Top-k eXtreme Contextual Bandits with Arm Hierarchy

Rajat Sen [1] **Alexander Rakhlin** [2 3] **Lexing Ying** [4 3] **Rahul Kidambi** [3] **Dean Foster** [3] **Daniel Hill** [3]
Inderjit S. Dhillon [5 3]

Abstract

Motivated by modern applications, such as online advertisement and recommender systems, we study the top-k eXtreme contextual bandits problem, where the total number of arms can be enormous, and the learner is allowed to select k arms and observe all or some of the rewards for the chosen arms. We first propose an algorithm for the non-eXtreme realizable setting, utilizing the Inverse Gap Weighting strategy for selecting multiple arms. We show that our algorithm has a regret guarantee of $O(k\sqrt{(A-k+1)T\log(|\mathcal{F}|T)})$, where A is the total number of arms and $\mathcal{F}$ is the class containing the regression function, while only requiring $\tilde{O}(A)$ computation per time step. In the eXtreme setting, where the total number of arms can be in the millions, we propose a practically-motivated arm hierarchy model that induces a certain structure in mean rewards to ensure statistical and computational efficiency. The hierarchical structure allows for an exponential reduction in the number of relevant arms for each context, thus resulting in a regret guarantee of $O(k\sqrt{(\log A-k+1)T\log(|\mathcal{F}|T)})$. Finally, we implement our algorithm using a hierarchical linear function class and show superior performance with respect to well-known benchmarks on simulated bandit feedback experiments using eXtreme multi-label classification datasets. On a dataset with three million arms, our reduction scheme has an average inference time of only 7.9 milliseconds, which is a 100x improvement.

[1]Google Research, Mountain View (work done while at Amazon) [2]Massachusetts Institute of Technology, Boston [3]Amazon [4]Stanford University, Palo Alto [5]Department of Computer Science, University of Texas, Austin. Correspondence to: Rajat Sen <rajat.sen@utexas.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

1. Introduction

The *contextual bandit* is a sequential decision-making problem, in which, at every time step, the learner observes a context, chooses one of the A possible actions (arms), and receives a reward for the chosen action. Over the past two decades, this problem has found a wide range of applications, from e-commerce and recommender systems (Yue & Guestrin, 2011; Li et al., 2016) to medical trials (Durand et al., 2018; Villar et al., 2015). The aim of the decision-maker is to minimize the difference in total expected reward collected when compared to an optimal policy, a quantity termed *regret*. As an example, consider an advertisement engine in an online shopping store, where the context can be the user's query, the arms can be the set of millions of sponsored products and the reward can be a click or a purchase. In such a scenario, one must balance between *exploitation* (choosing the best ad (arm) for a query (context) based on current knowledge) and *exploration* (choosing a currently unexplored ad for the context to enable future learning).

The contextual bandits literature can be broadly divided into two categories. The *agnostic* setting (Agarwal et al., 2014; Langford & Zhang, 2007; Beygelzimer et al., 2011; Rakhlin & Sridharan, 2016) is a model-free setting where one competes against the best policy (in terms of expected reward) in a class of policies. On the other hand, in the *realizable* setting it is assumed that a known class $\mathcal{F}$ contains the function mapping contexts to expected rewards. Most of the algorithms in the realizable setting are based on Upper Confidence Bound or Thompson sampling (Filippi et al., 2010; Chu et al., 2011; Krause & Ong, 2011; Agrawal & Goyal, 2013) and require specific parametric assumptions on the function class. Recently there has been exciting progress on contextual bandits in the realizable case with general function classes. Foster & Rakhlin (2020) analyzed a simple algorithm for general function classes that reduced the adversarial contextual bandit problem to online regression, with a minimax optimal regret scaling. The algorithm was then analyzed for i.i.d. contexts using offline regression in (Simchi-Levi & Xu, 2020). The proposed algo-

rithms are general and easily implementable but have two main shortcomings.

First, in many practical settings the task actually involves selecting a small number of arms per time instance rather than a single arm. For instance, in our advertisement example, the website can have multiple slots to display ads and one can observe the clicks received from some or from all the slots. It is not immediately obvious how the techniques in (Simchi-Levi & Xu, 2020; Foster & Rakhlin, 2020) can be extended to selecting k of a total of A arms while avoiding the combinatorial explosion from $\binom{A}{k}$ possibilities. Second, the total number of arms A can be in tens of millions and we need to develop algorithms that only require $o(A)$ computation per time-step and also have a much smaller dependence on the total number of arms in the regret bounds. Therefore, in this paper, we consider the top-k eXtreme contextual bandit problem where the number of arms is potentially enormous and at each time-step one is allowed to select $k \geq 1$ arms.

This extreme setting is both theoretically and practically challenging, due to the sheer size of the arm space. On the theoretical side, most of the existing results on contextual bandit problems address the small arm space case, where the complexity and regret typically scales polynomially (linearly or as square root) in terms of the number of arms (with the notable exception of the case when arms are embedded in a d-dimensional vector space (Foster et al., 2020a)). Such a scaling inevitably results in large complexity and regret in the extreme setting. On the implementation side, most contextual bandit algorithms have not been shown to scale to millions of arms. The goal of this paper is to bridge the gaps both in theory and in practice. We show that the freedom to present more than one arm per time step provides valuable exploration opportunities. Moreover in many applications, for a given context, the rewards from the arms that are correlated to each other but not directly related to the context are often quite similar, while large variations in the reward values are only observed for the arms that are closely related to the context. For instance in the advertisement example, for an electronics query (context) there might be finer variation in rewards among computer accessories related display ads while very little variation in rewards among items in an unrelated category like culinary books. This prior knowledge about the structure of the reward function can be modeled via a judicious choice of the model class $\mathcal{F}$, as we show in this paper.

The **main contributions** of this paper are as follows:

- We define the top-k contextual bandit problem in Section 3.1. We propose a natural modification of the inverse gap weighting (IGW) sampling strategy employed in (Foster & Rakhlin, 2020; Simchi-Levi & Xu, 2020; Abe & Long, 1999) as Algorithm 1. In Section 3.3 we show that our algorithm can achieve a top-k regret bound of $O(k\sqrt{(A-k+1)T\log(|\mathcal{F}|T)})$ where T is the time-horizon. Even though the action space is combinatorial, our algorithm's computational cost for a time-step is $O(A)$ as it can leverage the additive structure in the total reward obtained from a set of arms chosen. We also prove that if the problem setting is only approximately realizable then our algorithm can achieve a regret scaling of $O(k\sqrt{(A-k+1)T\log(|\mathcal{F}|T)} + \epsilon k\sqrt{A-k+1}T)$, where ϵ is a measure of the approximation.
- Inspired by success of tree-based approached for eXtreme output space problems in supervised learning (Prabhu et al., 2018; Yu et al., 2020; Khandagale et al., 2020), in Section 4 we introduce a hierarchical structure on the set of arms to tackle the eXtreme setting. This allows us to propose an eXtreme reduction framework that reduces an extreme contextual bandit problem with A arms (A can be in millions) to an equivalent problem with only $O(\log A)$ arms. Then we show that our regret guarantees from Section 3.3 carry over to this reduced problem.
- We implement our eXtreme contextual bandit algorithm with a hierarchical linear function class and test the performance of different exploration strategies under our framework on eXtreme multi-label datasets (Bhatia et al., 2016) in Section 5, under simulated bandit feedback (Bietti et al., 2018). On the amazon-3m dataset, with around three million arms, our reduction scheme leads to a 100x improvement in inference time over a naively evaluating the estimated reward for every arm given a context. We show that the eXtreme reduction also leads to a 29% improvement in progressive mean rewards collected on the eurlex-4k dataset. More over we show that our exploration scheme has the highest win percentage among the 6 datasets w.r.t the baselines.

2. Related Work

The general contextual bandits problem has been studied both in the agnostic setting (Auer et al., 2002; McMahan & Streeter, 2009; Beygelzimer et al., 2011; Agarwal et al., 2014; Langford & Zhang, 2007) where the mean rewards of the arms are not fully captured by the function class as well as in the realizable setting (Filippi et al., 2010; Chu et al., 2011; Krause & Ong, 2011; Agrawal & Goyal, 2013). Most algorithms in the latter setting are based on Upper Confidence Bound strategies or Thompson Sampling leading to

exploration schemes that depend heavily on the parametric function class. Recently there has been some notable advancement in the realizable setting where the exploration strategy can be independent of the specific function class used while providing optimal regret guarantees (Foster & Rakhlin, 2020; Simchi-Levi & Xu, 2020; Foster et al., 2020b). We build on these techniques and extend them to the top-k problem.

The top-k problem has been studied in contextual bandits under specific assumptions on the function class as well as the value derived from a set of arms (for instance the set function being submodular) (Qin et al., 2014; Yue & Guestrin, 2011). In the context of off-policy learning from logged data there are several works that address the top-k arms selection problem under the context of slate recommendations (Swaminathan et al., 2017; Narita et al., 2019). There is a large body of literature on combinatorial multi-armed bandits and we refer the reader to Appendix A for a more in depth discussion. The problem of learning from logged bandit feedback when the number of arms is extreme was studied recently in (Lopez et al., 2020). In (Majzoubi et al., 2020) the authors address the contextual bandit problem for continuous action spaces by using a cost sensitive classification oracle for large number of classes, which is itself implemented as a hierarchical tree of binary classifiers. Our arm hierarchy model for the eXtreme case is inspired by tree search based models for eXtreme Multi-Label Classification/Ranking (XMC/ XMR) (Bhatia et al., 2016; Jasinska et al., 2016; Prabhu et al., 2018; Khandagale et al., 2020; Wydmuch et al., 2018; You et al., 2019; Yu et al., 2020).

3. Top-k Stochastic Contextual Bandit Under Realizability

In the standard contextual bandit problem, at each round, a context is revealed to the learner, the learner picks a single arm, and the reward for only that arm is revealed. In this section, we will study the top-k version of this problem, i.e. at each round the learner selects k distinct arms, and the total reward corresponds to the sum of the rewards for the subset. As feedback, the learner observes some of the rewards for actions in the chosen subset, and we allow this feedback to be as rich as the rewards for all the k selected arms or as scarce as no feedback at all on the given round.

3.1. The Top-k Problem

Suppose that at each time step $t \in \{1, \ldots, T\}$, the environment generates a context $x_t \in \mathcal{X}$ and rewards $\{r_t(a)\}_{a\in[A]}$ for the A arms. The set of arms will be denoted by $\mathcal{A} = [A] := \{1, 2, \cdots, A\}$. As standard in the stochastic model of contextual bandits, we shall assume that $(x_t, r_t(1), \cdots r_t(A))$ are generated i.i.d. from a fixed but unknown distribution $\mathcal{D}$ at each time step. In this work we will assume for simplicity that $r_t(a) \in [0, 1]$ almost surely for all t and $a \in [A]$. We will work under the realizability assumption (Agarwal et al., 2012; Foster et al., 2018; Foster & Rakhlin, 2020). We also provide some results under approximate realizabilty or the misspecified setting similar to (Foster & Rakhlin, 2020).

Assumption 1 (Realizability). *There exists an $f^* \in \mathcal{F}$ such that, $\mathbb{E}[r_t(a)|X = x] = f^*(x, a)\ \ \forall x \in \mathcal{X}, a \in [A]$, where $\mathcal{F}$ is a class of functions $\mathcal{X} \times \mathcal{A} \to [0, 1]$ known to the decision-maker.*

Assumption 2 (ϵ-Realizability). *There exists an $f^* \in \mathcal{F}$ s.t. $|\mathbb{E}[r_t(a)|X = x] - f^*(x, a)| \leq \epsilon\ \ \forall x \in \mathcal{X}, a \in [A]$.*

We assume that the misspecification level ϵ is known to the learner and refer to (Foster et al., 2020a) for techniques on adapting to this parameter.

Feedback Model and Regret. At the beginning of the time step t, the learner observes the context x_t and then chooses a set of k *distinct* arms $\mathcal{A}_t \subseteq \mathcal{A}$, $|\mathcal{A}_t| = k$. The learner receives feedback for a subset $\Phi_t \subseteq \mathcal{A}_t$, that is, $r_t(a)$ is revealed to the learner for every $a \in \Phi_t$.

Assumption 3. *Conditionally on $x_t, \mathcal{A}_t$ and the history $\mathcal{H}_{t-1}$ up to time $t-1$, the set $\Phi_t \subseteq \mathcal{A}_t$ is random and for any $a \in \mathcal{A}_t$, $\mathbb{P}(a \in \Phi_t | x_t, \mathcal{A}_t, \mathcal{H}_{t-1}) \geq c$ for some $c \in (0, 1]$ which we assume to be known to the learner.*

For the advertisement example, Assumption 3 means that the user providing feedback has at least some non-zero probability $c > 0$ of choosing each of the presented ads, marginally. The choice $c = 1$ corresponds to the most informative case – the learner receives feedback for all the k chosen arms. On the other hand, for $c < 1$ it may happen that no feedback is given on a particular round (for instance, if Φ_t includes each $a \in \mathcal{A}_t$ independently with probability c). When $\mathcal{A}_t$ is a ranked list, behavioral models postulate that the user clicks on an advertisement according to a certain distribution with decreasing probabilities; in this case, c would correspond to the smallest of these probabilities. A more refined analysis of regret bounds in terms of the distribution of Φ_t is beyond the scope of this work.

The total reward obtained in time step t is given by the sum $\sum_{a\in\mathcal{A}_t} r_t(a)$ of all the individual arm rewards in the chosen set, regardless of whether only some of these rewards are revealed to the learner. The performance of the learning algorithm will be measured in terms of *regret* which is the difference in mean rewards obtained as compared to an optimal policy which al-

ways selects the top k distinct actions with the highest mean reward. To this end, let $\mathcal{A}_t^*$ be the set of k distinct actions that maximizes $\sum_{a\in\mathcal{A}_t^*} f(x_t,a)$ for the given x_t. Then the expected regret is

$$R(T) := \sum_{t=1}^{T} \mathbb{E}\left[\sum_{a\in\mathcal{A}_t^*} f^*(x_t,a) - \sum_{a\in\mathcal{A}_t} f^*(x_t,a)\right]. \quad (1)$$

Regression Oracle. As in (Foster et al., 2018; Simchi-Levi & Xu, 2020), we will rely on the availability of an optimization oracle regression-oracle for the class $\mathcal{F}$ that can perform least-squares regression $\text{argmin}_{f\in\mathcal{F}} \sum_{(x,a,r)} (f(x,a)-r)^2$, where $(x,a,r) \in \mathcal{X}\times\mathcal{A}\times[0,1]$ ranges over the collected data.

3.2. IGW for top-k Contextual Bandits

Our proposed algorithm for top-k arm selection in general contextual bandits in a non-extreme setting is provided as Algorithm 1. It is a natural extension of the Inverse Gap Weighting (IGW) sampling scheme (Abe & Long, 1999; Foster & Rakhlin, 2020; Simchi-Levi & Xu, 2020). In Section 3.3 we will show that this algorithm with $r = 1$ has good regret guarantees for the top-k problem even though the action space is combinatorial, thanks to the linearity of the regret objective in terms of rewards of individual arms in the subset. Note that a naive extension of IGW by treating each action in $\mathcal{A}^k$ as a separate arm would require a computation of $O\left(\binom{A}{k}\right)$ per time step and a similar regret scaling. In contrast, Algorithm 1 only requires $\tilde{O}(A)$ computation for the sampling per time step.

The Inverse Gap Weighting strategy was introduced in (Abe & Long, 1999) and has since then been used for contextual bandits in the realizable setting with general function classes (Foster & Rakhlin, 2020; Simchi-Levi & Xu, 2020; Foster et al., 2020b). Given a set of arms $\mathcal{A}$, an estimate $\widehat{y} : \mathcal{X}\times\mathcal{A} \to \mathbb{R}$ of the reward function, and a context x, the distribution $p = \mathsf{IGW}(\mathcal{A}; \widehat{y}(x,\cdot))$ over arms is given by

$$p(a|x) = \begin{cases} \frac{1}{|\mathcal{A}|+\gamma_l(\widehat{y}(x,a_\star)-\widehat{y}(x,a))} & \text{if } a \neq a_\star \\ 1 - \sum_{a'\in\mathcal{A}:a'\neq a_\star} p(a'|x) & \text{otherwise} \end{cases}$$

where $a_\star = \text{argmax}_{a\in\mathcal{A}} \widehat{y}(x,a)$, γ_l is a scaling factor.

Algorithm 1 proceeds in epochs, indexed by $l = 1,\ldots,e(T)$. Note that $N_{e(T)} = \sum_{l=1}^{e(T)} n_l = T$. The regression model is updated at the beginning of the epoch with all the past data and used throughout the epoch (n_l time steps). The arm selection procedure for the top-k problem involves selecting the top $(k-r)$ arms *greedily* according to the current estimate $\widehat{y}_l$ and then selecting the rest of the arms at random according to the Inverse Gap Weighted distribution over the set of remaining arms. For $r > 1$, the distribution is recomputed over the remaining support every time an arm is selected.

Algorithm 1 Top-k Contextual Bandits with IGW

Arguments: k and r (number of explore slots, $1 \le r \le k$)
for $l \leftarrow 1$ **to** $e(T)$ **do**
  Fit regression oracle to all past data
  $\widehat{y}_l = \text{argmin}_{f\in\mathcal{F}} \sum_{t=1}^{N_{l-1}} \sum_{a\in\Phi_t} (f(x_t,a) - r_t(a))^2$
  for $s \leftarrow N_{l-1}+1$ **to** $N_{l-1}+n_l$ **do**
    Receive x_s
    Let $\widehat{a}_s^1,\ldots,\widehat{a}_s^A$ be the arms ordered in decreasing order according to $\widehat{y}_l(x_s,\cdot)$ values.
    $\mathcal{A}_s = \{\widehat{a}_s^1,\cdots,\widehat{a}_s^{k-r}\}$.
    for $cnt \leftarrow 1$ **to** r **do**
      Compute randomization distribution
      $p = \mathsf{IGW}(\{\mathcal{A}\setminus\mathcal{A}_s\}; \widehat{y}_l(x_s,\cdot))$.
      Sample $a \sim p$. Let $\mathcal{A}_s = \mathcal{A}_s \cup \{a\}$.
    end for
    Obtain rewards $r_s(a)$ for actions $a \in \Phi_s \subseteq \mathcal{A}_s$.
  end for
  Let $N_l = N_{l-1} + n_l$
end for

3.3. Regret of IGW for top-k Contextual Bandits

In this section we show that our algorithm has favorable regret guarantees. Our regret guarantees are only derived for the case when Algorithm 1 is run with $r = 1$. However, we will see that other values of r also work well in practice in Section 5. For ease of exposition we assume $\mathcal{F}$ is finite; our results can be extended to infinite function classes with standard techniques (see e.g. (Simchi-Levi & Xu, 2020)). We first present the bounds under exact realizability.[1]

Theorem 1. *Algorithm 1 under Assumptions 1 and 3, when run with parameters*

$$r = 1; \quad N_l = 2^l; \quad \gamma_l = \frac{1}{32}\sqrt{\frac{c(A-k+1)N_{l-1}}{162\log\left(\frac{|\mathcal{F}|T^3}{\delta}\right)}},$$

has regret bound

$$R(T) = \mathcal{O}\left(k\sqrt{c^{-1}(A-k+1)T\log\left(\frac{|\mathcal{F}|T}{\delta}\right)}\right)$$

with probability at least $1-\delta$, for a finite function class $\mathcal{F}$.

[1] We have not optimized the constants in the definition of γ_l.

In the next theorem we bound the regret under ϵ-realizability.

Theorem 2. *Algorithm 1 under Assumptions 2 and 3, when run with parameters*

$$r = 1; \quad N_l = 2^l; \quad \gamma_l = \frac{\sqrt{c(A-k+1)}}{32\sqrt{\frac{420}{N_{l-1}}\log\left(\frac{|\mathcal{F}|T^3}{\delta}\right)} + 2\epsilon^2}$$

has regret bound

$$R(T) = \mathcal{O}\left(k\sqrt{c^{-1}(A-k+1)T\log\left(\frac{|\mathcal{F}|T}{\delta}\right)} + \epsilon kT\sqrt{A-k+1}\right)$$

with probability at least $1-\delta$, for a finite function class $\mathcal{F}$.

The proofs for both of our main theorems are provided in Appendix C. One of the key ingredients in the proof is an induction hypothesis which helps us relate the top-k regret of a policy with respect to the estimated reward function $\widehat{y}_l \in \mathcal{F}$ at the beginning of epoch l to the actual regret with respect to $f^* \in \mathcal{F}$. The argument can be seen as a generalization of (Simchi-Levi & Xu, 2020) to $k > 1$.

Remark 1. *Note that the constant c which denotes the lowest probability of choosing a presented arm can be made context dependent i.e the probability can be $c(x)$ for a context $x \in \mathcal{X}$. Observe that if $c(x)$ is low or 0, it is not possible to guarantee low regret for such an x. A natural approach is to scale the distribution $p(x)$ by $c(x)$ in the definition of regret. If $\mathbb{E}_{x\sim p}[c(x)] = 1$, the problem is identical to the one with $c = 1$, but now wrt the tilted measure $c(x)p(x)$. If $\mathbb{E}[c(x)]$ is not 1, by rescaling, the regret bound is enlarged by the factor $(\mathbb{E}[c(x)])^{-1}$, an inverse average propensity. Under the above argument the knowledge of $c(x)$ would not be required by the algorithm.*

4. eXtreme Contextual Bandits and Arm Hierarchy

When the number of arms A is large, the goal is to design algorithms so that the computational cost per round is poly-logarithmic in A (i.e. $\mathcal{O}(\text{polylog}(A))$) and overall regret as well. However, owing to known lower bounds (Foster & Rakhlin, 2020), this cannot be achieved without imposing further assumptions on the contextual bandit problem.

Main idea. A key observation is that the regression-oracle framework does not impose any restriction on the structure of the arms and in fact the set of arms can even be context-dependent. We assume that

- For each x, there is an x-dependent decomposition
$$\mathcal{A}_x := \{\boldsymbol{a}_{x,1}, \cdots, \boldsymbol{a}_{x,Z}\}, \tag{2}$$
where $\boldsymbol{a}_{x,1}, \cdots, \boldsymbol{a}_{x,Z}$ form a disjoint union of $\mathcal{A}$ with $Z = \mathcal{O}(\log A)$.
- For any two arms a and a' from any subset $\boldsymbol{a}_{x,i}$, the expected reward function $r(x,a) = \mathbb{E}[r(a)|X = x]$ satisfies the following consistency condition
$$|r(x,a) - r(x,a')| \le \epsilon. \tag{3}$$

By treating $\boldsymbol{a}_{x,1}, \cdots, \boldsymbol{a}_{x,Z}$ as effective arms, the results of Section 3.3 can be applied by *working with functions that are piecewise constant over each $\boldsymbol{a}_{x,i}$*. Such a context-dependent arm space decomposition is a reasonable assumption, because often the rewards from a large subset of arms exhibit minor variations for a given context x.

Motivating example. To motivate and justify the conditions (2) and (3), consider a simple but representative setting where the contexts in $\mathcal{X}$ and arms in $\mathcal{A}$ are both represented as feature vectors in $\mathbb{R}^d$ for a fixed dimension d and the distance between two vectors is measured by the Euclidean norm $\|\cdot\|$. In many applications, the expected reward $r(x,a)$ satisfies the gradient condition $|\partial_a r(x,a)| \le \frac{\eta}{\|x-a\|}$, for some $\eta > 0$, i.e., $r(x,a)$ is sensitive in a only when a is close to x and insensitive when a is far away from x.

Figure 1 illustrates such a reward structure in the 1D case. We can form a binary tree over the arms in $\mathcal{A}$. We can then associate each tree node $e_{h,i}$ (h for height and i for index within this height) with a *routing function* $g_{h,i}(x) = \frac{\text{rad}_{h,i}}{\|x-\text{ctr}_{h,i}\|}$ with $\text{rad}_{h,i}$ and $\text{ctr}_{h,i}$ being the radius and the center of $e_{h,i}$. Given a context x, we perform an adaptive traversal from the root that further explores the children of a node $e_{h,i}$ only when $g_{h,i}(x) = \frac{\text{rad}_{h,i}}{\|x-\text{ctr}_{h,i}\|} > \beta$ for some $\beta > 0$. This traversal breaks the arm space into the disjoint union of several *effective arms* (dependent on x), each of which is either a single arm (in Fig. 1) or a node not being further explored. Condition (2) holds as the number of the effective arms is $O(\log A)$. By choosing β appropriately, one can ensure Condition (3) is also satisfied. These effective arms can then be used as the arms in Algorithm 1 for IGW sampling. The details of this example are given in Appendix B.

General setting. Based on the motivating example, we propose an arm hierarchy for general $\mathcal{X}$ and $\mathcal{A}$. We assume access to a hierarchical partitioning $\mathcal{T}$ of $\mathcal{A}$ that breaks progressively into finer subgroups of similar arms. The partitioning can be represented by a balanced tree that is p-ary till the leaf level. At the

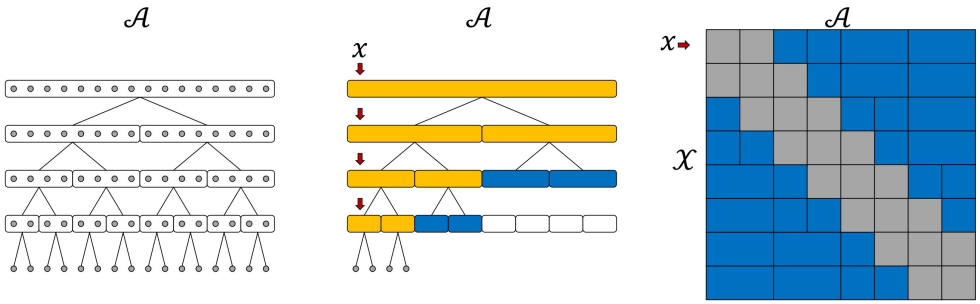


Figure 1: **Left**: an illustration of the hierarchical decomposition for $\mathcal{A}$, where each gray dot indicates an arm. **Middle**: the adaptive search for a given context x. The yellow nodes are further explored as they are close to x while the blue nodes are not as they are far from x. The set of effective arms for x consists of the blue nodes and the singleton arms in the yellow leaf nodes. **Right**: For a fixed x, the corresponding row shows the x-dependent hierarchical arm space decomposition. As x varies, the decomposition also changes. Each blue block stands for a non-singleton effective arm, valid for a contiguous block of contexts. Each gray block contains the singleton effective arms, valid again for a contiguous block of contexts.

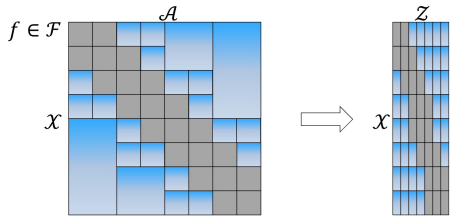


Figure 2: **Left**: A $(\mathcal{T}, g, b)$-constant predictor function $f(x, a)$ in the 1D motivating example with $\mathcal{X} \subset [0, 1]$ and $\mathcal{A} \subset [0, 1]$. Within each blue block, $f(x, a)$ is constant in a but varies with x. **Right**: the function $\tilde{f}$ after the reduction.

leaf level, each node can have a maximum of $m > p$ children, each of which is a singleton arm in $\mathcal{A}$. The height of such a tree is $H = \lceil \log_p \lceil A/m \rceil \rceil$. With a slight abuse of notation, we use $e_{h,i}$ to denote a node in the tree as well as the subset of singleton arms in the subtree of the node.

Each internal node $e_{h,i}$ is assumed to be associated with a routing function $g_{h,i}(x)$ mapping $\mathcal{X} \to [0, 1]$ and $\mathcal{C}_{h,i}$ is used to denote the immediate children of node $e_{h,i}$. Based on these routing functions and an integer parameter b, we define a beam search in Algorithm 2 for any context $x \in \mathcal{X}$ as an input. During its execution, this beam search keeps at each level h only the top b nodes that return the highest $g_{h,i}(x)$ values. The output of the beam search, denoted also by $\mathcal{A}_x$, is the union of a set of nodes denoted as I_x and a set of singleton arms denoted as S_x. The tree structure ensures that there are at most bm singleton arms in S_x and at most $(p-1)b(H-1)$ nodes in I_x. Therefore, $|\mathcal{A}_x| \leq (p-1)b(H-1) + bm = O(\log A)$, implying that $\mathcal{A}_x$ satisfies (2). Though the cardinality $|\mathcal{A}_x|$ can vary slightly depending on the context x, in what follows we make the simplifying assumption that $|\mathcal{A}_x|$ is equal to a constant $Z = O(\log A)$ independent of x and denote $\mathcal{A}_x = \{\boldsymbol{a}_{x,1}, \ldots, \boldsymbol{a}_{x,Z}\}$.

Algorithm 2 Beam search

1: **Arguments**: beam-size b, $\mathcal{T}$, routing functions $\{g\}$, x
2: Initialize $\texttt{codes} = [(1, 1)]$ and $I_x^b = \emptyset$.
3: **for** $h = 1, \cdots, H-1$ **do**
4: Let $\texttt{labels} = \cup_{(h-1,i) \in \texttt{codes}} \mathcal{C}_{h-1,i}$.
5: Let $\texttt{codes}$ be top-b nodes in $\texttt{labels}$ according to the values $g_{h,i}(x)$.
6: Add the nodes in $\texttt{labels} \setminus \texttt{codes}$ to I_x.
7: **end for**
8: Let $S_x = \cup_{(H-1,i) \in \texttt{codes}} \mathcal{C}_{H-1,i}$.
9: Return $\mathcal{A}_x = S_x \cup I_x$.

To ensure the consistency condition (3) in the general case, one requires the expected reward function $r(x, a)$ to be nearly constant over each effective arm $\boldsymbol{a}_{x,i}$ and work with a function class that is constant over each $\boldsymbol{a}_{x,i}$. The following definition formalizes this.

Definition 1. *Given a hierarchy $\mathcal{T}$ with routing function family $\{g_{h,i}(\cdot)\}$ and a beam-width b, a function $f(x, a)$ is $(\mathcal{T}, g, b)$-constant if for every $x \in \mathcal{X}$*

$$f(x, a) = f(x, a') \quad \textit{for all} \quad a, a' \in e_{h,i},$$

for any node $e_{h,i}$ in $I_x \subset \mathcal{A}_x$. A class of functions $\mathcal{F}$ is $(\mathcal{T}, g, b)$-constant if each $f \in \mathcal{F}$ is $(\mathcal{T}, g, b)$-constant.

Figure 2 (left) provides an illustration of a $(\mathcal{T}, g, b)$-constant predictor function for the simple case $\mathcal{X} \subset [0, 1]$ and $\mathcal{A} \subset [0, 1]$. In the eXtreme setting, we always assume that our predictor class $\mathcal{F}$ is $(\mathcal{T}, g, b)$-constant. By further assuming that the expected reward $r(x, a)$ satisfies either Assumption 1 or Assumption 2, Condition (3) is satisfied.

4.1. IGW for top-k eXtreme Contextual Bandits

In this section we provide our algorithm for the eXtreme setting. As Definition 1 reduces the eXtreme problem with A arms to a non-extreme problem with only $Z = \mathcal{O}(\log A)$ effective arms, Algorithm 3 essentially uses the beam-search method in Algorithm 2 to construct this reduced problem. The IGW randomization is performed over the effective arms and if a non-singleton arm (i.e., an internal node of $\mathcal{T}$) is chosen, we substitute it with a randomly chosen singleton arm that lies in the sub-tree of that node. More specifically, for a $(\mathcal{T}, g, b)$-constant class $\mathcal{F}$, we define for each $f \in \mathcal{F}$ a new function $\tilde{f} : \mathcal{X} \times [Z] \to [0,1]$ s.t. for any $z = 1, \ldots, Z$ we have $\tilde{f}(x, z) = f(x, a)$ for some fixed $a \in \boldsymbol{a}_{x,z}$. Here, we assume that for any x the beam-search process in Algorithm 2 returns the effective arms in $\mathcal{A}_x$ in a fixed order and $\boldsymbol{a}_{x,z}$ is the z-th arm in this order. The collection of these new functions over the context set $\mathcal{X}$ and the reduced arm space $\mathcal{Z} = [Z]$ is denoted by $\tilde{\mathcal{F}} = \{\tilde{f} : f \in \mathcal{F}\}$. Figure 2 (right) provides an illustration of a function $\tilde{f}(x, z)$ obtained after the reduction.

Algorithm 3 eXtreme Top-k Contextual Bandits with IGW

Arguments: k, number of explore slots: $1 \le r \le k$
for $l \leftarrow 1$ **to** $e(T)$ **do**
  Fit regression oracle to all past data
  $\widehat{y}_l = \operatorname{argmin}_{\tilde{f} \in \tilde{\mathcal{F}}} \sum_{t=1}^{N_{l-1}} \sum_{z \in \Phi_t} (\tilde{f}(x_t, z) - \tilde{r}_t(z))^2$
  for $s \leftarrow N_{l-1} + 1$ **to** $N_{l-1} + n_l$ **do**
    Receive x_s
    Use Algorithm 2 to get $\mathcal{A}_{x_s} = \{\boldsymbol{a}_{x_s,1}, \ldots, \boldsymbol{a}_{x_s,Z}\}$.
    Let $z_1, \ldots, z_Z$ be the arms in $[Z]$ in the descending order according to $\widehat{y}_l$.
    $\mathcal{Z}_s = \{z_1, \cdots, z_{k-r}\}$.
    for $c \leftarrow 1$ **to** r **do**
      Compute randomization distribution
      $p = \mathsf{IGW}([Z] \setminus \mathcal{Z}_s; \hat{y}_l(x_s, \cdot))$.
      Sample $z \sim p$. Let $\mathcal{Z}_s = \mathcal{Z}_s \cup \{z\}$.
    end for
    $B_s = \{\}$.
    for z in $\mathcal{Z}_s$ **do**
      If $\boldsymbol{a}_{x_s,z}$ is singleton arm, then add it to B_s.
      Otherwise sample a singleton arm a in the sub-tree rooted at the node $\boldsymbol{a}_{x_s,z}$ and add a to B_s.
    end for
    Choose the arms in B_s.
    Map the rewards back to the corresponding effective arms in $\mathcal{Z}_s$ and record $\{\tilde{r}_s(z), z \in \Phi_s\}$.
  end for
  Let $N_l = N_{l-1} + n_l$
end for

As a practical example, we can maintain the function class $\mathcal{F}$ such that each member $f \in \mathcal{F}$ is represented as a set of regressors at the internal nodes as well as the singleton arms in the tree. These regressors map contexts to $[0,1]$. For an $f \in \mathcal{F}$, the regressor at each node is constant over the arms a within this node and is only trained on past samples for which that node was selected as a whole in $\mathcal{Z}_s$ in Algorithm 3; the regressor at a singleton arm can be trained on all samples obtained by choosing that arm. Note that even though we might have to maintain a lot of regression functions, many of them can be sparse if the input contexts are sparse, because they are only trained on a small fraction of past training samples.

4.2. Top-k Analysis in the eXtreme Setting

We can analyze Algorithm 3 under the realizability assumptions (Assumption 1 or Assumption 2) when the class of functions satisfies Definition 1). Our main result is a reduction style argument that provides the following corollary of Theorems 1 and 2.

Corollary 1. *Algorithm 3 when run with parameter $r = 1$ has the following regret guarantees:*

(i) If Assumptions 1 and 3 hold and the function class $\mathcal{F}$ is $(\mathcal{T}, g, b)$-constant (Definition 1), then setting parameters as in Theorem 1 ensures that the regret bound stated in Theorem 1 holds with A replaced by $O(\log A)$.

(ii) If Assumptions 2 and 3 hold and the function class $\mathcal{F}$ is $(\mathcal{T}, g, b)$-constant (Definition 1), then setting parameters as in Theorem 2 ensures that the regret bound stated in Theorem 2 holds with A replaced by $O(\log A)$.

5. Empirical Results

We compare our algorithm with well known baselines on various real world datasets. We first perform a semi-synthetic experiment in a realizable setting. Then we use eXtreme Multi-Label Classification (XMC) (Bhatia et al., 2016) datasets to test our reduction scheme. The different exploration sampling strategies used in our experiments are [2]: **Greedy-topk:** The top-k effective arms for each context are chosen greedily according to the regression score; **Boltzmann-topk:** The top-$(k-r)$ arms are selected greedily. Then the next r arms are selected one by one, each time recomputing the Boltzmann distribution over the remaining arms. Under this sampling scheme the probability of sampling arm $\tilde{a}$ is proportional to $\exp(\log(N_{l-1})\beta\tilde{f}(x, \tilde{a}))$ (Cesa-

[2] Note that all these exploration strategies have been extended to the top-k setting using the ideas in Algorithm 1 and many popular contextual bandit algorithms like the ones in (Bietti et al., 2018) cannot be easily extended to the top-k setting.

Bianchi et al., 2017); **ϵ-greedy-topk:** Same as above but the last r arms are selected one by one using a scheme where the probability of sampling arm $\tilde{a}$ is proportional to $(1-\epsilon)+\epsilon/A'$ if $\tilde{a}$ is the arm with the highest score, otherwise the probability is ϵ/A' where A' is the number of arms remaining; **IGW-topk:** This is essentially the sampling strategy in Algorithm 1. We set $\gamma_l = \sqrt{CN_{l-1}A'}$ for the l-th epoch where A' is the number of remaining arms.

Realizable Experiment. In order to create a realizable setting that is realistic, we choose the eurlex-4k XMC dataset (Bhatia et al., 2016) in Table 1 and for each arm/label $a \in A$, we fit linear regressor weights ν_a^* that minimizes $\mathbb{E}_x[([x;1.0]^T\nu_a^* - \mathbb{E}[r_a(t)|x])^2]$ over the dataset. Then we consider a derived system where $\mathbb{E}[r_a(t)|x] = [x;1.0]^T\nu_a^*$ for all x, a that is the learnt weights from before exactly represent the mean rewards of the arms. This system is then realizable for Algorithm 1 when the function $\mathcal{F}$ is linear. Figure 3(a) shows the progressive mean reward (sum of rewards till time t divided by t) for all the sampling strategies compared. We see that the IGW sampling strategy in Algorithm 1 outperforms all the others by a large margin. For more details please refer to Appendix H. Note that the hyper-parameters of all the algorithms are tuned on this dataset in order to demonstrate that even with tuned hyper-parameter choices IGW is the optimal scheme for this realizable experiment. The experiment is done with $k = 50, r = 25$ and $b = 10$.

eXtreme Experiments. We now present our empirical results on eXtreme multi-label datasets. Our experiments are performed under simulated bandit feedback using real-world eXtreme multi-label classification datasets (Bhatia et al., 2016). This experiment startegy is widely used in the literature (Agarwal et al., 2014; Bietti et al., 2018) with non-eXtreme multi-class datasets (see Appendix H for more details). Our implementation uses a hierarchical linear function class inspired by (Yu et al., 2020). The hyper-parameters in all the algorithms are tuned on the eurlex-4k datasets and then held fixed. This is in line with (Bietti et al., 2018), where the parameters are tuned on a set of datasets and then held fixed.

We follow the framework described in Section 4 using a hierarchical linear function class. We first form the tree and the routing functions from the held out portion of each dataset. The assumption is that there is a small supervised dataset available to each algorithm before proceeding with the simulated bandit feedback experiment. This dataset is used to form an approximately balanced binary tree over the labels till the penultimate level. The nodes in the penultimate level can have a maximum of m children which are the original arms. The value of m is specified in Table 1 for each dataset. The division of the labels in each level of the tree is done through hierarchical clustering over label embeddings, where at each clustering step we use the algorithm from (Dhillon, 2001). The specific label embedding technique that we use is called Positive Instance Feature Aggregation (PIFA) (see (Jasinska et al., 2016) for more details).

The routing functions for each internal node in the tree is essentially a one-vs-all linear classifier trained on the held out set. The classifiers are trained using a SVM ℓ_2-hinge loss. The positive and negative examples for each internal node is selected similar to the strategy in (Prabhu et al., 2018). Finally for the regression function $\tilde{f}(x, \tilde{a})$ where $\tilde{a}$ can be an original arm or an internal node in the tree, we train a linear regressor $\tilde{f}(x, \tilde{a}) = \nu_{\tilde{a}}^T[x;1]$ as we progress through the experiment as in Algorithm 3. Note that the held out dataset is only used to train the tree and the routing function for each of the algorithms, while the regression functions are trained from scratch only using the samples observed during the bandit feedback experiment. In the interest of space we refer the readers to Appendix H for more implementation details. We provide our implementation here.

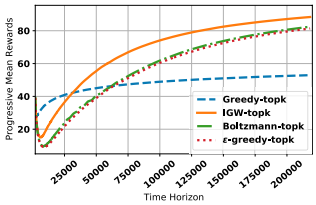
We use 6 XMC datasets for our experiments. Table 1 provides some basic properties of each dataset. We can see that the number of arms in the largest dataset is as large as 2.8MM. The column Initialization Size denotes the size of the held out set used to intialize our algorithms. Note that for the datasets eurlex-4k and wiki10-31k we bootstrap the original training dataset to a larger size by sampling with replacement, as the original number of samples are too small to show noticeable effects.

Dataset	Initialization Size	Time-Horizon	No. of Arms	Max. Leaf Size (m)
eurlex-4k	5000	154490	4271	10
amazoncat-13k	5000	1186239	13330	10
wiki10-31k	5000	141460	30938	10
wiki-500k	20000	1779881	501070	100
amazon-670k	20000	490449	670091	100
amazon-3m	50000	1717899	2812281	100

Table 1: Properties of eXtreme Datasets

	X-Greedy	X-IGW-topk	X-Boltzmann-topk	X-ϵ-greedy-topk
X-Greedy	-	0W/0D/6L	1W/0D/5L	0W/1D/5L
X-IGW-topk	6W/0D/0L	-	4W/1D/1L	6W/0D/0L
X-Boltzmann-topk	5W/0D/1L	1W/1D/4L	-	3W/0D/3L
X-ϵ-greedy-topk	5W/1D/0L	0W/0D/6L	3W/0D/3L	-

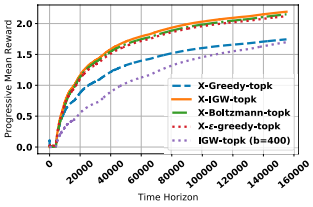
Table 2: Win/Draw/Loss statistics among algorithms for the 6 datasets. When the difference in results between two algorithms is not significant according to the statistical significance formula in (Bietti et al., 2018) then it is deemed to be a draw.



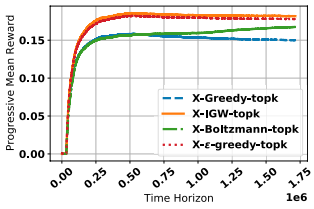
(a) Realizable eurlex-4k

Beam Size (b)	Inference Time (ms)
10	7.85
30	12.84
100	27.83
2.9K (all arms)	799.06

(b) Inference Time per context on amazon-3m



(c) eurlex-4k



(d) amazon-3m

Figure 3: In (a) we compare the different sampling strategies on a realizable setting with $k = 50$ and $r = 25$, derived from the eurlex-4k dataset. In (b) we compare the avg. inference times per context vs different beam sizes on the amazon-3m dataset. Note that for this dataset $b = 290,000$ will include all arms in the beam in our setting and is order wise equivalent to no hierarchy. This comparison is done for inference in a setting with $k = 5, r = 3$. Note that for larger datasets in Table 1 our implementation with $b = 10, 30$ remains efficient for real-time inference as the time-complexity scales only with the beams-size and the height of the tree. We plot the progressive mean rewards collected by each algorithm as a function of time in two of our 6 datasets in (c)-(d) where the algorithms are implemented under our eXtreme reduction framework. In our experiments in (c)-(d) we have $k = 5$ and $r = 3$. The beam size is 10 except for IGW-topk (b=400) in (c), which serves as a proxy for Algorithm 1 without the extreme reduction, as $b = 400$ includes all the arms in this dataset.

We plot the progressive mean rewards (total rewards collected till time t divided by t) for all the algorithms in Figure 3 (c)-(d) for two datasets. The rest of the plots are included in Figure 4 in Appendix G due to space constraints. The algorithm names are prepended with an X to denote that the sampling is performed under the reduction framework of Algorithm 3. In our experiments the number of arms allowed to be chosen each time is $k = 5$. In Algorithm 3 we set the number of explore slots $r = 3$ and $b = 10$ (unless otherwise specified). We see that all the exploratory algorithms do much better than the greedy version i.e our eXtreme reduction framework works for structured exploration when the number of arms are in thousands or millions. The efficacy of the reduction framework is further demonstrated by X-IGW-topk(b=10) being better than IGW-topk (b=400) by 29% in terms of the mean reward, in Figure 3(c). Note that here IGW-topk(b=400) serves as a proxy for Algorithm 1 directly applied without the hierarchy, as the beam includes all the arms. The IGW scheme is always among the top 2 strategies in all datasets. It is the only strategy among the baselines that has optimal theoretical performance and this shows that the algorithm is practical. Table 2 provides Win(W)/Draw(D)/Loss(L) for each algorithm against the others. We use the same W/D/L scheme as in (Bietti et al., 2018) to create this table. Note that X-IGW-topk has the highest win percentage overall. In Figure 3(b) we compare the inference times for IGW of our hierarchical linear implementation for different beam-sizes on amazon-3m. Note that $b = 2.9K$ will include all arms in this dataset and is similar to a flat hierarchy. This shows that our algorithm will remain practical for real time inference on large datasets when $b \leq 30$ is used.

6. Discussion

We provide regret guarantees for the top-k arm selection problem in realizable contextual bandits under general function classes. The algorithm can be theoretically and practically extended to extreme number of arms under our proposed reduction framework which models a practically motivated arm hierarchy. We benchmark our algorithms on XMC datasets under simulated bandit feedback.

There are interesting directions for future work, for instance extending the analysis to a setting where the reward derived from the k arms is a set function with interesting structures like sub-modularity. It would also be interesting to analyze the eXtreme setting where the routing functions and hierarchy can be updated in a data driven manner after every few epochs. The routing function training can be potentially de-biased from the effect of bandit feedback using importance sampling approaches. It is an open problem to adapt the clustering algorithm to data collected from bandit feedback (or to principally show that the regular training is sufficient).

We do not anticipate any negative ethical or social impact of this work.

Acknowledgement

Rajat Sen would like to thank Hsiang-Fu Yu for helpful discussions regarding the implementation of our algorithm in the eXtreme setting.

References

Abe, N. and Long, P. M. Associative reinforcement learning using linear probabilistic concepts. In *ICML*, pp. 3–11. Citeseer, 1999.

Agarwal, A., Dudík, M., Kale, S., Langford, J., and Schapire, R. Contextual bandit learning with predictable rewards. In *Artificial Intelligence and Statistics*, pp. 19–26, 2012.

Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L., and Schapire, R. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pp. 1638–1646, 2014.

Agarwal, M. and Aggarwal, V. Regret bounds for stochastic combinatorial multi-armed bandits with linear space complexity. *arXiv preprint arXiv:1811.11925*, 2018.

Agrawal, S. and Goyal, N. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pp. 127–135, 2013.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Bartlett, P., Dani, V., Hayes, T., Kakade, S., Rakhlin, A., and Tewari, A. High-probability regret bounds for bandit online linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory - COLT 2008*, pp. 335–342. Omnipress, 2008.

Beygelzimer, A., Langford, J., Li, L., Reyzin, L., and Schapire, R. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 19–26, 2011.

Bhatia, K., Dahiya, K., Jain, H., Mittal, A., Prabhu, Y., and Varma, M. The extreme classification repository: Multi-label datasets and code, 2016. URL http://manikvarma.org/downloads/XC/XMLRepository.html.

Bietti, A., Agarwal, A., and Langford, J. A contextual bandit bake-off. *arXiv preprint arXiv:1802.04064*, 2018.

Cesa-Bianchi, N. and Lugosi, G. Combinatorial bandits. *Journal of Computer and System Sciences*, 78 (5):1404–1422, 2012.

Cesa-Bianchi, N., Gentile, C., Lugosi, G., and Neu, G. Boltzmann exploration done right. In *Advances in neural information processing systems*, pp. 6284–6293, 2017.

Chen, W., Hu, W., Li, F., Li, J., Liu, Y., and Lu, P. Combinatorial multi-armed bandit with general reward functions. In *Advances in Neural Information Processing Systems*, pp. 1659–1667, 2016.

Chu, W., Li, L., Reyzin, L., and Schapire, R. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 208–214, 2011.

Combes, R., Talebi Mazraeh Shahi, M. S., Proutiere, A., et al. Combinatorial bandits revisited. *Advances in neural information processing systems*, 28:2116–2124, 2015.

Dani, V., Hayes, T. P., and Kakade, S. M. Stochastic linear optimization under bandit feedback. 2008.

Dhillon, I. S. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 269–274, 2001.

Durand, A., Achilleos, C., Iacovides, D., Strati, K., Mitsis, G. D., and Pineau, J. Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis. In *Machine Learning for Healthcare Conference*, pp. 67–82, 2018.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. Liblinear: A library for large linear classification. *the Journal of machine Learning research*, 9:1871–1874, 2008.

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

Foster, D. J. and Rakhlin, A. Beyond ucb: Optimal and efficient contextual bandits with regression oracles. *arXiv preprint arXiv:2002.04926*, 2020.

Foster, D. J., Agarwal, A., Dudík, M., Luo, H., and Schapire, R. E. Practical contextual bandits with regression oracles. *arXiv preprint arXiv:1803.01088*, 2018.

Foster, D. J., Gentile, C., Mohri, M., and Zimmert, J. Adapting to misspecification in contextual bandits. *Advances in Neural Information Processing Systems*, 33, 2020a.

Foster, D. J., Rakhlin, A., Simchi-Levi, D., and Xu, Y. Instance-dependent complexity of contextual bandits and reinforcement learning: A disagreement-based perspective. *arXiv preprint arXiv:2010.03104*, 2020b.

Guennebaud, G., Jacob, B., et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

Jasinska, K., Dembczynski, K., Busa-Fekete, R., Pfannschmidt, K., Klerx, T., and Hullermeier, E. Extreme f-measure maximization using sparse probability estimates. In *International Conference on Machine Learning*, pp. 1435–1444, 2016.

Khandagale, S., Xiao, H., and Babbar, R. Bonsai: diverse and shallow trees for extreme multi-label classification. *Machine Learning*, pp. 1–21, 2020.

Krause, A. and Ong, C. Contextual gaussian process bandit optimization. *Advances in neural information processing systems*, 24:2447–2455, 2011.

Kveton, B., Wen, Z., Ashkan, A., and Szepesvari, C. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, pp. 535–543, 2015.

Langford, J. and Zhang, T. The epoch-greedy algorithm for contextual multi-armed bandits. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pp. 817–824. Citeseer, 2007.

Li, S., Karatzoglou, A., and Gentile, C. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 539–548, 2016.

Lin, T., Abrahao, B., Kleinberg, R., Lui, J., and Chen, W. Combinatorial partial monitoring game with linear feedback and its applications. In *International Conference on Machine Learning*, pp. 901–909, 2014.

Lopez, R., Dhillon, I., and Jordan, M. I. Learning from extreme bandit feedback. *arXiv preprint arXiv:2009.12947*, 2020.

Majzoubi, M., Zhang, C., Chari, R., Krishnamurthy, A., Langford, J., and Slivkins, A. Efficient contextual bandits with continuous actions. *arXiv preprint arXiv:2006.06040*, 2020.

McMahan, H. B. and Streeter, M. Tighter bounds for multi-armed bandits with expert advice. 2009.

Merlis, N. and Mannor, S. Batch-size independent regret bounds for the combinatorial multi-armed bandit problem. *arXiv preprint arXiv:1905.03125*, 2019.

Narita, Y., Yasui, S., and Yata, K. Efficient counterfactual learning from bandit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4634–4641, 2019.

Prabhu, Y., Kag, A., Harsola, S., Agrawal, R., and Varma, M. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *Proceedings of the 2018 World Wide Web Conference*, pp. 993–1002, 2018.

Qin, L., Chen, S., and Zhu, X. Contextual combinatorial bandit and its application on diversified online recommendation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pp. 461–469. SIAM, 2014.

Rakhlin, A. and Sridharan, K. Bistro: An efficient relaxation-based method for contextual bandits. In *ICML*, pp. 1977–1985, 2016.

Rejwan, I. and Mansour, Y. Top-k combinatorial bandits with full-bandit feedback. In *Algorithmic Learning Theory*, pp. 752–776. PMLR, 2020.

Simchi-Levi, D. and Xu, Y. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Available at SSRN*, 2020.

Swaminathan, A., Krishnamurthy, A., Agarwal, A., Dudik, M., Langford, J., Jose, D., and Zitouni, I. Off-policy evaluation for slate recommendation. In *Advances in Neural Information Processing Systems*, pp. 3632–3642, 2017.

Villar, S. S., Bowden, J., and Wason, J. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 30(2):199, 2015.

Wydmuch, M., Jasinska, K., Kuznetsov, M., Busa-Fekete, R., and Dembczynski, K. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In *Advances in Neural Information Processing Systems*, pp. 6355–6366, 2018.

You, R., Zhang, Z., Wang, Z., Dai, S., Mamitsuka, H., and Zhu, S. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In *Advances in Neural Information Processing Systems*, pp. 5820–5830, 2019.

Yu, H.-F., Zhong, K., and Dhillon, I. S. Pecos: Prediction for enormous and correlated output spaces. *arXiv preprint arXiv:2010.05878*, 2020.

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