it follows from the analysis in (Kleinberg, 2004; Kleinberg et al., 2008b) that regret of GridUCB1 is Equation (40) with $N_{(\delta,r)} = N_\delta(X_r)$. For the zooming algorithm, the μ-specific bound can be improved to Equation (40) with $N_{(\delta,r)} = N_r(X_r)$. These results are summarized in Table C.2.

For the worst-case μ one could have $N_\delta(X_r) = N_\delta(X)$, in which case the μ-specific bound for GridUCB1 essentially reduces to Equation (39).

algorithm	regret is (40) with ...
`UCB1`	$N_{(\delta,r)} = \lvert X_r \rvert$
`GridUCB1`	$N_{(\delta,r)} = N_\delta(X_r)$
zooming algorithm	$N_{(\delta,r)} = N_r(X_r)$
`ContextZoom`	$N_{(\delta,r)} = N_r(X_{\text{dc},r})$.

Table 2: Regret bounds in terms of covering numbers

For the zooming algorithm, the μ-specific bound above implies an improved version of Equation (39) with a different, smaller d called the *zooming dimension*:

$$\texttt{ZoomDim}(X,D,\mu) \triangleq \inf\{d \geq 0 : N_r(X_r) \leq c\, r^{-d} \quad \forall r > 0\}.$$

Note that the zooming dimension depends on the triple (X,D,μ) rather than on the metric space alone. It can be as high as the covering dimension for the worst-case μ, but can be much smaller (e.g., $d = 0$) for "nice" problem instances, see (Kleinberg et al., 2008b) for further discussion. For a simple example, suppose an ε-exponential tree metric has a "high-reward" branch and a "low-reward" branch with respective branching factors $b \ll b'$. Then the zooming dimension is $\log_{1/\varepsilon}(b)$, whereas the covering dimension is $\log_{1/\varepsilon}(b')$.

C.3 Contextual Bandits in Metric Spaces

Let $\mu(x|h)$ denote the expected reward from arm x given context h. Recall that the algorithm is given metrics D and D_{c} on documents and contexts, respectively, such that for any two documents x, x' and any two contexts h, h' we have

$$|\mu(x|h) - \mu(x'|h')| \leq D(x,x') + D_{\text{c}}(h,h').$$

Let X_{c} be the set of contexts, and $X_{\text{dc}} = X \times X_{\text{c}}$ be the set of all (document, context) pairs. More abstractly, one considers the metric space $(X_{\text{dc}}, D_{\text{dc}})$, henceforth the *DC-space*, where the metric is

$$D_{\text{dc}}((x,h),(x',h')) = D(x,x') + D_{\text{c}}(h,h').$$

We partition X_{dc} according to reward scales $r \in \mathcal{S}$:

$$\Delta(x|h) \triangleq \max \mu(\cdot|h) - \mu(x|h), \quad x \in X, h \in X_{\text{c}}.$$
$$X_{\text{dc},r} \triangleq \{(x,h) \in X_{\text{dc}} : r < \Delta(x|h) \leq 2r\}.$$

Then contextual regret of `ContextZoom` can be bounded by Equation (40) with $N_{(\delta,r)} = N_r(X_{\text{dc},r})$, where $N_r(\cdot)$ now refers to the covering numbers in the DC-space (see Table C.2).

Further, one can define the *contextual* zooming dimension as

$$d_{\text{dc}}(X,D,\mu) \triangleq \inf\{d \geq 0 : N_r(X_r) \leq c\, r^{-d} \quad \forall r > 0\}.$$

Then one obtains Equation (39) with $d = d_{\text{dc}}$. In the worst case, we could have μ such that $N_r(X_{\text{dc},r}) = N_r(X_{\text{dc}})$, in which case $d_{\text{dc}} \leq \texttt{CovDim}(X_{\text{dc}}, D_{\text{dc}})$.

The regret bounds for `ContextZoom` can be improved by taking into account "benign" context arrivals: effectively, one can prune the regions of X_{c} that correspond to infrequent context arrivals, see (Slivkins, 2009) for details. This improvement can be especially significant if $\texttt{CovDim}(X_{\text{c}}, D_{\text{c}}) > \texttt{CovDim}(X,D)$.

References

Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *21th Conf. on Learning Theory (COLT)*, pages 263–274, 2008.

Rajeev Agrawal. The continuum-armed bandit problem. *SIAM J. Control and Optimization*, 33(6): 1926–1951, 1995.

David J. Aldous. Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII*, pages 1–198, 1985.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. of Machine Learning Research (JMLR)*, 3:397–422, 2002. Preliminary version in *41st IEEE FOCS*, 2000.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002a. Preliminary version in *15th ICML*, 1998.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002b. Preliminary version in *36th IEEE FOCS*, 1995.

Peter Auer, Ronald Ortner, and Csaba Szepesvári. Improved rates for the stochastic continuum-armed bandit problem. In *20th Conf. on Learning Theory (COLT)*, pages 454–468, 2007.

Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *J. of Computer and System Sciences*, 74(1):97–114, February 2008. Preliminary version in *36th ACM STOC*, 2004.

Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, 1996.

Dirk Bergemann and Juuso Välimäki. Bandit problems. In Steven Durlauf and Larry Blume, editors, *The New Palgrave Dictionary of Economics, 2nd ed.* Macmillan Press, 2006.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning (Draft under submission)*, 2012. Available at `www.princeton.edu/∼sbubeck/pub.html`.

Sébastien Bubeck and Rémi Munos. Open loop optimistic planning. In *23rd Conf. on Learning Theory (COLT)*, pages 477–489, 2010.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvari. Online optimization in x-armed bandits. *J. of Machine Learning Research (JMLR)*, 12:1587–1627, 2011. Preliminary version in *NIPS 2008*.

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In *Intl. Conf. on Machine Learning (ICML)*, pages 89–96, 2005.

Jaime G. Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *ACM Intl. Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 335–336, 1998.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge Univ. Press, 2006.

Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. *J. of Machine Learning Research*, 6:1019–1041, 2005.

Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandits with linear payoff functions. In *14thIntl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2011.

Varsha Dani, Thomas P. Hayes, and Sham Kakade. The price of bandit information for online optimization. In *20th Advances in Neural Information Processing Systems (NIPS)*, 2007.

Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. of Computer and System Sciences*, 69(3):485–497, 2004.

Abraham Flaxman, Adam Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 385–394, 2005.

Daniel Golovin, Andreas Krause, and Matthew Streeter. Online learning of assignments. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

Anupam Gupta, Robert Krauthgamer, and James R. Lee. Bounded geometries, fractals, and low–distortion embeddings. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, 2003.

Elad Hazan and Satyen Kale. Better algorithms for benign bandits. In *20th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 38–47, 2009.

Elad Hazan and Nimrod Megiddo. Online learning with prior information. In *20th Conf. on Learning Theory (COLT)*, pages 499–513, 2007.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 133–142, 2002.

Satyen Kale, Lev Reyzin, and Robert E. Schapire. Non-stochastic bandit slate problems. In *24th Advances in Neural Information Processing Systems (NIPS)*, pages 1054–1062, 2010.

Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *18th Advances in Neural Information Processing Systems (NIPS)*, 2004.

Robert Kleinberg and Aleksandrs Slivkins. Sharp dichotomies for regret minimization in metric spaces. In *21st ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2010.

Robert Kleinberg, Alexandru Niculescu-Mizil, and Yogeshwer Sharma. Regret bounds for sleeping experts and bandits. In *21st Conf. on Learning Theory (COLT)*, pages 425–436, 2008a.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *40th ACM Symp. on Theory of Computing (STOC)*, pages 681–690, 2008b.

Levente Kocsis and Csaba Szepesvari. Bandit based Monte-Carlo planning. In *17th European Conf. on Machine Learning (ECML)*, pages 282–293, 2006.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

John Langford and Tong Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *21st Advances in Neural Information Processing Systems (NIPS)*, 2007.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *19th Intl. World Wide Web Conf. (WWW)*, 2010.

Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *4th ACM Intl. Conf. on Web Search and Data Mining (WSDM)*, 2011.

Tyler Lu, Dávid Pál, and Martin Pál. Showing relevant ads via Lipschitz context multi-armed bandits. In *14thIntl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2010.

Odalric-Ambrym Maillard and Rémi Munos. Online learning in adversarial lipschitz environments. In *European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 305–320, 2010.

Rémi Munos and Pierre-Arnaud Coquelin. Bandit algorithms for tree search. In *23rd Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2007.

Sandeep Pandey, Deepak Agarwal, Deepayan Chakrabarti, and Vanja Josifovski. Bandits for taxonomies: A model-based approach. In *SIAM Intl. Conf. on Data Mining (SDM)*, 2007.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *25th Intl. Conf. on Machine Learning (ICML)*, pages 784–791, 2008.

Philippe Rigollet and Assaf Zeevi. Nonparametric bandits with covariates. In *23rd Conf. on Learning Theory (COLT)*, pages 54–66, 2010.

Aleksandrs Slivkins. Contextual bandits with similarity information. http://arxiv.org/abs/0907.3986, 2009. Has been published in *24th COLT 2011*.

Aleksandrs Slivkins. Multi-armed bandits on implicit metric spaces. In *25th Advances in Neural Information Processing Systems (NIPS)*, 2011.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *27th Intl. Conf. on Machine Learning (ICML)*, pages 1015–1022, 2010.

Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1577–1584, 2008.

Rangarajan K. Sundaram. Generalized bandit problems. In David Austen-Smith and John Duggan, editors, *Social Choice and Strategic Decisions: Essays in Honor of Jeffrey S. Banks (Studies in Choice and Welfare)*, pages 131–162. Springer, 2005. First appeared as *Working Paper, Stern School of Business*, 2003.

Michael J. Taylor, John Guiver, Stephen Robertson, and Tom Minka. Softrank: Optimizing non-smooth rank metrics. In *ACM Intl. Conf. on Web Search and Data Mining (WSDM)*, pages 77–86, 2008.

Taishi Uchiya, Atsuyoshi Nakamura, and Mineichi Kudo. Algorithms for adversarial bandit problems with multiple plays. In *21st Intl. Conf. on Algorithmic Learning Theory (ALT)*, pages 375–389, 2010.

Chih-Chun Wang, Sanjeev R. Kulkarni, and H. Vincent Poor. Bandit problems with side observations. *IEEE Trans. on Automatic Control*, 50(3):338355, 2005.

Yizao Wang, Jean-Yves Audibert, and Rémi Munos. Algorithms for infinitely many-armed bandits. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1729–1736, 2008.

Michael Woodroofe. A one-armed bandit problem with a concomitant variable. *J. Amer. Statist. Assoc.*, 74(368), 1979.