Contextual Bandits with Similarity Information

Aleksandrs Slivkins

First version: February 2009
This revision: July 2009

Abstract

In a multi-armed bandit (MAB) problem, an online algorithm makes a sequence of choices. In each round it chooses from a time-invariant set of alternatives and receives the payoff associated with this alternative. While the case of small strategy sets is by now well-understood, a lot of recent work has focused on MAB problems with exponentially or infinitely large strategy sets, where one needs to assume extra structure in order to make the problem tractable. In particular, recent literature considered information on similarity between arms.

We consider similarity information in the setting of contextual bandits, a natural extension of the basic MAB problem where before each round an algorithm is given the context—a hint about the payoffs in this round. Contextual bandits are directly motivated by placing advertisements on webpages, one of the crucial problems in sponsored search. A particularly simple way to represent similarity information in the contextual bandit setting is via a similarity distance between the context-arm pairs which gives an upper bound on the difference between the respective expected payoffs.

Prior work on contextual bandits with similarity uses “uniform” partitions of the similarity space, so that, essentially, each context-arm pair is approximated by the closest pair in the partition. Algorithms based on “uniform” partitions disregard the structure of the payoffs and the context arrivals. This is potentially wasteful, as it may be advantageous to maintain a finer partition in high-payoff regions of the similarity space and popular regions of the context space. In this paper, we design algorithms that are based on adaptive partitions which are adjusted to the popular and high-payoff regions, and thus can take advantage of the problem instances with “benign” payoffs or context arrivals.

ACM Categories and subject descriptors: F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems; F.1.2 [Computation by Abstract Devices]: Modes of Computation—Online computation

General Terms: Algorithms, Theory.

Keywords: online learning, multi-armed bandits, contextual bandits, regret minimization, metric spaces.

* A preliminary draft of Section 5 has been completed while the author was a postdoctoral research associate at Brown University, and circulated in 2007.
† MSR-SVC (Microsoft Research, Silicon Valley Center), Mountain View, CA 94043. slivkins at microsoft.com.
1 Introduction

In a multi-armed bandit problem (henceforth, “multi-armed bandit” will be abbreviated as MAB), an algorithm is presented with a sequence of trials. In each round, the algorithm chooses one alternative from a set of alternatives (arms) based on the past history, and receives the payoff associated with this alternative. The goal is to maximize the total payoff of the chosen arms. The MAB setting has been introduced in 1952 by Robbins [31] and studied intensively since then in Operations Research, Economics and Computer Science, e.g. see [9, 16, 11, 8]. This setting is used to model the trade-off between exploration and exploitation, a crucial issue in sequential decision-making under uncertainty.

One standard way to evaluate the performance of a bandit algorithm is regret, defined as the difference between the expected payoff of an optimal arm and that of the algorithm. By now the MAB problem with a small finite set of arms is quite well understood (e.g. see [26, 3, 3]). However, if the arms set is exponentially or infinitely large, the problem becomes intractable unless we make further assumptions about the problem instance. Essentially, a bandit algorithm needs to find a needle in a haystack; for each algorithm there are inputs on which it performs as badly as random guessing.

The bandit problems with large arm sets have been studied in [7, 2, 4, 6, 20, 28, 21, 15, 12, 13, 14, 5, 19, 24]. The common theme in these works is to assume a certain structure on payoff functions. Assumptions of this type are natural in many applications, and often lead to efficient learning algorithms [21]. In particular, the line of work [2, 20, 5, 24] on the (non-contextual) bandits, a particularly simple way to represent similarity information in [32, 30, 27], a natural extension of the basic MAB problem where before each round an algorithm is given the context – a hint about the payoffs in this round. Contextual bandits are directly motivated by the problem of placing advertisements on webpages, one of the crucial problems in sponsored search. One can cast it as a bandit problem so that arms correspond to the possible ads, and payoffs correspond to the user clicks. Then the context consists of information about the page, and perhaps the user this page is served to. Furthermore, we assume that similarity information is available on both the context and the arms. Following the work in [2, 20, 5, 24] on the (non-contextual) bandits, a particularly simple way to represent similarity information in the contextual bandit setting is via a similarity distance between the context-arm pairs, gives an upper bound on the difference between the corresponding payoffs.

Our model: contextual bandits with similarity information. For simplicity, let us develop the model for the stochastic (non-adversarial) case only. Let $X$ be the context set and $Y$ be the arms set. In each round $t$, the following events happen in succession: first, the current context arrival $x_t \in X$ is chosen by an adversary and revealed to the algorithm, then the algorithm chooses an arm $y_t \in Y$, and finally the payoff $\pi_t(x_t, y_t) \in [0, 1]$ is realized. Here $\pi_t : X \times Y \rightarrow [0, 1]$ is the payoff function defined as an independent random sample from some fixed distribution $\Pi$ over functions $X \times Y \rightarrow [0, 1]$. Both the contexts $(x_t)_{t \in \mathbb{N}}$ and the distribution $\Pi$ are fixed by an adversary before the first round and are not revealed to the algorithm.

The performance of a contextual bandit algorithm is benchmarked in terms of the context-specific best arm. Denoting $\mu(x, y) \triangleq \mathbb{E}[\pi_t(x, y)]$, the contextual regret is defined as

$$R(T) \triangleq \sum_{t=1}^{T} \mu(x_t, y_t) - \mu^*(x_t), \quad \text{where} \quad \mu^*(x) \triangleq \sup_{y \in Y} \mu(x, y).$$

The similarity information is given to an algorithm as two metric spaces $(X, \mathcal{D}_X)$ and $(Y, \mathcal{D}_Y)$ called, respectively, the context space and the arms space, such that the following Lipschitz condition holds:

$$|\mu(x, y) - \mu(x', y')| \leq \mathcal{D}_X(x, x') + \mathcal{D}_Y(y, y').$$  \hspace{1cm} (1)

\(^1\)The absence of similarity information on arms is modeled as $\mathcal{D}_Y \triangleq 1$. 


It will be useful to consider the product metric space \((X \times Y, D_X + D_Y)\), which we call the \textit{similarity space}. Without loss of generality, the distances in \((X, D_X)\) and \((Y, D_Y)\) are truncated at 1; to guarantee the existence of a context-specific best arm, we will assume that both metric spaces are compact.

We term this problem the \textbf{contextual MAB problem}. Note that if the context \(x_t\) is time-invariant, the contextual MAB problem reduces to the stochastic MAB problem \cite{4}, or (when the metric similarity information is present) to the Lipschitz MAB problem \cite{24}; the contextual regret reduces to the “standard” (context-free) regret. In general, the context-specific best arm is a more demanding benchmark than the best arm used in the “standard” (context-free) definition of regret.

\textbf{Uniform vs adaptive partitions.} Hazan and Megiddo \cite{18} suggest (for a somewhat different setting) an algorithm that chooses a “uniform” partition \(S_X\) of the context space (namely, an \(r\)-net) and approximates \(x_t\) by the closest point in \(S_X\), call it \(x'_t\). Specifically, the algorithm creates an instance \(A(x)\) of some bandit algorithm \(A\) for each point \(x \in S_X\), and invokes \(A(x'_t)\) in each round \(t\). The granularity of the partition is adjusted to the time horizon, the context space, and the black-box regret guarantee for \(A\).

Furthermore, Kleinberg \cite{20} provides a bandit algorithm \(A\) for the adversarial MAB problem on a metric space that has a similar flavor: pick a “uniform” partition \(S_Y\) of the arms space, and run a \(k\)-arm bandit algorithm such as \(\text{EXP3}\) \cite{4} on the points in \(S_Y\). Again, the granularity of the partition is adjusted to the time horizon, the arms space, and the black-box regret guarantee for \(\text{EXP3}\).

For our setting, putting these two ideas together results in an algorithm with contextual regret

\[
R(T) \leq O(T^{1-1/(2+d_X+d_Y)})(\log T),
\]

where \(d_X\) is the covering dimension of the context space and \(d_Y\) is that of the arms space. (This holds even for adversarial payoffs.) We will call it the \textit{uniform algorithm}.

Using “uniform” partitions disregards the structure of the payoffs and the context arrivals. This is potentially wasteful, as it may be advantageous to maintain a finer partition in higher-paying regions of the similarity space, and more popular regions of the context space. The central topic in this paper is designing algorithms that are based on \textit{adaptive partitions} which are adjusted to the popular/high-payoff regions, and thus can take advantage of the problem instances in which the expected payoffs or the context arrivals are, in some sense, “benign” (“low-dimensional”). A similar task has been accomplished in \cite{24} for the context-free setting; however, the present contextual setting is considerably more challenging.

\textbf{Main contribution: the contextual zooming algorithm.} We consider the contextual MAB problem, and present an algorithm, called the \textit{contextual zooming algorithm}, which “zooms in” on both the “popular” regions on the context space and the high-payoff regions of the arms space. The algorithm considers the context space and the arms space jointly – it maintains a partition of the similarity space, rather than one partition for contexts and another for arms – which allows it to use the similarity information more efficiently. We develop provable guarantees that capture the “benign-ness” of the context arrivals and the expected payoffs. In the worst case, we match the guarantee (2) for the uniform algorithm. For the context-free setting, our guarantees match those in \cite{24}. We obtain nearly matching lower bounds using the KL-divergence techniques from \cite{4, 20, 24}. Our algorithm and analysis extends to a more general setting where some context-arms pairs may be unfeasible, and moreover the right-hand side of (1) is replaced by an arbitrary metric on the feasible context-arms pairs (which is revealed to the algorithm).

We apply the contextual zooming algorithm to a (context-free) adversarial MAB problem in which an adversary is constrained to change the expected payoffs of each arm \textit{gradually}, e.g. by a small amount in each round. This setting is naturally modeled as a contextual MAB problem in which the \(t\)-th context arrival is simply \(x_t = t\). Then \(\mu(t, y)\) corresponds to the expected payoff of arm \(y\) at time \(t\), and the context metric
\( D_X(t, t') \) provides an upper bound on the temporal change \( |\mu(t, y) - \mu(t', y)| \). We term it the **drifting MAB problem**. Interestingly, this problem incorporates significant constraints both across time (for each arm) and across arms (for each time); to the best of our knowledge, such MAB models are quite rare in the literature. Notable special cases of \( D_X \) include \( D_X(t, t') = \sigma|t - t'| \) and \( D_X(t, t') = \sigma\sqrt{|t - t'|} \), which corresponds to, respectively, the bounded change per round and the high-probability behavior of a random walk. We derive provable guarantees for these two examples, and show that they are essentially optimal.

Interestingly, the contextual MAB problem subsumes the stochastic **sleeping bandits** problem [23], where in each round some arms are “asleep”, i.e. not available in this round. Each context arrival \( x_t \) corresponds to the set of arms that are “awake” in this round. More precisely, contexts \( x \) correspond to subsets \( S_x \) of arms, so that only the context-arm pairs \((x, y), y \in S_x\) are feasible, and the context distance is \( D_X \triangleq 0 \). Moreover, the contextual MAB problem extends the sleeping bandits setting by incorporating similarity information on arms. The contextual zooming algorithm (and its analysis) applies, and is geared to exploit this additional similarity information. In the absence of such information the algorithms essentially reduces to the “highest awake index” algorithm in [23].

**Other contributions.** We turn our attention to the contextual MAB problem with adversarial payoffs (see Section 2 for the setup). We describe an algorithm that is geared to take advantage of “benign” sequence of context arrivals. It is in fact a **meta-algorithm**: given an adversarial bandit algorithm Bandit, and a regret bound for that algorithm, we present a contextual bandit algorithm ContextualBandit which calls Bandit as a subroutine. The crucial feature of our algorithm is that it maintains a partition of the context space that is adapted to the context arrivals. Again, we develop provable guarantees that capture the “benignancy” of the context arrivals. In the worst case, the contextual regret is bounded in terms of the covering dimension of the context space (and the regret bound for Bandit). In particular, we match the guarantee (2) for the uniform algorithm.

One inherent drawback of MAB problems on metric spaces is that they require the entire metric space to be revealed to the algorithm, whereas in reality only coarser similarity information may be available. To this end, we consider a version of the stochastic MAB problem where the similarity information is given by a tree-shaped taxonomy on arms, without an explicitly specified similarity distance. However, the taxonomy implicitly defines a similarity distance as follows: the distance between two arms is equal to the maximal variation in expected payoffs in their least common subtree. We provide an algorithm for this setting, called TaxonomyBandit. Under appropriate assumptions about the “quality” of the taxonomy, we obtain guarantees similar to those for the setting where the induced similarity metric is explicitly revealed to the algorithm. The novel technical issue here is that apart from the familiar exploration-exploitation trade-off regarding payoffs there is now a similar exploration-exploitation trade-off regarding the induced similarity distance. The two trade-offs need to be handled jointly, which leads to significant technical complications. Obviously, TaxonomyBandit can be used as a subroutine for the contextual meta-algorithm from the previous paragraph. We conjecture that the same technique can be used to extend the contextual zooming algorithm to a setting where both the context space and the arms space are given via taxonomies of suitable “quality”, with essentially matching guarantees.

---

2 The only other MAB model with this flavor that we are aware of, found in Hazan and Kale [17], combines linear payoffs and bounded “total variation” of the cost functions. The latter is, essentially, an aggregate bound on the temporal change.

3 Apart from using the corresponding “naive” algorithm, our presentation allows us to leverage prior work on other adversarial MAB formulations, such as the basic \( k \)-arm version [4], linear payoffs [28, 6, 14, 17] and convex payoffs [20, 15].

4 This setting has been introduced in [30], an experimental paper with a web advertising motivation, and later studied in [25, 29]. The guarantees in this paper are incomparable with the ones in [25, 29]: they are much stronger, but in a more tractable model.
Organization of the paper. The contextual zooming algorithm is presented in Section 3 (see Section 3.2 for the application to the drifting MAB problem). Lower bounds are in Section 4. The meta-algorithm for adversarial payoffs is presented in Section 5. The taxonomy-based similarity information is treated in Section 6.

2 Preliminaries

We will use the notation from the Introduction without further notice. In particular, \( x_t \) will denote the \( t \)-th context arrival, i.e. the context that arrives in round \( t \), and \( y_t \) will denote the arm chosen by the algorithm in that round. We will use \( x_{(1:T)} \) to denote the sequence of the first \( T \) context arrivals \( (x_1, \ldots, x_T) \).

Metric spaces. Fix a metric space. A ball with center \( x \) and radius \( r \) is denoted \( B(x, r) \). Formally, we will treat a ball as a (center, radius) pair rather than a set of points. Let \( S \) be a set of points. The \( r \)-covering number of \( S \) is the smallest number of sets of diameter \( r \) that is sufficient to cover \( S \). The covering dimension of \( S \) (with constant \( c \)) is the smallest \( d \) such that the \( r \)-covering number is at most \( c r^{-d} \) for each \( r > 0 \). The doubling constant of \( S \) is the smallest \( c_{\text{dBL}} \) such that for any \( \delta \in (0, 1) \), any subset \( S' \subset S \) of diameter \( r \) can be covered by \( c_{\text{dBL}} \log \delta \) sets of diameter \( \delta r \). The covering dimension (with constant 1) is at most \( \log c_{\text{dBL}} \), but can be much smaller in general. The following fact is well-known: if distance between any two points in \( S \) is \( > r \), then any ball of radius \( r \) contains at most \( c_{\text{dBL}}^2 \) points of \( S \).

A function \( f : X \to \mathbb{R} \) if a Lipschitz function on a metric space \( (X, D) \) if the following Lipschitz condition holds: \( |f(x) - f(x')| \leq D(x, x') \) for each \( x, x' \in X \).

Adversarial payoffs. Let us set up the contextual MAB problem with adversarial payoffs and support \( \mathcal{F} \). The main distinction is that now in each round \( t \), the payoff function \( \pi_t \) is sampled from a time-specific distribution \( \Pi_t \) over the set \( \mathcal{F} \) of all possible payoff functions. Fixing the time horizon \( T \) and denoting \( \mu_t(x, y) \doteq \mathbb{E}[\pi_t(x, y)] \), the context-specific best arm is defined as

\[
y_t^*(x) \in \arg\max_{y \in Y} \sum_{t=1}^{T} \mu_t(x, y),
\]

where the ties are broken in an arbitrary but fixed way.\(^5\) The contextual regret is defined as

\[
R(T) \doteq \sum_{t=1}^{T} \mu_t(x_t, y_t) - \mu_t^*(x_t), \quad \text{where} \quad \mu_t^*(x) \doteq \mu_t(x, y^*(x_t)).
\]

Analogously to (1), for each round \( t \) the expected payoffs function \( \mu_t \) is a Lipschitz function on the similarity space. Moreover, the benchmark payoff \( \mu_t^* \) is a Lipschitz function on the context space.

Accessing the metric spaces. We assume full and computationally unrestricted access to the similarity information. While the issues of efficient representation thereof are important in practice, we believe that a proper treatment of these issues would be specific to the particular application and the particular similarity metric used, and would obscure the present paper. One clean formal way to model this issue is to assume oracle access to the metric spaces in question, e.g. via an oracle that inputs a collection of balls and a set, and outputs an arbitrary point in the set which is not covered by these balls, if there is any.

Fixed vs arbitrary time horizon. There is a well-known doubling trick which converts a bandit algorithm ALG with a fixed time horizon to one that runs indefinitely and achieves essentially the same regret bound for every given round. Namely, in each phase \( i = 1, 2, 3, \ldots \), run a fresh instance of ALG for \( 2^i \) rounds. With this trick in mind, we will assume fixed time horizon throughout this paper.

\(^5\)The max in (3) is attained by some \( y \in Y \) as a maximum of a Lipschitz function on a compact metric space.
3 The contextual zooming algorithm

In this section we consider the (stochastic) contextual MAB problem. We present an algorithm for this problem, called the contextual zooming algorithm, which takes advantage of both the “benign” context arrivals and the “benign” expected payoffs by “zooming in” on both the “popular” regions on the context space and the high-payoff regions of the arms space. The algorithm considers the context space and the arms space jointly; it adaptively maintains a partition of the similarity space.

We will consider a model that generalizes the one from the Introduction in the following two directions. First, not all context-arm pairs \((x, y)\) will be allowed: there will be a set \(\mathcal{P} \subset X \times Y\) of feasible context-arms pairs. Using the notation \(\mu(x, y) \triangleq \mu_t(x, y)\) for the expected payoff function, the benchmark payoff is \(\mu^*(x) \triangleq \sup_{(x, y) \in \mathcal{P}} \mu(x, y)\). Second, instead of the context space and the arms space there will be a metric space \((\mathcal{P}, D)\), revealed to the algorithm, such that the following Lipschitz conditions hold:

\[
|\mu(x, y) - \mu(x', y')| \leq D((x, y), (x', y')) \tag{4}
\]

This \((\mathcal{P}, D)\) is called the similarity space. This will be the only metric space that appears in this section.

Our algorithm maintains a covering of the similarity space with balls. We formulate the provable guarantees in terms of (essentially) the maximal number of balls in such a covering, under various restrictions that are implicitly enforced by the algorithm.

**Definition 3.1.** Consider an instance of the contextual MAB problem. The \(r\)-zooming number \(N(r)\) is defined as the \(r\)-covering number of the set \(\mathcal{P}_r \triangleq \{(x, y) \in \mathcal{P} : \mu^*(x) - \mu(x, y) \leq 12r\}\). Given a constant \(c > 0\), the contextual zooming dimension is the smallest \(d > 0\) such that \(N(r) \leq c r^{-d}\) for all \(r > 0\).

It is easy to see that the contextual zooming dimension is bounded from above by the covering dimension, but can be much smaller for “benign” expected payoffs. To take into account “benign” context arrivals, a more complicated setup is needed (without any modification to the algorithm), see Section 3.1 for details.

**Theorem 3.2.** The contextual regret of the contextual zooming algorithm is

\[
R(T) \leq \inf_{r_0 > 0} O \left( r_0 T + \sum_{r=2^{-i}: r_0 \leq r \leq 1} \frac{1}{r} N(r) \log(T) \right) \leq O(c T^{1-1/(2+d)} \log T), \tag{5}
\]

where \(N(r)\) is the \(r\)-zooming number and \(d\) is the contextual zooming dimension with any given constant \(c\).

Note that the bounds in Theorem 3.2 hold for each \(c > 0\). This is useful since different values of \(c\) may result in drastically different values of the \(c\)-zooming dimension.

The **contextual zooming algorithm.** The algorithm is parameterized by the time horizon \(T\). In each round \(t\), it maintains a finite collection of balls (called active balls) which collectively cover the similarity space. Adding active balls is called activating; balls stay active once they are activated. Initially there is only one active ball which has radius 1 and contains the entire similarity space.

In each round \(t\), each active ball \(B\) is assigned a number \(I_t(B)\) called index, defined as follows. Let \(r(B)\) the radius of \(B\), let \(n_t(B)\) be the number of times it has been selected so far by the algorithm, and let \(\nu_t(B)\) be the corresponding average payoff. (Define the latter to be 0 if the ball has not been selected yet.) The pre-index of \(B\) is defined as

\[
I_t^\text{pre}(B) \triangleq \nu_t(B) + 2r(B) + \text{rad}_t(B), \quad \text{where} \quad \text{rad}_t(B) \triangleq 4 \sqrt{\frac{\log T}{1 + n_t(B)}}. \tag{6}
\]

The quantity \(\text{rad}_t(B)\) is called the confidence radius of \(B\) at time \(t\). Finally, the index of \(B\) is

\[
I_t(B) \triangleq \min_{B'} (I_t^\text{pre}(B') + D(B, B')), \tag{7}
\]
where the minimum is over all balls $B'$ of radius $r(B') \geq r(B)$ that have been activated before time $t$, and the distance $D(B, B')$ between two balls $B$ and $B'$ is defined as the distance between their centers.

The algorithm operates as follows. In each round $t$, the domain of an active ball $B$, denoted $\text{dom}_t(B)$, is a subset of $B$ defined as $B$ minus the union of all balls of strictly smaller radius that are active at time $t$. Ball $B$ is called relevant in round $t$ if it is active and $(x_t, y) \in \text{dom}_t(B)$ for some arm $y \in Y$. The algorithm picks an arm to play according to the following selection rule: it selects a relevant ball $B$ with the maximal index (breaking ties arbitrarily) and an arbitrary arm $y$ such that $(x_t, y) \in \text{dom}_t(B)$. Balls are activated according to the following activation rule: if an active ball $B$ is selected in a round $t$ such that $r_\text{act}(B) \leq r(B)$, then a ball $B'$ with center $(x_t, y_t)$ and radius $\frac{1}{2} r(B)$ is activated; $B$ is then called the parent of $B'$. This completes the specification.

**Remark.** In the algorithm, the “domains” and the activation rule are defined in a specific way to ensure that each active ball is centered in a point $(x_t, y_t)$ such that in round $t$ its parent ball is selected (rather than in some arbitrary point inside the parent ball). This is useful for the general setting considered in this section, whereas with some other reasonable versions of the activation rule we need an additional assumption that (essentially) the benchmark payoff $\mu^*$ satisfies a Lipschitz property.

**Claim 3.3.** The following invariants are maintained:

- (covering) in each round, active balls cover the similarity space,
- (separation) for any two active balls of radius $r$, their centers are at distance at least $r$.

**Proof.** The activation rule makes sure that the following property (*) holds: if a ball $B(p, r)$ is activated in round $t$, then in any subsequent round $t'$ it is covered by active balls of radius $\leq r$. (To show this, use induction on $t'$.) The covering invariant holds initially, and so by (*) it holds in every subsequent round.

To show the separation invariant, suppose $B$ and $B'$ be two balls of radius $r$ such that $B$ is activated at time $t$, with parent $B_\text{par}$, and $B'$ is activated before time $t$. The parent ball $B_\text{par}$ has radius $2r$, so by property (*) its round-$t$ domain is disjoint with $B'$. By the activation rule, the center of $B$ is $(x_t, y_t)$, and by the selection rule $(x_t, y_t) \in \text{dom}_t(B_\text{par})$. It follows that $(x_t, y_t) \notin B'$.

**Analysis.** Throughout the analysis we will use the following notation. For a ball $B$ with center $(x, y)$, denote the radius as $r(B)$, and define the expected payoff of $B$ as $\mu(B) \equiv \mu(x, y)$. The badness of $(x, y)$ is defined as $\Delta(x, y) \equiv \mu^*(x) - \mu(x, y)$. Let $B_t^\text{sel}$ be the active ball selected by the algorithm in round $t$.

**Claim 3.4.** If ball $B$ is active in round $t$, then with probability at least $1 - T^{-3}$ we have that

$$|\nu_t(B) - \mu(B)| \leq r(B) + r\text{act}(B).$$

**Proof.** Fix ball $V$ with center $(x, y)$. Let $S$ be the set of rounds $s \leq t$ when ball $B$ was selected by the algorithm, and let $n = |S|$ be the number of such rounds. Then $\nu_t(B) = \frac{1}{n} \sum_{s \in S} \pi_s(x_s, y_s)$.

Define $Z_k = \sum (\pi_s(x_s, y_s) - \mu(x_s, y_s))$, where the sum is taken over the $k$ smallest elements $s \in S$. Then $\{Z_k\}$ is a martingale with bounded increments, so by the Azuma-Hoeffding inequality with probability at least $1 - T^{-3}$ we have that $\frac{1}{n} |Z_n| \leq r\text{act}(B)$. Note that $|\mu(x_s, y_s) - \mu(B)| \leq r(B)$ for each $s \in S$, so $|\nu_t(B) - \mu(B)| \leq r(B) + \frac{1}{n} |Z_n|$, which completes the proof.

Call a run of the algorithm clean if (8) holds for each round. From now on we will focus on a clean run, and argue deterministically using (8). The heart of the analysis is the following lemma.

**Lemma 3.5.** Consider a clean run of the algorithm. Then $\Delta(x_t, y_t) \leq 15 r(B_t^\text{sel})$ in each round $t$.

---

6This assumption is satisfied for a narrower setting defined in the Introduction.
Proof. Fix round $t$. By the covering invariant, $(x_t, y^*(x_t)) \in B$ for some active ball $B$. Recall from (7) that $I_t(B) = I_{pre}(B') + D(B, B')$ for some active ball $B'$ of radius $r(B') \geq r(B)$. Therefore

\[
I_t(B_t^\text{sel}) \geq I_t(B) = I_{pre}(B') + D(B, B') = \nu_t(B') + 2r(B') + \text{rad}_t(B') + D(B, B') \\
\geq \mu(B') + r(B) + D(B, B') \quad \text{(selection rule, defn of index (7))} \\
\geq \mu(B) + r(B) \geq \mu(x, y^*(x_t)) = \mu^*(x_t). \quad \text{(Lipschitz property (4), twice)} \quad (9)
\]

On the other hand, letting $B_{t, \text{par}}$ be the parent of $B_t^\text{sel}$ and noting that by the selection rule

\[
\text{rad}_t(B_{t, \text{par}}) \leq r(B_{t, \text{par}}) = 2r(B_t^\text{sel}),
\]

we can upper-bound $I_t(B_t^\text{sel})$ as follows:

\[
I_t(B_t^\text{sel}) \leq I_{pre}(B_{t, \text{par}}) + r(B_{t, \text{par}}) = \nu_t(B_{t, \text{par}}) + 3r(B_{t, \text{par}}) + \text{rad}_t(B_{t, \text{par}}) \\
\leq \mu(B_{t, \text{par}}) + 4r(B_{t, \text{par}}) + 2\text{rad}_t(B_{t, \text{par}}) \quad \text{(Claim 3.4) (10)} \\
\leq \mu(B_{t, \text{par}}) + 12r(B_t^\text{sel}) \quad \text{("parenthood" (10))} \\
\leq \mu(x, y_t) + 15r(B_t^\text{sel}). \quad \text{(Lipschitz property (4)).} \quad (11)
\]

In the last inequality we used the fact that $(x_t, y_t)$ is within distance $3r(B_t^\text{sel})$ from the center of $B_{t, \text{par}}$. Putting the pieces together, $\mu^*(x_t) \leq I_t(B_t^\text{sel}) \leq \mu(x, y_t) + 15r(B_t^\text{sel})$. \hfill \Box

Corollary 3.6. In a clean run, if ball $B$ is activated in round $t$ then $\Delta(x_t, y_t) \leq 12r(B)$. \hfill \Box

Proof. By the activation rule, $B_t^\text{sel}$ is the parent of $B$. Thus by Lemma 3.5 we immediately have $\Delta(x_t, y_t) \leq 15r(B_t^\text{sel}) = 30r(B)$. To obtain the constant of 12 that is claimed here, it suffices to prove a more efficient special case of Lemma 3.5 if $\text{rad}_t(B_t^\text{sel}) \leq r(B_t^\text{sel})$ then $\Delta(x_t, y_t) \leq 6r(B_t^\text{sel})$. To prove this, we simply replace (11) in the proof of Lemma 3.5 by similar inequality in terms of $I_{pre}(B_t^\text{sel})$ rather than $I_{pre}(B_{t, \text{par}})$:

\[
I_t(B_t^\text{sel}) \leq I_{pre}(B_t^\text{sel}) = \nu_t(B_t^\text{sel}) + 2r(B_t^\text{sel}) + \text{rad}_t(B_t^\text{sel}) \quad \text{(defs (6,7))} \\
\leq \mu(B_t^\text{sel}) + 3r(B_t^\text{sel}) + 2\text{rad}_t(B_t^\text{sel}) \quad \text{(Claim 3.4)} \\
\leq \mu(x, y_t) + 6r(B_t^\text{sel}) \quad \Box
\]

Now we are ready for the final regret computation.

**Proof of Theorem 3.2.** For a given $\gamma = 2^{-i}, i \in \mathbb{N}$, let $\mathcal{F}_r$ be the collection of all balls of radius $r$ that have been activated throughout the execution of the algorithm. A ball $B \in \mathcal{F}_r$ is called full in round $t$ if $\text{rad}_t(B) \leq r$. Note that in each round, if a full ball is selected then some other ball is activated. Thus, we will partition the rounds among active balls as follows: for each ball $B \in \mathcal{F}_r$, let $S_B$ be the set of rounds which consists of the round when $B$ was activated and all rounds $t$ when $B$ was selected and not full. It is easy to see that $|S_B| \leq O(r^{-2} \log T)$. Moreover, by Lemma 3.5 and Corollary 3.6 we have $\Delta(x_t, y_t) \leq 15r$ in each round $t \in S_B$.

If ball $B \in \mathcal{F}_r$ is activated in round $t$, then by the activation rule its center is $(x_t, y_t)$, and Corollary 3.6 asserts that $(x_t, y_t) \in \mathcal{F}_r$, as defined in Definition 3.1. By the separation invariant, the centers of balls in $\mathcal{F}_r$ are within distance $\gamma$ from one another. It follows that these centers can be covered by $N(r)$ sets of diameter $r$, where $N(r)$ is the $r$-zooming dimension of the problem instance, and so $|\mathcal{F}_r| \leq N(r)$.  

8
Fixing some $r_0 > 0$, note that in each rounds $t$ when a ball of radius $< r_0$ was selected, regret is $\Delta(x_t, y_t) \leq O(r_0)$, so the total regret from all such rounds is at most $O(r_0 T)$. Therefore, contextual regret can be written as follows:

$$R(T) = \sum_{t=1}^{T} \Delta(x_t, y_t) = O(r_0 T) + \sum_{r \geq r_0} \sum_{B \in \mathcal{F}_r} \sum_{t \in S_B} \Delta(x_t, y_t)$$

$$\leq O(r_0 T) + \sum_{r} \sum_{B \in \mathcal{F}_r} |S_B| O(r)$$

$$\leq O \left( r_0 T + \sum_{r \geq r_0} \frac{1}{r} N(r) \log(T) \right),$$

which proves the first inequality in (5). The second one is obtained by setting $r_0 = T^{1/(d+2)}$. \qed

### 3.1 Improved guarantees for the contextual zooming algorithm

Let us develop provable guarantees for contextual zooming algorithm that take into account both the expected payoffs and the context arrivals $x_{(1:T)}$. Instead of the covering numbers used in Theorem 3.2 and Theorem 5.1, we will use the dual packing numbers. An $r$-packing of a set $S$ in a metric space is a subset $S' \subset S$ such that any two points in $S'$ are distance $> r$ apart. Note that the size of any $r$-packing of $S$ is bounded from above by the $r$-covering number of $S$. We will consider the $r$-packings of the set

$$\mathcal{P}_r \triangleq \{(x, y) \in \mathcal{P} : \Delta(x, y) \leq 18 r\},$$

same set is in Definition 3.1. To take into account $x_{(1:T)}$, we will count only the $r$-packings of $\mathcal{P}_r$ that are “plausible” given $x_{(1:T)}$.

**Definition 3.7.** A context $x$ relevant to a ball $B$ if there exists an arm $y$ such that $(x, y) \in B$. A set $S \subset \mathcal{P}$ is plausible w.r.t. $x_{(1:T)}$ if each point $p \in S$ can be mapped to a subset $\phi_p \subset x_{(1:T)}$ of at least $r^{-2}$ context arrivals that are relevant to $B(p, r)$ so that the sets $\{\phi_p : p \in S\}$ are disjoint. The $(r, \mu)$-packing number is the largest size of an $r$-packing of $\mathcal{P}_r$ that is plausible w.r.t. $x_{(1:T)}$.

The idea is that if $S$ is an $r$-packing of $\mathcal{P}_r$ that is not plausible then we can rule out an execution of contextual zooming algorithm such that each ball $B(p, r)$, $p \in S$ is activated and then deactivated. Now, it is easy to modify the analysis in Section 3 so that instead of counting the balls of radius $r$ that are activated by the algorithm, we count the balls of radius $r$ that are activated and deactivated. This suffices because each active ball is a child of some ball that have been deactivated, and we bound the number of children using the doubling constant of the similarity space. We obtain the following theorem.

**Theorem 3.8.** Consider the contextual MAB problem with time-invariant payoffs. Then, letting $N(r)$ be the $(r, \mu)$-packing number of the problem instance, the contextual zooming algorithm achieves regret

$$R(T) \leq O(c_{\text{DBL}}) \inf_{r_0 > 0} \left( r_0 T + \sum_{r=2^{-t} : r_0 \leq r \leq 1} \frac{1}{r} N(r) \log(T) \right),$$

where $c_{\text{DBL}}$ is the doubling constant of the similarity space.

It is easy to see that the $(r, \mu)$-packing number from Theorem 3.8 is bounded from above by the $r$-zooming number from Theorem 3.2, so the former theorem provides stronger guarantees. Just like in Theorem 3.2 one can define a version of the contextual zooming dimension based on the $(r, \mu)$-packing numbers, call it $d$, and show that $R(T) \leq O(c_{\text{DBL}} T^{1-1/(2+d)} \log T)$.


3.2 Application to the drifting MAB problem

Let us consider the drifting MAB problem. Recall that this is simply the contextual MAB problem in which the t-th context arrival is \( x_t = t \), the quantity \( \mu(t, y) \) is the expected payoff of arm \( y \) at time \( t \), and the context space \( (X, D_X) \) represents the bound on temporal change. Contextual regret becomes dynamic regret – regret with respect to a benchmark which in each round plays the best arm for this round. Typically, the covering numbers of the context space are proportional to the time horizon \( T \), so the quantity of interest is the average dynamic regret \( \bar{R}(T) \triangleq R(T)/T \).

The general provable guarantees are given in Theorem 3.2 in terms of the \( r \)-zooming numbers, and the construction in Theorem 4.1 provides the matching lower bound, up to polylog(\( T \)) factors (we omit the easy details). For some specific numeric examples, suppose \( D_X(t, t') \triangleq \sigma \sqrt{|t - t'|} \) and consider the worst-case dynamic regret for this context space. Then the \( r \)-covering number of the context space is \( O(\sigma/r)^2 \).

If the arms space consists of \( k \) arms with no similarity information, then from \( 5 \) it is easy to see that \( \bar{R}(T) \leq O(k \sigma^2 \log T)^{1/4} \) as long as \( T \geq T_0 \), where \( T_0 = \Omega(\sqrt{\sigma/k})^{-1} \).

More generally, if \( d_Y \) is the covering dimension of the arms space, with constant \( C_Y \), then \( \bar{R}(T) \leq O(C_Y \sigma^2 \log T)^{1/(4 + d_Y)} \) as long as \( T \geq \Omega(\sigma \sqrt{C_Y})^{-1} \). Similar bounds can be obtained, e.g., for \( D_X(t, t') \triangleq \sigma |t - t'| \). For all these examples, the construction in Theorem 4.1 provides the matching lower bounds up to polylog(\( T \)) factors.

4 Lower bounds

Let us formulate the lower bounds which essentially match the upper bound in Theorem 3.2.

**Theorem 4.1.** Consider the contextual MAB problem with time-invariant payoffs. Fix the context space and the arms space. Let \( N_X(r) \) and \( N_Y(r) \) be their respective \( r \)-covering numbers. Then for any \( r > 0 \) and any \( N \leq N_X(r) N_Y(r) \) there exists a family of \( \Theta(N) \) problem instances such that any algorithm has regret \( R(N/r^2) \geq \Omega(N/r) \) on at least a half of these instances. If the context space and the arms space have doubling constant \( \leq c_{DBL} \), then an \( \Omega(r) \)-zooming number of each problem instance is at most \( c_{DBL}^O(1) N \).

**Remark.** For time horizon \( T = N/r^2 \) the lower bound in Theorem 4.1 matches the upper bound in Theorem 3.2 up to \( O(\log T) \) factors (i) for every problem instance if the context space and the arms space are doubling, (ii) for a “worst-case” problem instance whose \( r \)-zooming number equals \( N_X(r) N_Y(r) \). Recall that the \( r \)-zooming number of each problem instance in Theorem 4.1 is trivially at most \( N_X(r) N_Y(r) \), the \( r \)-covering number of the similarity space \( (\mathcal{P}, \mathcal{D}) = (X \times Y, D_X + D_Y) \).

**Proof Sketch.** For simplicity, let \( N = n_X n_Y \), where \( 1 \leq n_X \leq N_X(r) \) and \( 2 \leq n_Y \leq N_Y(r) \). By definition of the \( r \)-covering number, any \( r \)-net on the context space has size at least \( N_X(r) \). Let \( S_X \) be an arbitrary set of \( n_X \) points from one such \( r \)-net. Similarly, let \( S_Y \) be an arbitrary set of \( n_Y \) points from some \( r \)-net on the arms space. The sequence \( x(1 : T) \) of context arrivals is defined in an arbitrary round-robin fashion over the points in \( S_X \). For each \( x \in S_X \) we construct a standard needle-in-the-haystack lower-bounding example (from [3]) on the set \( S_Y \). Specifically, pick one point \( y^*(x) \in S_Y \) (but the algorithm does not know which), and define \( \mu(x, y^*(x)) = \frac{1}{2} + \frac{C_Y}{4} \) and \( \mu(x, y) = \frac{1}{2} + \frac{C_Y}{4} \) for each \( y \in S_Y \setminus \{ y^*(x) \} \). We smoothen the expected payoffs so that for the context-arms pairs that are far away from \( S_X \times S_Y \) the expected payoff is \( \frac{1}{2} \), and the Lipschitz condition (1) holds:

\[
\mu(x, y) \triangleq \max_{x_0 \in S_X, y_0 \in S_Y} \max \left( \frac{1}{2}, \mu(x_0, y_0) - D_X(x, x_0) - D_Y(y, y_0) \right).
\]

It is easy to see that on such problem instance, all points of badness at most \( \frac{C_Y}{4} \) can be covered by \( N \) balls of radius \( \frac{C_Y}{4} \), hence the claimed bound on the \( r' \)-zooming number for a suitable \( r' = \Omega(r) \). Using the

\(^7\)Note that one can replace the \( \log(T) \) factor in dynamic regret by a \( \log(k \sigma) \) factor by restarting the algorithm every \( T_0 \) rounds.
Consider the contextual MAB problem with adversarial payoffs and \( k \) is at most \( \{ r, k \} \) is a \( r \)-convex payoff set in the context space, where \( r \) is the doubling constant of the set \( \mathcal{X} \). Recall that we maintain a partition of the context space that is adapted to the context arrivals. ContextualBandit with a convex subset of \( \mathcal{X} \) contributes a regret guarantee for each arm \( y \in \mathcal{Y} \) at least \( \Omega(r^{-2}) \) times, incurring regret \( \Omega(r) \) for each try of each \( y \neq y^*(x) \).

5 Adversarial payoffs: the contextual meta-algorithm

In this section we consider the contextual MAB problem with adversarial payoffs. We describe an algorithm that is geared to take advantage of “benign” (“low-dimensional”) sequence of context arrivals. It is in fact a meta-algorithm: given an adversarial bandit algorithm Bandit, we present a contextual bandit algorithm ContextualBandit which calls Bandit as a subroutine. The crucial feature of our algorithm is that it maintains a partition of the context space that is adapted to the context arrivals.

We consider the contextual MAB problem with a known support \(( \mathcal{X}, \mathcal{F} \)). Our algorithm is parameterized by a regret guarantee for Bandit on instances with support \(( \mathcal{Y}, \mathcal{F} \)). For a more concrete theorem statement, we will assume that the convergence time \( T_0(r) \) of Bandit is at most \( T_0(r) \lesssim c_Y r^{-(2+2d_Y)} \log(\frac{1}{\delta}) \) for some constants \( c_Y \) and \( d_Y \) that are known to the algorithm. In particular, an algorithm that (essentially) can be found in [20] achieves this guarantee if \( d_Y \) is the \( c \)-covering dimension of the arms space and \( c_Y = \tilde{O}(r^{2+d_Y}) \).

We will consider the context space \(( \mathcal{X}, D_X \))—this is the only metric space that appears in the rest of this section. The term “ball” will refer to a ball in the context space. Recall that \( \mathcal{X}(1..T) \) denotes the sequence (or a multiset) of the first \( T \) context arrivals \(( x_1, \ldots, x_T \)). Let \( c_{DBL} = c_{DBL}(\mathcal{X}(1..T), D_X) \).

We quantify the “goodness” of the context sequence in terms of the covering number of \( \mathcal{X}(1..T) \) in the context space, where \( T \) is the time horizon. In fact, we will use a more permissive notion that allows a limited number of “outliers”: the \( (r, k) \)-covering number of a multiset \( S \) is the \( r \)-covering number of the set \( \{ x \in S : |B(x, r) \cap S| \geq k \} \). One can naturally define a version of the covering dimension that uses the \( (r, k) \)-covering numbers rather than the \( r \)-covering numbers. Specifically, given a constant \( c \) and a function \( k : (0, 1) \to \mathbb{N} \), the relaxed covering dimension of \( S \) with slack \( k(\cdot) \) is the smallest \( d > 0 \) such that the \( (r, k(r)) \)-covering number of \( S \) is at most \( cr^{-\delta} \) for all \( r > 0 \).

Theorem 5.1. Consider the contextual MAB problem with adversarial payoffs and \( \mathcal{F} \). Let Bandit be an algorithm for the adversarial MAB problem whose convergence time on problem instances with support \( \mathcal{F} \) is at most \( T_0(r) \lesssim c_Y r^{-(2+2d_Y)} \log(\frac{1}{\delta}) \) for some constants \( c_Y \) and \( d_Y \) that are known to the algorithm. Then ContextualBandit achieves contextual regret, for any time \( T \) and \( c_X \),

\[
R(T) \leq O(c_{DBL}^2 (c_X n_Y)^{1/(2+d_X+d_Y)} T^{1-1/(2+d_X+d_Y)} \log T),
\]

where \( c_{DBL} \) is the doubling constant of \( \mathcal{X}(1..T) \), and \( d_X \) is the relaxed covering dimension of \( \mathcal{X}(1..T) \) with slack \( T_0(\cdot) \) and constant \( c_X \).

Remark. The guarantee (13) should be contrasted with the one for the “naive” algorithm (2): the difference is that (13) uses a more “permissive” notion of dimensionality for the context space. For a version of (13) that is stated in terms of the “raw” \( (r, k(r)) \)-covering numbers of \( \mathcal{X}(1..T) \), see (15) in the analysis (page 13).

Remark. For a bounded number of arms, algorithm EXP3 [4] achieves \( d_Y = 0 \) and \( c_Y = O(\sqrt{\mathcal{Y}}) \). For linear payoffs — problem instances whose support \( \mathcal{F} \) consists of linear functions on \( \mathcal{Y} \), where \( \mathcal{Y} \) is identified with a convex subset of \( \mathbb{R}^d \) — there exist algorithms with \( d_Y = 0 \) and \( c_Y = \text{poly}(d) \) [14, 1]. Likewise, for convex payoffs there exist algorithms with \( d_Y = 2 \) and \( c_Y = O(d) \) [15].

---

8The \( r \)-convergence time \( T_0(r) \) is the smallest \( T_0 \) such that regret is \( R(T) \leq rT \) for each \( T \geq T_0 \).

9By abuse of notation, here \( |B(x, r) \cap S| \) denotes the number of points \( x \in S \), with multiplicities, that lie in \( B(x, r) \).
The algorithm. The contextual bandit algorithm \texttt{ContextualBandit} is parameterized by an adversarial MAB algorithm \texttt{Bandit}, which it uses as a subroutine, and a function \( T_0(\cdot) : (0, 1) \rightarrow \mathbb{N} \).

The algorithm maintains a finite collection of balls, called active balls. A ball stays active once it is activated. Initially there is one active ball of radius 1 which contains the entire context space. Some active balls will be designated full, as defined below.

In each round \( t \), one ball may be activated according to the following activation rule. Suppose all active balls that contain \( x_t \) are full. Among these balls, pick the one with the smallest radius (break ties arbitrarily), call it \( B \). Activate the ball \( B(x_t, \frac{r}{2}) \), where \( r \) is the radius of \( B \), and call it a child of \( B \).

Let us say that a ball \( B \) is hit in round \( t \) if (after the activation rule is called) it is active, not full, contains \( x_t \), and among all such balls it has been activated last (any other tie-breaking rule can be used as well).

A ball of radius \( r \) is full if it is active and it has been hit \( T_0(r) \) times.

Once ball \( B \) is activated, we initialize a fresh instance \( (A_B) \) of algorithm \texttt{Bandit} whose “arms” are indexed by points in \( Y \). In each round, if \( B \) is the ball that is hit, algorithm \( A_B \) is called to select an arm \( y \in Y \) to be played, and the resulting payoff is fed back to \( A_B \). This completes the specification.

\textbf{Claim 5.2.} The algorithm satisfies the following basic properties:

(a) (Correctness) In each round \( t \), exactly one active ball is hit.

(b) Each active ball of radius \( r \) is hit at most \( T_0(r) \) times.

(c) (Separation) For any two active balls \( B(x, r) \) and \( B(x', r) \) we have \( D_X(x, x') > r \).

(d) Each active ball has at most \( c_{\text{DBL}}^2 \) children, where \( c_{\text{DBL}} \) is the doubling constant of \( X \).

\textbf{Proof.} For (a), it suffices to show that after the activation rule is called, there exists an active non-full ball containing \( x_t \). For the sake of contradiction, suppose such ball does not exist. Then it was the case even before the activation rule has been called. Thus, the activation rule did activate some ball \( B \) centered at \( x_t \), as a child of some other ball \( B' \) of twice the radius. The radius of \( B' \) must be the smallest among all balls that in round \( t \) are full and contain \( x_t \). It follows that \( B \) is not full in round \( t \) (and therefore not active).

For (b), simply note that by the algorithms’ specification a ball is hit only when it is not full.

To prove (c), suppose that \( D_X(x, x') \leq r \) and suppose \( B(x', r) \) is activated in some round \( t \) while \( B(x, r) \) is active. Then \( B(x', r) \) was activated as a child of some ball \( B^* \) of radius \( 2r \). On the other hand, \( x' = x_t \in B(x, r) \), so \( B(x, r) \) must have been full in round \( t \) (else no ball would have been activated), and consequently the radius of \( B^* \) is at most \( r \). Contradiction.

For (d), consider the children of a given active ball \( B(x, r) \). Note that by the activation rule the centers of these children are points in \( x_{(1..T)} \cap B(x, r) \), and by the separation property any two of these points lie within distance \( > \frac{r}{2} \) from one another. By the doubling property, there can be at most \( c_{\text{DBL}}^2 \) such points. \( \Box \)

In the remainder of the section we prove Theorem 5.1. Let us fix the time horizon \( T \), and let \( R(T) \) denote the contextual regret of \texttt{ContextualBandit}. Partition \( R(T) \) into the contributions of active balls as follows. Let \( B \) be the set of all balls that are active after round \( T \). For each \( B \in B \), let \( S_B \) be the set of all rounds \( t \) when \( B \) has been hit. Then

\[ R(T) = \sum_{B \in B} R_B(T), \quad \text{where} \quad R_B(T) \triangleq \sum_{t \in S_B} \mu_t^*(x_t) - \mu_t(x_t, y_t). \]

\textbf{Claim 5.3.} For each ball \( B = B(x, r) \in B \), we have \( R_B \leq 3 r T_0(r) \).

\textbf{Proof.} By the Lipschitz conditions on \( \mu_t \) and \( \mu_t^* \), for each round \( t \in S_B \) it is the case that

\[ \mu_t^*(x_t) \leq r + \mu_t^*(x) = r + \mu_t(x, y^*(x)) \leq 2rn + \mu_t(x_t, y^*(x)). \]
Therefore the number of convergence time corresponding set of leaves (i.e., arms). The random payoffs. Let
Consider the (context-free) MAB problem with time-invariant
not only the corresponding expected payoff, but also information about the implicit distances. We obtain the distance between two arms is equal to the maximal variation in payoffs in the least common subtree. We obtain
arms. A tree-shaped taxonomy on arms implicitly defines a similarity distance as follows: the distance
In this section we consider a setting where the similarity information is given by a topical taxonomy on
6 Implicit similarity information: the taxonomy MAB problem
Proof. Fix $r$ and let $k = T_0(r)$. Let us say that a point $x \in x_{(1,T)}$ is heavy if $B(x,r)$ contains at least $k$ points of $x_{(1,T)}$, counting multiplicities. Clearly, $B(x,r)$ is full only if its center is heavy. By definition of the $(r,k)$-covering number, there exists a family $\mathcal{S}$ of $N(r,k)$ sets of diameter $\leq r$ that cover all heavy points in $x_{(1,T)}$. For each full ball $B = B(x,r)$, let $S_B$ be some set in $\mathcal{S}$ that contains $x$. By Claim 5.2(c), the sets $S_B$, $B \in \mathcal{F}_r$ are all distinct. Thus, $|\mathcal{F}_r| = |\mathcal{S}| \leq N(r,k)$. □
Proof of Theorem 5.1. Let $\mathcal{B}_r$ be the set of all balls of radius $r$ that are active after round $T$. By the algorithm’s specification, each ball in $\mathcal{F}_r$ has been hit $T_0(r)$ times, so $|\mathcal{F}_r| \leq T/T_0(r)$. Then using Claim 5.2(b) and Claim 5.4, we have
$$|\mathcal{B}_{r/2}| \leq c_{\text{DBL}}^2 |\mathcal{F}_r| \leq c_{\text{DBL}}^2 \min(T/T_0(r), N(r,T_0(r)))$$
$$\sum_{B \in \mathcal{B}_{r/2}} R_B \leq O(r) T_0(r) |\mathcal{B}_{r/2}| \leq O(c_{\text{DBL}}^2) \min(r T, r T_0(r)) N(r,T_0(r)).$$
(14)
Trivially, for any full ball of radius $r$ we have $T_0(r) \leq T$. Thus, summing (14) over all such $r$, we obtain
$$R(T) \leq O(c_{\text{DBL}}^2) \sum_{r=2^{-t}; t \in \mathbb{N} \text{ and } T_0(r) \leq T} \min(r T, r T_0(r)) N(r,T_0(r)).$$
(15)
Note that (15) makes no assumptions on $N(r,T_0(r))$. Now, plugging in $T_0(r) = c_X r^{-(2+d_X)}$ and $N(r,T_0(r)) \leq c_X r^{-d_X}$ into (15) and optimizing it for $r$ it is easy to derive the desired bound (13). □
6 Implicit similarity information: the taxonomy MAB problem
In this section we consider a setting where the similarity information is given by a topical taxonomy on arms. A tree-shaped taxonomy on arms implicitly defines a similarity distance as follows: the distance between two arms is equal to the maximal variation in payoffs in the least common subtree. We obtain guarantees similar to those for the Lipschitz MAB problem [24] in which the induced similarity metric is explicitly revealed to the algorithm. On a technical level, the crucial issue is the more complex version of the exploration-exploitation trade-off: taking sufficiently many random samples from a given subtree reveals not only the corresponding expected payoff, but also information about the implicit distances.

The setting: taxonomy MAB problem. Consider the (context-free) MAB problem with time-invariant random payoffs. Let $X$ be the set of arms, and let $\mu : X \rightarrow [0,1]$ be the expected payoffs: $\mu(x)$ is the expected payoff of arm $x$. An algorithm is given a taxonomy $T$ on $X$, which is simply a rooted tree whose leaf set is $X$. For an internal node $v \in T$ let $T(v)$ be the subtree rooted at $v$, and let $X(v)$ be the corresponding set of leaves (i.e., arms). The weight of $v$ is defined as $\text{wgt}(v) = \sup_{x,y \in X(v)} |\mu(x) - \mu(y)|$. The taxonomy is seen as an implicit representation of the following natural similarity distance $(X, D)$: for
any $x, y \in X$ we define $D(x, y)$ as the weight of their least common ancestor. Note that $\mu$ is a Lipschitz function on $(X, D)$.

The quality of the taxonomy is expressed as follows. Consider the random walk on a directed version of $T$ oriented away from the root. Let $P(v, u)$ be the probability that this random walk reaches a node $u$ starting from $v$. A sample from this distribution is called a random sample from $T(v)$. Define the expected payoff from the subtree $T(v)$ as that of the random sample: $\mu(v) = \sum_{x \in X(v)} \mu(x) P(v, x)$. Then the quality of the taxonomy is the largest number $q$ such that the following condition holds:

- for each subtree $T(v)$ containing an optimal strategy, there exist internal nodes $u, u' \in T(v)$ such that $|\mu(u) - \mu(u')| \geq \text{wgt}(v)/2$, and moreover $\min( P(v, u), P(v, u') ) \geq q$.

**Remark.** Here we have a more complex exploration-exploitation trade-off: taking sufficiently many random samples from a given subtree $T(v)$ reveals not only its expected payoff $\mu(v)$ but also its weight $\text{wgt}(v)$. Accordingly, both payoffs and weights are subject to the trade-off.

**Provable guarantees.** Let us define the zooming dimension of the problem instance (with constant $c$) as that of the corresponding instance $(X, D, \mu)$ of the Lipschitz MAB problem, as defined in [24]. In the present setting, it is the smallest number $d$ such that the following covering property holds for each $\delta > 0$:

- the set $X_{\delta} = \{ x \in X : \mu(x) - \mu(x) \leq \delta \}$ can be covered with $c \delta^{-d}$ subtrees whose weight is $\leq \delta/8$.

Recall that it is upper-bounded by the covering dimension, but takes into account the expected payoff function. In particular, it is not impacted by a subtree with high covering dimension but low payoffs.

We provide an algorithm TaxonomyBandit which, for a large enough time horizon, matches the guarantee in [24] for the corresponding instance $(X, D, \mu)$ of the Lipschitz MAB problem up to polylog($T$) factors.

**Theorem 6.1.** Consider the taxonomy MAB problem on a constant-degree taxonomy. Then for any $c > 0$ algorithm TaxonomyBandit achieves regret $R(T) \leq O(\frac{\delta}{q})^{1/(2+\delta)} T^{1-1/(2+\delta)} \log(T)$ for any $T \geq 2^{1/q}$, where $d$ is the zooming dimension with constant $c$, and $q$ is the quality of the taxonomy.

**Algorithm TaxonomyBandit.** The algorithm is parameterized by the time horizon $T$ and a quality parameter $\hat{q}$, a pessimistic (lower) bound on the quality of the taxonomy\(^{[10]}\). In each round the algorithm selects one of the tree nodes, say $v$, and plays a randomly sampled strategy $x$ from $T(v)$. We say that a subtree $T(u)$ is hit in this round if $u \in T(v)$ and $x \in T(u)$. For each tree node $v \in V$ and time $t$, let $n_t(v)$ be the number of times the subtree $T(v)$ has been hit by the algorithm before time $t$, and let $\mu_t(v)$ be the corresponding average reward. Note that $E[\mu_t(v)] = \mu(v)$ if $n_t(v) > 0$. Define the confidence radius of $v$ at time $t$ as

$$\text{rad}_t(v) \triangleq \sqrt{8 \log(T) / (2 + n_t(v))}.$$  \hspace{1cm} (16)

The meaning of the confidence radius is that $|\mu_t(v) - \mu(v)| \leq \text{rad}_t(v)$ with high probability.

For each tree node $v$ and time $t$, let us define the index of $v$ at time $t$ as

$$I_t(v) \triangleq \mu_t(v) + (1 + 2 k_A) \text{rad}_t(v), \quad \text{where} \quad k_A \triangleq 4 \sqrt{2/\hat{q}}.$$ \hspace{1cm} (17)

Here we posit $\mu_t(v) = 0$ if $n_t(v) = 0$. Let us define the weight estimate

$$\text{wgt}_t(v) = \max_{u_1, u_2 \in T(v)} \max( 0, |\mu_t(u_1) - \mu_t(u_2)| - \text{rad}_t(u_1) - \text{rad}_t(u_2) ).$$ \hspace{1cm} (18)

\(^{[10]}\)The dependency on $T$ can be removed via the doubling trick, as described in the Preliminaries. The dependency on $\hat{q}$ can be removed as follows: in each phase $i$ run a fresh instance of TaxonomyBandit for $2^i$ steps, with $\hat{q} = 1/i$. 

14
Throughout the phase, some tree nodes are designated active. We maintain the following invariant:

\[ \text{wgt}_t(v) < k_A \text{rad}_t(v) \] for each active internal node \( v \).

(19)

**Analysis.** We identify a certain high-probability behavior of the system, and argue deterministically under assumption that the system follows this behavior.

**Definition 6.2.** An execution of TaxonomyBandit is called clean if for each time \( t \leq T \) and each tree node \( v \in V \) the following two properties hold:

(P1) \(|\mu_t(v) - \mu(v)| \leq \text{rad}_t(v)\) as long as \( n_t(v) > 0 \).

(P2) for each tree node \( u \in T(v) \) we have

\[ n_t(v) \mathcal{P}(v, u) \geq 8 \log T \implies n_t(u) \geq \frac{1}{2} n_t(v) \mathcal{P}_T(v, u). \]

**Claim 6.3.** An execution of TaxonomyBandit is clean with probability at least \( 1 - T^{-2} \).

From now on we will argue about a clean execution. Note that in a clean execution, \( \text{wgt}_t(v) \geq \text{wgt}_t(v) \).

Let \( \mu^* = \max_{x \in X} \mu(x) \) be the maximal reward. A strategy \( x \in X \) is optimal if \( \mu(x) = \mu^* \). For each tree node \( v \), define the badness of \( v \) as \( \Delta(v) = \mu^* - \mu(v) \). Note that

\[ \mu(v) \leq \mu(u) + \text{wgt}(v) \] for any tree node \( u \in T(v) \).

(20)

The crux of the proof is that at all times the maximal index is at least \( \mu^* \).

**Lemma 6.4.** Consider a clean execution of algorithm \( \mathcal{A}(T) \). Then the following property holds:

- in any round \( t \leq T \), at any point in the execution such that the invariant (19) holds, there exists an active tree node \( v^* \) such that \( I_t(v) \geq \mu^* \).

**Proof.** Fix any optimal strategy \( x^* \), and any tree node \( v^* \) such that \( x^* \in T(v^*) \). Let us denote \( \Delta = \text{wgt}(v^*) \).

By definition of the parameter \( \tilde{q} \), there exist internal nodes \( v_0, v_1 \in T(v^*) \) such that \( \mathcal{P}(v^*, v_j) \geq \tilde{q}, j \in \{0, 1\} \), and moreover \( |\mu(v_1) - \mu(v_0)| \geq \Delta/2 \).

Let us define \( f(\Delta) = 8^3 \log(T) \Delta^{-2} \). Then for each tree node \( v \) and each \( \Delta > 0 \) we have

\[ \text{rad}_t(v) \leq \Delta/8 \iff n_t(v) \geq f(\Delta). \]

Now, for the sake of contradiction let us suppose that \( n_t(v^*) \geq \frac{1}{4} k_A 2 f(\Delta) \). This is equivalent to \( \Delta \geq 2 k_A \text{rad}_t(v^*) \). Note that \( n_t(v^*) \geq (2/\tilde{q}) f(\Delta) \) by our assumption on \( k_A \), so by property (P2) in the definition of the clean execution, for each node \( v_j, j \in \{0, 1\} \) we have \( n_t(v_j) \geq f(\Delta) \), which implies \( \text{rad}_t(v_j) \leq \Delta/8 \). Therefore (18) gives a good estimate of \( \text{wgt}(v^*) \), namely \( \text{wgt}_t(v^*) \geq \Delta/4 \). It follows that \( \text{wgt}_t(v^*) \geq k_A \text{rad}_t(v^*) \), which violates the invariant (19).

We proved that \( \text{wgt}(v^*) \leq 2 k_A \text{rad}_t(v^*) \). Finally, using (20) we have

\[ \Delta(v^*) \leq \text{wgt}(v^*) < 2 k_A \text{rad}_t(v^*) \]

\[ I_t(v^*) \geq \mu(v^*) + 2 k_A \text{rad}_t(v^*) \geq \mu^*, \]

(21)

where the first inequality in (21) holds by definition of the index (17) and by property (P1) in the definition of the clean execution. \( \square \)
We use Lemma 6.4 to show that, essentially, there are not too many strategies with large badness $\Delta(\cdot)$, and each strategy is not played too often. For each tree node $v$, let $N(v)$ be the number of times node $v$ was selected in step (S2) of the algorithm. Call $v$ positive if $N(v) > 0$. We partition all positive tree nodes and all deactivated tree nodes into sets

\[ S_i = \{ \text{tree nodes } v : N(v) > 0 \text{ and } 2^{-i} < \Delta(v) \leq 2^{-i+1} \}, \]

\[ S^*_i = \{ \text{deactivated tree nodes } v : 2^{-i} < 4 \text{wgt}(v) \leq 2^{-i+1} \}. \]

**Lemma 6.5.** Consider a clean execution of algorithm $A(T)$.

(a) for each tree node $v$ we have $N(v) \leq O(k_A^2 \log T) \Delta^{-2}(v)$.

(b) if node $v$ is de-activated at some point in the execution, then $\Delta(v) \leq 4 \text{wgt}(v)$.

(c) $|S^*_i| \leq K_i \triangleq c 2^{(i+1)\dim(v)}$ for each $i$.

(d) $|S_i| \leq O(d_T K_{i+1})$ for each $i$.

**Proof.** For part (a), fix an arbitrary tree node $v$ and let $t$ be the last time $v$ was selected in step (S2) of the algorithm. By part (a), at that point in the execution there was a tree node $v^*$ such that $I_t(v^*) \geq \mu^*$. Then using the selection rule (step (S2)) in the algorithm and the definition of index (17), we have

\[ \mu^* \leq I_t(v^*) \leq I_t(v) \leq \mu(v) + (2 + 2k_A) \text{rad}_t(v), \]

\[ \Delta(v) \leq (2 + 2k_A) \text{rad}_t(v). \]

\[ N(v) \leq n_t(v) \leq O(k_A^2 \log T) \Delta^{-2}(v). \]  \tag{22}

For part (b), suppose tree node $v$ was de-activated at time $s$. Let $t$ be the last round in which $v$ was selected. Then

\[ \text{wgt}(v) \geq \text{wgt}_s(v) \geq k_A r_s(v) \geq \frac{1}{3} (2 + 2k_A) \text{rad}_t(v) \geq \frac{1}{3} \Delta(v). \]  \tag{23}

Indeed, the first inequality in (23) holds since we are in a clean execution, the second inequality in (23) holds because $v$ was de-activated, the third inequality holds because $n_s(v) = n_t(v) + 1$, and the last inequality in (23) holds by (22).

For part (c), let us fix $i$ and define $X_i = \{ x \in X : \Delta(x) \leq 2^{-i+1} \}$. By definition of the $c$-zooming dimension, this set can be covered by $K = c 2^{(i+1)\dim(c)}$ subtrees $T(v_1), \ldots, T(v_K)$, each of weight $< 2^{-i}/4$. Fix a deactivated tree node $v \in S^*_i$. For each strategy $x \in X$ in subtree $T(v)$ we have, by part (b),

\[ \Delta(x) \leq \Delta(v) + \text{wgt}(v) \leq 4 \text{wgt}(v) \leq 2^{-i+1}, \]

so $x \in X_i$ and therefore is contained in some $T(v)$. Note that $v_j \in T(v)$ since $\text{wgt}(v) > \text{wgt}(v_j)$. It follows that the subtrees $T(v_1), \ldots, T(v_K)$ cover the leaf set of $T(v)$.

Consider the graph $G$ on the node set $S^*_i \cup \{v_1, \ldots, v_K\}$, where two nodes $u, v$ are connected by a directed edge $(u, v)$ if there is a path from $u$ to $v$ in the tree $T$. This is a directed forest of out-degree at least 2, whose leaf set is a subset of $\{v_1, \ldots, v_K\}$. Since in any directed tree of out-degree $\geq 2$ the number of nodes is at most twice the number of leaves, $G$ contains at most $K$ internal nodes. Therefore, $|S^*_i| \leq K$, proving part (c).

For part (d), let us fix $i$ and consider a positive tree node $u \in S_i$. Since $N(u) > 0$, either $u$ is active at time $T$, or it was deactivated in some round before $T$. In the former case, let $v$ be the parent of $u$. In the latter case, let $v = u$. Then by part (b) we have $2^{-i} \leq \Delta(u) \leq \Delta(v) + \text{wgt}(v) \leq 4 \text{wgt}(v)$, so $v \in S^*_j$ for some $j \leq i + 1$.

For each tree node $v$, define its family as the set which consists of $u$ itself and all its children. We have proved that each positive node $u \in S_i$ belongs to the family of some deactivated node $v \in \bigcup_{j=1}^{i+1} S^*_j$. Since each family consists of at most $1 + d_T$ nodes, it follows that

\[ |S_i| \leq (1 + d_T) \left( \sum_{j=1}^{i+1} K_j \right) \leq O(d_T K_{i+1}). \]
Now we can bound the regret of TaxonomyBandit.

**Theorem 6.6.** Consider the taxonomy MAB problem on a taxonomy of degree \(d_T\). Then algorithm TaxonomyBandit, parameterized by time horizon \(T\) and the quality parameter \(\hat{q}\), achieves regret (for any \(c > 0\))

\[
R_A(T) \leq O\left(\frac{c d_T}{\hat{q}}\right)^{1/(2+d)} T^{1-1/(2+d)} \sqrt{\log T}
\]

(24)

where \(d\) is the zooming dimension with constant \(c\).

**Proof.** The theorem follows from parts (a) and (d) of Lemma 6.5. Let us fix \(c > 0\) and let \(d\) be the \(c\)-zooming dimension of the problem instance. Let us assume a clean execution. (Recall that by Claim 6.3 the failure probability is negligibly small.) Then:

\[
\sum_{v \in S_i} N(v) \Delta(v) \leq O(k_A^2 \log t) \sum_{v \in S_i} \frac{1}{\Delta(v)} \leq O(k_A^2 \log t) |S_i| 2^i \leq K 2^{(i+2)(1+d)},
\]

where \(K = O(c d_T k_A^2 \log t)\). For any \(\delta = 2^{-i_0}\) we have

\[
R_A(t) \leq \sum_{\text{tree nodes } v} N(v) \Delta(v) = \left(\sum_{v: \Delta(v) < \delta} N(v) \Delta(v)\right) + \left(\sum_{v: \Delta(v) \geq \delta} N(v) \Delta(v)\right)
\]

\[
\leq \delta t + \left(\sum_{i \leq i_0} \sum_{v \in S_i} N(v) \Delta(v)\right) \leq \delta t + \sum_{i \leq i_0} K 2^{(i+2)(1+d)} \leq \delta t + O(K) (\frac{8}{\delta})^{(1+d)}.
\]

We obtain (24) by setting \(\delta = (\frac{K}{t})^{1/(d+2)}\). \(\square\)

**Acknowledgements.** The author is grateful to Ittai Abraham, Bobby Kleinberg and Eli Upfal for many conversations about multi-armed bandits.

**References**

[1] Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the Dark: An Efficient Algorithm for Bandit Linear Optimization. In 21th Conference on Learning Theory (COLT), pages 263–274, 2008.

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

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

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

[5] Peter Auer, Ronald Ortner, and Csaba Szepesvári. Improved Rates for the Stochastic Continuum-Armed Bandit Problem. In 20th Conference on Learning Theory (COLT), pages 454–468, 2007.

[6] Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, February 2008. Preliminary version appeared in STOC 2004.

[7] Jeffrey Banks and Rangarajan Sundaram. Denumerable-armed bandits. *Econometrica*, 60(5):1071–1096, 1992.

[8] 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.

[9] Donald Berry and Bert Fristedt. *Bandit problems: sequential allocation of experiments*. Chapman&Hall, 1985.

[10] Sebastien Bubeck, Remi Munos, Gilles Stoltz, and Csaba Szepesvari. Online Optimization in X-Armed Bandits. In *Advances in Neural Information Processing Systems (NIPS)*, pages 201–208, 2008.

[11] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.
[12] Eric Cope. Regret and convergence bounds for immediate-reward reinforcement learning with continuous action spaces, 2004. Unpublished manuscript.

[13] Varsha Dani and Thomas P. Hayes. Robbing the bandit: less regret in online geometric optimization against an adaptive adversary. In 17th ACM-SIAM Symp. on Discrete Algorithms (SODA), pages 937–943, 2006.

[14] 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.

[15] 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.

[16] J. C. Gittins. Multi-Armed Bandit Allocation Indices. John Wiley & Sons, 1989.

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

[18] Elad Hazan and Nimrod Megiddo. Online Learning with Prior Information. In 20th Conference on Learning Theory (COLT), pages 499–513, 2007.

[19] Sham M. Kakade, Adam T. Kalai, and Katrina Ligett. Playing Games with Approximation Algorithms. In 39th ACM Symp. on Theory of Computing (STOC), 2007.

[20] Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In 18th Advances in Neural Information Processing Systems (NIPS), 2004. Full version appeared in the author’s thesis (MIT, 1995).

[21] Robert Kleinberg. Online Decision Problems with Large Strategy Sets. PhD thesis, MIT, Boston, MA, 2005.

[22] Robert Kleinberg. Anytime algorithms for multi-armed bandit problems. In 17th ACM-SIAM Symp. on Discrete Algorithms (SODA), pages 928–936, 2006. Full version appeared in the author’s thesis (MIT, 1995).

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

[24] 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, 2008.

[25] Levente Kocsis and Csaba Szepesvari. Bandit Based Monte-Carlo Planning. In ECML, pages 282–293, 2006.

[26] T.L. Lai and Herbert Robbins. Asymptotically efficient Adaptive Allocation Rules. Advances in Applied Mathematics, 6:4–22, 1985.

[27] John Langford and Tong Zhang. The Epoch-Greedy Algorithm for Contextual Multi-armed Bandits. In 21st Advances in Neural Information Processing Systems (NIPS), 2007.

[28] H. Brendan McMahan and Avrim Blum. Online Geometric Optimization in the Bandit Setting Against an Adaptive Adversary. In 17th Conference on Learning Theory (COLT), pages 109–123, 2004.

[29] Remi Munos and Pierre-Arnaud Coquelin. Bandit algorithms for tree search. In UAI, 2007.

[30] 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.

[31] Herbert Robbins. Some Aspects of the Sequential Design of Experiments. Bull. Amer. Math. Soc., 58:527–535, 1952.

[32] Chih-Chun Wang, Sanjeev R. Kulkarni, and H. Vincent Poor. Bandit problems with side observations. IEEE Trans. on Automatic Control, 50(3):338-355, 2005.