Taming the Monster: A Fast and Simple Algorithm for Contextual Bandits

Alekh Agarwal[1], Daniel Hsu[2], Satyen Kale[3], John Langford[1], Lihong Li[1], and Robert E. Schapire[4]

[1]Microsoft Research
[2]Columbia University
[3]Yahoo! Labs
[4]Princeton University

February 5, 2014

Abstract

We present a new algorithm for the contextual bandit learning problem, where the learner repeatedly takes an *action* in response to the observed *context*, observing the *reward* only for that action. Our method assumes access to an oracle for solving cost-sensitive classification problems and achieves the statistically optimal regret guarantee with only $\tilde{O}(\sqrt{T})$ oracle calls across all T rounds. By doing so, we obtain the most practical contextual bandit learning algorithm amongst approaches that work for general policy classes. We further conduct a proof-of-concept experiment which demonstrates the excellent computational and prediction performance of (an online variant of) our algorithm relative to several baselines.

1 Introduction

In the contextual bandit problem, an agent collects rewards for actions taken over a sequence of rounds; in each round, the agent chooses an action to take on the basis of (i) *context* (or features) for the current round, as well as (ii) *feedback*, in the form of rewards, obtained in previous rounds. The feedback is *incomplete*: in any given round, the agent observes the reward only for the chosen action; the agent does not observe the reward for other actions. Contextual bandit problems are found in many important applications such as online recommendation and clinical trials, and represent a natural half-way point between supervised learning and reinforcement learning. The use of features to encode context is inherited from supervised machine learning, while *exploration* is necessary for good performance as in reinforcement learning.

The choice of exploration distribution on actions is important. The strongest known results (Auer et al., 2002; McMahan and Streeter, 2009; Beygelzimer et al., 2011) provide an algorithm that carefully controls the exploration distribution to achieve an optimal regret after T rounds of

$$O\left(\sqrt{KT\log|\Pi|}\right)$$

relative to a set of policies $\Pi \subseteq X^A$ mapping contexts $x \in X$ to actions $a \in A$ (where K is the number of actions). The regret is the difference between the cumulative reward of the best policy in Π and the cumulative reward collected by the algorithm. Because the bound has a mild logarithmic dependence on $|\Pi|$, the algorithm can compete with very large policy classes that are likely to yield high rewards, in which case the algorithm also earns high rewards. However, the computational complexity of the algorithm is linear in $|\Pi|$, making it tractable for only simple policy classes.

A sub-linear in $|\Pi|$ running time is possible for policy classes that can be efficiently searched. In this work, we use the abstraction of an optimization oracle to capture this property: given a set of context/reward vector pairs, the oracle returns a policy in Π with maximum total reward. Using such an oracle in an i.i.d. setting (formally defined in Section 2.1), it is possible to create ϵ-greedy (Sutton and Barto, 1998) or epoch-greedy (Langford and Zhang, 2007) algorithms that run in time $O(\log |\Pi|)$ with only a single call to the oracle per round. However, these algorithms have suboptimal regret bounds of $O((K \log |\Pi|)^{1/3} T^{2/3})$ because the algorithms randomize uniformly over actions when they choose to explore.

The Randomized UCB algorithm of Dudík et al. (2011a) achieves the optimal regret bound (up to logarithmic factors) in the i.i.d. setting, and runs in time $\mathrm{poly}(T, \log |\Pi|)$ with $\mathrm{poly}(T)$ calls to the optimization oracle. This is a fascinating result because it shows that the oracle can provide an exponential speed-up over previous algorithms with optimal regret bounds. However, the algorithm is not practical because the degree of the running time polynomial is high.

In this work, we prove the following[1]:

Theorem 1. *There is an algorithm for the i.i.d. contextual bandit problem with an optimal regret bound requiring $\tilde{O}(\sqrt{KT})$ calls to the optimization oracle over T rounds.*

Concretely, we make $\tilde{O}(\sqrt{KT})$ calls to the oracle with a net running time of $\tilde{O}(T^{1.5} K^{1/2} \log |\Pi|)$, vastly improving over the complexity of Randomized UCB. The major components of the new algorithm are (i) a new coordinate descent procedure for computing a very sparse distribution over policies which can be efficiently sampled from, and (ii) a new epoch structure which allows the distribution over policies to be updated very infrequently. We consider variants of the epoch structure that make different computational trade-offs; on one extreme we concentrate the entire computational burden on $O(\log T)$ rounds with $\tilde{O}(\sqrt{KT})$ oracle calls each time, while on the other we spread our computation over $\sqrt{T}$ rounds with $\tilde{O}(K)$ oracle calls for each of these rounds. We stress that in either case, the total number of calls to the oracle is only sublinear. Finally, we develop a more efficient online variant, and conduct a proof-of-concept experiment showing low computational complexity and high reward relative to several natural baselines.

Motivation and related work. The EXP4-family of algorithms (Auer et al., 2002; McMahan and Streeter, 2009; Beygelzimer et al., 2011) solve the contextual bandit problem with optimal regret by updating weights (multiplicatively) over all policies in every round. Except for a few special cases (Helmbold and Schapire, 1997; Beygelzimer et al., 2011), the running time of such *measure-based* algorithms is generally linear in the number of policies.

In contrast, the Randomized UCB algorithm of Dudík et al. (2011a) is based on a natural abstraction from supervised learning—the ability to efficiently find a function in a rich function class that minimizes the loss on a training set. This abstraction is encapsulated in the notion of an optimization oracle, which is also useful for ϵ-greedy (Sutton and Barto, 1998) and epoch-greedy (Langford and Zhang, 2007). However, these algorithms have only suboptimal regret bounds.

Another class of approaches based on Bayesian updating is Thompson sampling (Thompson, 1933; Li, 2013), which often enjoys strong theoretical guarantees in expectation over the prior and good empirical performance (Chapelle and Li, 2011). Such algorithms, as well as the closely related upper-confidence bound algorithms (Auer, 2002; Chu et al., 2011), are computationally tractable in cases where the posterior distribution over policies can be efficiently maintained or approximated. In our experiments, we compare to a strong baseline algorithm that uses this approach (Chu et al., 2011).

To circumvent the $\Omega(|\Pi|)$ running time barrier, we restrict attention to algorithms that only access the policy class via the optimization oracle. Specifically, we use a cost-sensitive classification oracle, and a key challenge is to design good supervised learning problems for querying this oracle. The Randomized UCB algorithm of Dudík et al. (2011a) uses a similar oracle to construct a distribution over policies that solves a certain convex program. However, the number of oracle calls in their work is prohibitively large, and the statistical analysis is also rather complex.[2]

[1] Throughout this paper, we use the $\tilde{O}$ notation to suppress dependence on logarithmic factors.

[2] The paper of Dudík et al. (2011a) is colloquially referred to, by its authors, as the "monster paper" (Langford, 2014).

Main contributions. In this work, we present a new and simple algorithm for solving a similar convex program as that used by Randomized UCB. The new algorithm is based on coordinate descent: in each iteration, the algorithm calls the optimization oracle to obtain a policy; the output is a sparse distribution over these policies. The number of iterations required to compute the distribution is small—at most $\tilde{O}(\sqrt{Kt})$ in any round t. In fact, we present a more general scheme based on epochs and warm start in which the total number of calls to the oracle is, with high probability, just $\tilde{O}(\sqrt{KT})$ *over all* T *rounds*; we prove that this is nearly optimal for a certain class of optimization-based algorithms. The algorithm is natural and simple to implement, and we provide an arguably simpler analysis than that for Randomized UCB. Finally, we report proof-of-concept experimental results using a variant algorithm showing strong empirical performance.

2 Preliminaries

In this section, we recall the i.i.d. contextual bandit setting and some basic techniques used in previous works (Auer et al., 2002; Beygelzimer et al., 2011; Dudík et al., 2011a).

2.1 Learning Setting

Let A be a finite set of K actions, X be a space of possible contexts (*e.g.*, a feature space), and $\Pi \subseteq A^X$ be a finite set of policies that map contexts $x \in X$ to actions $a \in A$.[3] Let $\Delta^\Pi := \{Q \in \mathbb{R}^\Pi : Q(\pi) \geq 0 \,\forall \pi \in \Pi, \sum_{\pi\in\Pi} Q(\pi) \leq 1\}$ be the set of non-negative weights over policies with total weight at most one, and let $\mathbb{R}^A_+ := \{r \in \mathbb{R}^A : r(a) \geq 0 \,\forall a \in A\}$ be the set of non-negative reward vectors.

Let $\mathcal{D}$ be a probability distribution over $X \times [0,1]^A$, the joint space of contexts and reward vectors; we assume actions' rewards from $\mathcal{D}$ are always in the interval $[0,1]$. Let $\mathcal{D}_X$ denote the marginal distribution of $\mathcal{D}$ over X.

In the i.i.d. contextual bandit setting, the context/reward vector pairs $(x_t, r_t) \in X \times [0,1]^A$ over all rounds $t = 1, 2, \ldots$ are randomly drawn independently from $\mathcal{D}$. In round t, the agent first observes the context x_t, then (randomly) chooses an action $a_t \in A$, and finally receives the reward $r_t(a_t) \in [0,1]$ for the chosen action. The (observable) record of interaction resulting from round t is the quadruple $(x_t, a_t, r_t(a_t), p_t(a_t)) \in X \times A \times [0,1] \times [0,1]$; here, $p_t(a_t) \in [0,1]$ is the probability that the agent chose action $a_t \in A$. We let $H_t \subseteq X \times A \times [0,1] \times [0,1]$ denote the *history* (set) of interaction records in the first t rounds. We use the shorthand notation $\widehat{\mathbb{E}}_{x\sim H_t}[\cdot]$ to denote expectation when a context x is chosen from the t contexts in H_t uniformly at random.

Let $\mathcal{R}(\pi) := \mathbb{E}_{(x,r)\sim\mathcal{D}}[r(\pi(x))]$ denote the expected (instantaneous) reward of a policy $\pi \in \Pi$, and let $\pi_\star := \arg\max_{\pi\in\Pi} \mathcal{R}(\pi)$ be a policy that maximizes the expected reward (the *optimal policy*). Let $\mathrm{Reg}(\pi) := \mathcal{R}(\pi_\star) - \mathcal{R}(\pi)$ denote the *expected (instantaneous) regret* of a policy $\pi \in \Pi$ relative to the optimal policy. Finally, the (empirical cumulative) regret of the agent after T rounds[4] is defined as

$$\sum_{t=1}^{T} \big(r_t(\pi_\star(x_t)) - r_t(a_t)\big).$$

2.2 Inverse Propensity Scoring

An unbiased estimate of a policy's reward may be obtained from a history of interaction records H_t using *inverse propensity scoring* (IPS; also called *inverse probability weighting*): the expected reward of policy $\pi \in \Pi$ is estimated as

$$\widehat{\mathcal{R}}_t(\pi) := \frac{1}{t}\sum_{i=1}^{t} \frac{r_i(a_i)\cdot \mathbb{1}\{\pi(x_i) = a_i\}}{p_i(a_i)}. \tag{1}$$

[3] Extension to VC classes is simple using standard arguments.

[4] We have defined empirical cumulative regret as being relative to $\pi_\star$, rather than to the empirical reward maximizer $\arg\max_{\pi\in\Pi} \sum_{t=1}^{T} r_t(\pi(x_t))$. However, in the i.i.d. setting, the two do not differ by more than $O(\sqrt{T\ln(N/\delta)})$ with probability at least $1-\delta$.

This technique can be viewed as mapping $H_t \mapsto \mathsf{IPS}(H_t)$ of interaction records $(x, a, r(a), p(a))$ to context/reward vector pairs $(x, \hat{r})$, where $\hat{r} \in \mathbb{R}_+^A$ is a fictitious reward vector that assigns to the chosen action a a scaled reward $r(a)/p(a)$ (possibly greater than one), and assigns to all other actions zero rewards. This transformation $\mathsf{IPS}(H_t)$ is detailed in Algorithm 3 (in Appendix A); we may equivalently define $\widehat{\mathcal{R}}_t$ by $\widehat{\mathcal{R}}_t(\pi) := t^{-1} \sum_{(x,\hat{r}) \in \mathsf{IPS}(H_t)} \hat{r}(\pi(x))$. It is easy to verify that $\mathbb{E}[\hat{r}(\pi(x))|(x, r)] = r(\pi(x))$, as $p(a)$ is indeed the agent's probability (conditioned on (x, r)) of picking action a. This implies $\widehat{\mathcal{R}}_t(\pi)$ is an unbiased estimator for any history H_t.

Let $\pi_t := \arg\max_{\pi \in \Pi} \widehat{\mathcal{R}}_t(\pi)$ denote a policy that maximizes the expected reward estimate based on inverse propensity scoring with history H_t (π_0 can be arbitrary), and let $\widehat{\mathrm{Reg}}_t(\pi) := \widehat{\mathcal{R}}_t(\pi_t) - \widehat{\mathcal{R}}_t(\pi)$ denote *estimated regret* relative to π_t. Note that $\widehat{\mathrm{Reg}}_t(\pi)$ is generally *not* an unbiased estimate of $\mathrm{Reg}(\pi)$, because π_t is not always $\pi_\star$.

2.3 Optimization Oracle

One natural mode for accessing the set of policies Π is enumeration, but this is impractical in general. In this work, we instead only acceess Π via an optimization oracle which corresponds to a cost-sensitive learner. Following (Dudík et al., 2011a), we call this oracle AMO[5].

Definition 1. *For a set of policies* Π*, the* $\arg\max$ *oracle (*AMO*) is an algorithm, which for any sequence of context and reward vectors,* $(x_1, r_1), (x_2, r_2), \ldots, (x_t, r_t) \in X \times \mathbb{R}_+^A$*, returns*

$$\arg\max_{\pi \in \Pi} \sum_{\tau=1}^{t} r_\tau(\pi(x_\tau)).$$

2.4 Projections and Smoothing

In each round, our algorithm chooses an action by randomly drawing a policy π from a distribution over Π, and then picking the action $\pi(x)$ recommended by π on the current context x. This is equivalent to drawing an action according to $Q(a|x) := \sum_{\pi \in \Pi : \pi(x) = a} Q(\pi)$, $\forall a \in A$. For keeping the variance of reward estimates from IPS in check, it is desirable to prevent the probability of any action from being too small. Thus, as in previous work, we also use a smoothed projection $Q^\mu(\cdot|x)$ for $\mu \in [0, 1/K]$, $Q^\mu(a|x) := (1 - K\mu) \sum_{\pi \in \Pi : \pi(x) = a} Q(\pi) + \mu$, $\forall a \in A$. Every action has probability at least μ under $Q^\mu(\cdot|x)$.

For technical reasons, our algorithm maintains non-negative weights $Q \in \Delta^\Pi$ over policies that sum to at most one, but not necessarily equal to one; hence, we put any remaining mass on a default policy $\bar{\pi} \in \Pi$ to obtain a legitimate probability distribution over policies $\tilde{Q} = Q + \left(1 - \sum_{\pi \in \Pi} Q(\pi)\right) \mathbb{1}_{\bar{\pi}}$. We then pick an action from the smoothed projection $\tilde{Q}^\mu(\cdot|x)$ of $\tilde{Q}$ as above. This sampling procedure $\mathsf{Sample}(x, Q, \bar{\pi}, \mu)$ is detailed in Algorithm 4 (in Appendix A).

3 Algorithm and Main Results

Our algorithm ($\mathsf{ILOVETOCONBANDITS}$) is an epoch-based variant of the $\mathsf{Randomized\ UCB}$ algorithm of Dudík et al. (2011a) and is given in Algorithm 1. Like $\mathsf{Randomized\ UCB}$, $\mathsf{ILOVETOCONBANDITS}$ also solves an optimization problem (OP) to obtain a distribution over policies to sample from (Step 7), but does so on an *epoch schedule*, *i.e.*, only on certain pre-specified rounds $\tau_1, \tau_2, \ldots$. The only requirement of the epoch schedule is that the length of epoch m is bounded as $\tau_{m+1} - \tau_m = O(\tau_m)$. For simplicity, we assume $\tau_{m+1} \leq 2\tau_m$ for $m \geq 1$, and $\tau_1 = O(1)$.

The crucial step here is solving (OP). This problem is very similar to the one in (Dudík et al., 2011a), and our coordinate descent algorithm in Section 3.1 gives a constructive proof that the problem is feasible. As in (Dudík et al., 2011a), we have the following regret bound:

[5] Cost-sensitive learners often need a cost instead of reward, in which case we use $c_t = \mathbb{1} - r_t$.

Theorem 2. *Suppose it is possible to solve the optimization problem (*OP*). With probability at least* $1-\delta$*, the regret of* ILOVETOCONBANDITS *after* T *rounds is*

$$O\left(\sqrt{KT\ln(T|\Pi|/\delta)} + K\ln(T|\Pi|/\delta)\right).$$

Algorithm 1 Importance-weighted LOw-Variance Epoch-Timed Oracleized CONtextual BANDITS algorithm (ILOVETOCONBANDITS)

input Epoch schedule $0 = \tau_0 < \tau_1 < \tau_2 < \cdots$, allowed failure probability $\delta \in (0,1)$.

Initial weights $Q_0 := \mathbf{0} \in \Delta^{\Pi}$, initial epoch $m := 1$.
Define $\mu_m := \min\{1/2K, \sqrt{\ln(16\tau_m^2|\Pi|/\delta)/(K\tau_m)}\}$ for all $m \geq 0$.
for round $t = 1, 2, \ldots$ **do**
  Observe context $x_t \in X$.
  $(a_t, p_t(a_t)) := \mathsf{Sample}(x_t, Q_{m-1}, \pi_{\tau_m - 1}, \mu_{m-1})$.
  Select action a_t and observe reward $r_t(a_t) \in [0,1]$.
  if $t = \tau_m$ **then**
    Let Q_m be the solution to (OP) with history H_t and minimum probability μ_m.
    $m := m + 1$.
  end if
end for

Optimization Problem (OP)

Given a history H_t and minimum probability μ_m, define $b_\pi := \frac{\widehat{\mathrm{Reg}}_t(\pi)}{\psi\mu_m}$ for $\psi := 100$, and find $Q \in \Delta^{\Pi}$ such that

$$\sum_{\pi\in\Pi} Q(\pi) b_\pi \;\leq\; 2K \tag{2}$$

$$\forall \pi \in \Pi: \;\widehat{\mathbb{E}}_{x\sim H_t}\left[\frac{1}{Q^{\mu_m}(\pi(x)|x)}\right] \;\leq\; 2K + b_\pi. \tag{3}$$

3.1 Solving (OP) via Coordinate Descent

We now present a coordinate descent algorithm to solve (OP). The pseudocode is given in Algorithm 2. Our analysis, as well as the algorithm itself, are based on a potential function which we use to measure progress. The algorithm can be viewed as a form of coordinate descent applied to this same potential function. The main idea of our analysis is to show that this function decreases substantially on every iteration of this algorithm; since the function is nonnegative, this gives an upper bound on the total number of iterations as expressed in the following theorem.

Theorem 3. *Algorithm 2 (with* $Q_{\text{init}} := \mathbf{0}$*) halts in* $\leq \frac{4\ln(1/(K\mu))}{\mu}$ *iterations, and outputs a solution* Q *to (*OP*).*

3.2 Using an Optimization Oracle

We now show how to implement Algorithm 2 via AMO (c.f. Section 2.3).

Lemma 1. *Algorithm 2 can be implemented using one call to* AMO *before the loop is started, and one call for each iteration of the loop thereafter.*

Algorithm 2 Coordinate Descent Algorithm

Require: History H_t, minimum probability μ, initial weights $Q_{\text{init}} \in \Delta^{\Pi}$.

1: Set $Q := Q_{\text{init}}$.
2: **loop**
3: Define, for all $\pi \in \Pi$,

$$\begin{aligned} V_\pi(Q) &= \widehat{\mathbb{E}}_{x\sim H_t}[1/Q^\mu(\pi(x)|x)] \\ S_\pi(Q) &= \widehat{\mathbb{E}}_{x\sim H_t}\left[1/(Q^\mu(\pi(x)|x))^2\right] \\ D_\pi(Q) &= V_\pi(Q) - (2K + b_\pi). \end{aligned}$$

4: **if** $\sum_\pi Q(\pi)(2K + b_\pi) > 2K$ **then**
5: Replace Q by cQ, where

$$c := \frac{2K}{\sum_\pi Q(\pi)(2K + b_\pi)} < 1. \tag{4}$$

6: **end if**
7: **if** there is a policy π for which $D_\pi(Q) > 0$ **then**
8: Add the (positive) quantity

$$\alpha_\pi(Q) = \frac{V_\pi(Q) + D_\pi(Q)}{2(1 - K\mu)S_\pi(Q)}$$

to $Q(\pi)$ and leave all other weights unchanged.
9: **else**
10: Halt and output the current set of weights Q.
11: **end if**
12: **end loop**

Proof. At the very beginning, before the loop is started, we compute the best empirical policy so far, π_t, by calling AMO on the sequence of historical contexts and estimated reward vectors; *i.e.*, on $(x_\tau, \hat{r}_\tau)$, for $\tau = 1, 2, \ldots, t$.

Next, we show that each iteration in the loop of Algorithm 2 can be implemented via one call to AMO. Going over the pseudocode, first note that operations involving Q in step Step 4 can be performed efficiently since Q has sparse support. Note that the definitions in step Step 3 don't actually need to be computed for all policies $\pi \in \Pi$, as long as we can identify a policy π for which $D_\pi(Q) > 0$. We can identify such a policy using one call to AMO as follows.

First, note that for any policy π, we have

$$V_\pi(Q) = \widehat{\mathbb{E}}_{x\sim H_t}\left[\frac{1}{Q^\mu(\pi(x)|x)}\right] = \frac{1}{t}\sum_{\tau=1}^{t}\frac{1}{Q^\mu(\pi(x_\tau)|x_\tau)},$$

and

$$b_\pi = \frac{\widehat{\mathrm{Reg}}_t(\pi)}{\psi\mu} = \frac{\widehat{\mathcal{R}}_t(\pi_t)}{\psi\mu} - \frac{1}{\psi\mu t}\sum_{\tau=1}^{t}\hat{r}_\tau(\pi(x_\tau)).$$

Now consider the sequence of historical contexts and reward vectors, $(x_\tau, \tilde{r}_\tau)$ for $\tau = 1, 2, \ldots, t$, where for any action a we define

$$\tilde{r}_\tau(a) := \frac{1}{t}\left(\frac{\psi\mu}{Q^\mu(a|x_\tau)} + \hat{r}_t(a)\right). \tag{5}$$

It is easy to check that

$$D_\pi(Q) = \frac{1}{\psi\mu}\sum_{\tau=1}^{t}\tilde{r}_\tau(\pi(x_\tau)) - \left(2K + \frac{\widehat{\mathcal{R}}_t(\pi_t)}{\psi\mu}\right).$$

Since $2K + \frac{\widehat{\mathcal{R}}_t(\pi_t)}{\psi\mu}$ is a constant independent of π, we have

$$\arg\max_{\pi\in\Pi} D_\pi(Q) = \arg\max_{\pi\in\Pi} \sum_{\tau=1}^{t} \tilde{r}_\tau(\pi(x_\tau)),$$

and hence, calling AMO once on the sequence $(x_\tau, \tilde{r}_\tau)$ for $\tau = 1, 2, \ldots, t$, we obtain a policy that maximizes $D_\pi(Q)$ and thereby identify a policy for which $D_\pi(Q) > 0$, if one exists. □

3.3 Epoch Schedule

Theorem 3 shows that Algorithm 2 solves (OP) with $\tilde{O}(\sqrt{t})$ calls to AMO in round t. Thus, if we use the epoch schedule $\tau_m = m$ (*i.e.*, run Algorithm 2 in every round), then we get a total of $\tilde{O}(T^{3/2})$ calls to AMO over all T rounds. This number can be dramatically reduced using a more carefully chosen epoch schedule.

Lemma 2. *For the epoch schedule $\tau_m := 2^{m-1}$, the total number of calls to* AMO *is $\tilde{O}(\sqrt{KT})$.*

Proof. The epoch schedule satisfies the requirement $\tau_{m+1} \leq 2\tau_m$. With this epoch schedule, Algorithm 2 is run only $O(\log T)$ times over T rounds, leading to $\tilde{O}(\sqrt{KT})$ total calls to AMO over the entire period. □

3.4 Warm Start

We now present a different technique to reduce the number of calls to AMO. This is based on the observation that practically speaking, it seems terribly wasteful, at the start of a new epoch, to throw out the results of all of the preceding computations and to begin yet again from nothing. Instead, intuitively, we expect computations to be more moderate if we begin again where we left off last, *i.e.*, a "warm-start" approach. Here, when Algorithm 2 is called at the end of epoch m, we use $Q_{\text{init}} := Q_{m-1}$ (the previously computed weights) rather than $\mathbf{0}$.

We can combine warm-start with a different epoch schedule to guarantee $\tilde{O}(\sqrt{KT})$ total calls to AMO, spread across $O(\sqrt{T})$ calls to Algorithm 2.

Lemma 3. *Define the epoch schedule $(\tau_1, \tau_2) := (3, 5)$ and $\tau_m := m^2$ for $m \geq 3$ (this satisfies $\tau_{m+1} \leq 2\tau_m$). With high probability, the warm-start variant of Algorithm 1 makes $\tilde{O}(\sqrt{KT})$ calls to* AMO *over T rounds and $O(\sqrt{T})$ calls to Algorithm 2.*

3.5 A Lower Bound on the Support Size

So far we have seen various ways to solve the optimization problem (OP), with corresponding bounds on the number of calls to AMO. An attractive feature of the coordinate descent algorithm, Algorithm 2, is that the number of oracle calls is directly related to the number of policies in the support of Q_m. Specifically, for the doubling schedule of Section 3.3, Theorem 3 implies that we never have non-zero weights for more than $\frac{4\ln(1/(K\mu_m))}{\mu_m}$ policies in epoch m. Similarly, the total number of oracle calls for the warm-start approach in Section 3.4 bounds the total number of policies which ever have non-zero weight over all T rounds. The support size of the distributions Q_m in Algorithm 1 is crucial to the computational complexity of sampling an action (Step 4 of Algorithm 1).

In this section, we demonstrate a lower bound showing that it is not possible to construct substantially sparser distributions that also satisfy the low-variance constraint (3) in the optimization problem (OP). To formally define the lower bound, fix an epoch schedule $0 = \tau_0 < \tau_1 < \tau_2 < \cdots$ and consider the following set of non-negative vectors over policies:

$$\mathcal{Q}_m := \{Q \in \Delta^\Pi : Q \text{ satisfies Eq. (3) in round } \tau_m\}.$$

(The distribution Q_m computed by Algorithm 1 is in $\mathcal{Q}_m$.) Let us also define $\text{supp}(Q)$ to be the set of policies where Q puts non-zero entries, with $|\text{supp}(Q)|$ denoting the cardinality of this set. Then we have the following lower bound.

Theorem 4. *For any epoch schedule $0 = \tau_0 < \tau_1 < \tau_2 < \cdots$ and any $M \in \mathbb{N}$ sufficiently large, there exists a distribution $\mathcal{D}$ over $X \times [0,1]^A$ and a policy class Π such that, with probability at least $1-\delta$,*

$$\inf_{\substack{m\in\mathbb{N}:\\ \tau_m \geq \tau_M/2}} \inf_{Q\in\mathcal{Q}_m} |\mathrm{supp}(Q)| = \Omega\left(\sqrt{\frac{K\tau_M}{\ln(|\Pi|\tau_M/\delta)}}\right).$$

The proof of the theorem is deferred to Appendix E. In the context of our problem, this lower bound shows that the bounds in Lemmas 2 and 3 are unimprovable, since the number of calls to AMO is at least the size of the support, given our mode of access to Π.

4 Regret Analysis

In this section, we outline the regret analysis for our algorithm ILOVETOCONBANDITS, with details deferred to Appendix B and Appendix C.

The deviations of the policy reward estimates $\widehat{\mathcal{R}}_t(\pi)$ are controlled by (a bound on) the variance of each term in Eq. (1): essentially the left-hand side of Eq. (3) from (OP), except with $\widehat{\mathbb{E}}_{x\sim H_t}[\cdot]$ replaced by $\mathbb{E}_{x\sim\mathcal{D}_X}[\cdot]$. Resolving this discrepancy is handled using deviation bounds, so Eq. (3) holds with $\mathbb{E}_{x\sim\mathcal{D}_X}[\cdot]$, with worse right-hand side constants.

The rest of the analysis, which deviates from that of Randomized UCB, compares the expected regret $\mathrm{Reg}(\pi)$ of any policy π with the estimated regret $\widehat{\mathrm{Reg}}_t(\pi)$ using the variance constraints Eq. (3):

Lemma 4 (Informally)**.** *With high probability, for each m such that $\tau_m \geq \tilde{O}(K\log|\Pi|)$, each round t in epoch m, and each $\pi \in \Pi$, $\mathrm{Reg}(\pi) \leq 2\widehat{\mathrm{Reg}}_t(\pi) + O(K\mu_m)$.*

This lemma can easily be combined with the constraint Eq. (2) from (OP): since the weights Q_{m-1} used in any round t in epoch m satisfy $\sum_{\pi\in\Pi} Q_{m-1}\widehat{\mathrm{Reg}}_{\tau_m-1}(\pi) \leq 2K\mu_{\tau_m-1}$, we obtain a bound on the (conditionally) expected regret in round t using the above lemma: with high probability,

$$\sum_{\pi\in\Pi} \widetilde{Q}_{m-1}\,\mathrm{Reg}(\pi) \leq O(K\mu_{m-1}).$$

Summing these terms up over all T rounds and applying martingale concentration gives the final regret bound in Theorem 2.

5 Analysis of the Optimization Algorithm

In this section, we give a sketch of the analysis of our main optimization algorithm for computing weights Q_m on each epoch as in Algorithm 2. As mentioned in Section 3.1, this analysis is based on a potential function.

Since our attention for now is on a single epoch m, here and in what follows, when clear from context, we drop m from our notation and write simply $\tau = \tau_m$, $\mu = \mu_m$, etc. Let $\mathcal{U}_A$ be the uniform distribution over the action set A. We define the following potential function for use on epoch m:

$$\Phi_m(Q) = \tau\mu\left(\frac{\widehat{\mathbb{E}}_x[\mathrm{RE}\,(\mathcal{U}_A\|Q^\mu(\cdot\mid x))]}{1-K\mu} + \frac{\sum_{\pi\in\Pi} Q(\pi)b_\pi}{2K}\right). \tag{6}$$

The function in Eq. (6) is defined for all vectors $Q \in \Delta^\Pi$. Also, $\mathrm{RE}\,(p\|q)$ denotes the unnormalized relative entropy between two nonnegative vectors p and q over the action space (or any set) A:

$$\mathrm{RE}\,(p\|q) = \sum_{a\in A}(p_a\ln(p_a/q_a) + q_a - p_a).$$

This number is always nonnegative. Here, $Q^\mu(\cdot|x)$ denotes the "distribution" (which might not sum to 1) over A induced by Q^μ for context x as given in Section 2.4. Thus, ignoring constants, this potential function is a combination of two terms: The first measures how far from uniform are the distributions induced by Q^μ, and the second is an estimate of expected regret under Q since b_π is proportional to the empirical regret of π. Making Φ_m small thus encourages Q to choose actions as uniformly as possible while also incurring low regret — exactly the aims of our algorithm. The constants that appear in this definition are for later mathematical convenience.

For further intuition, note that, by straightforward calculus, the partial derivative $\partial\Phi_m/\partial Q(\pi)$ is roughly proportional to the variance constraint for π given in Eq. (3) (up to a slight mismatch of constants). This shows that if this constraint is not satisfied, then $\partial\Phi_m/\partial Q(\pi)$ is likely to be negative, meaning that Φ_m can be decreased by increasing $Q(\pi)$. Thus, the weight vector Q that minimizes Φ_m satisfies the variance constraint for every policy π. It turns out that this minimizing Q also satisfies the low regret constraint in Eq. (2), and also must sum to at most 1; in other words, it provides a complete solution to our optimization problem. Algorithm 2 does not fully minimize Φ_m, but it is based roughly on coordinate descent. This is because in each iteration one of the weights (coordinate directions) $Q(\pi)$ is increased. This weight is one whose corresponding partial derivative is large and negative.

To analyze the algorithm, we first argue that it is correct in the sense of satisfying the required constraints, provided that it halts.

Lemma 5. *If Algorithm 2 halts and outputs a weight vector Q, then the constraints Eq. (3) and Eq. (2) must hold, and furthermore the sum of the weights $Q(\pi)$ is at most 1.*

The proof is rather straightforward: Following step 4, Eq. (2) must hold, and also the weights must sum to 1. And if the algorithm halts, then $D_\pi(Q) \le 0$ for all π, which is equivalent to Eq. (3).

What remains is the more challenging task of bounding the number of iterations until the algorithm does halt. We do this by showing that significant progress is made in reducing Φ_m on every iteration. To begin, we show that scaling Q as in step 4 cannot cause Φ_m to increase.

Lemma 6. *Let Q be a weight vector such that $\sum_\pi Q(\pi)(2K + b_\pi) > 2K$, and let c be as in Eq. (4). Then $\Phi_m(cQ) \le \Phi_m(Q)$.*

Proof sketch. We consider $\Phi_m(cQ)$ as a function of c, and argue that its derivative (with respect to c) at the value of c given in the lemma statement is always nonnegative. Therefore, by convexity, it is nondecreasing for all values exceeding c. Since $c < 1$, this proves the lemma. □

Next, we show that substantial progress will be made in reducing Φ_m each time that step 8 is executed.

Lemma 7. *Let Q denote a set of weights and suppose, for some policy π, that $D_\pi(Q) > 0$. Let Q' be a new set of weights which is an exact copy of Q except that $Q'(\pi) = Q(\pi) + \alpha$ where $\alpha = \alpha_\pi(Q) > 0$. Then*

$$\Phi_m(Q) - \Phi_m(Q') \ge \frac{\tau\mu^2}{4(1 - K\mu)}. \tag{7}$$

Proof sketch. We first compute exactly the change in potential for general α. Next, we apply a second-order Taylor approximation, which is maximized by the α used in the algorithm. The Taylor approximation, for this α, yields a lower bound which can be further simplified using the fact that $Q^\mu(a|x) \ge \mu$ always, and our assumption that $D_\pi(Q) > 0$. This gives the bound stated in the lemma. □

So step 4 does not cause Φ_m to increase, and step 8 causes Φ_m to decrease by at least the amount given in Lemma 7. This immediately implies Theorem 3: for $Q_{\text{init}} = \mathbf{0}$, the initial potential is bounded by $\tau\mu \ln(1/(K\mu))/(1 - K\mu)$, and it is never negative, so the number of times step 8 is executed is bounded by $4\ln(1/(K\mu))/\mu$ as required.

5.1 Epoching and Warm Start

As shown in Section 2.3, the bound on the number of iterations of the algorithm from Theorem 3 also gives a bound on the number of times the oracle is called. To reduce the number of oracle calls, one

Algorithm	ϵ-greedy	Explore-first	Bagging	LinUCB	Online Cover	Supervised
P.V. Loss	0.095	0.081	0.059	0.128	0.053	0.051
Searched	$0.02 = \epsilon$	2×10^5 first	16 bags	10^3 dim, minibatch-10	cover $n = 1$	nothing
Seconds	22	5.6	339	212×10^3	17	6.9

Table 1: Progressive validation loss of various algorithm on RCV1.

approach is the "doubling trick" of Section 3.3, which enables us to bound the total combined number of iterations of Algorithm 2 in the first T rounds is only $\tilde{O}(\sqrt{KT})$. This means that the average number of calls to the arg-max oracle is only $\tilde{O}(\sqrt{K/T})$ per round, meaning that the oracle is called far less than once per round, and in fact, at a vanishingly low rate.

We now turn to warm-start approach of Section 3.4, where in each epoch $m+1$ we initialize the coordinate descent algorithm with $Q_{\text{init}} = Q_m$, i.e. the weights computed in the previous epoch m. To analyze this, we bound how much the potential changes from $\Phi_m(Q_m)$ at the end of epoch m to $\Phi_{m+1}(Q_m)$ at the very start of epoch $m+1$. This, combined with our earlier results regarding how quickly Algorithm 2 drives down the potential, we are able to get an overall bound on the total number of updates across T rounds.

Lemma 8. *Let M be the largest integer for which $\tau_{M+1} \leq T$. With probability at least $1 - 2\delta$, for all (not too small) T, the total epoch-to-epoch increase in potential is*

$$\sum_{m=1}^{M} (\Phi_{m+1}(Q_m) - \Phi_m(Q_m)) \leq \tilde{O}\left(\sqrt{\frac{T}{K}}\right),$$

where M is the largest integer for which $\tau_{M+1} \leq T$.

Proof sketch. The potential function, as written in Eq. (6), naturally breaks into two pieces whose epoch-to-epoch changes can be bounded separately. Changes affecting the relative entropy term on the left can be bounded, regardless of Q_m, by taking advantage of the manner in which these distributions are smoothed. For the other term on the right, it turns out that these epoch-to-epoch changes are related to statistical quantities which can be bounded with high probability. Specifically, the total change in this term is related first to how the estimated reward of the empirically best policy compares to the expected reward of the optimal policy; and second, to how the reward received by our algorithm compares to that of the optimal reward. From our regret analysis, we are able to show that both of these quantities will be small with high probability. □

Thus, the total amount that the potential increases across T rounds is at most $\tilde{O}(\sqrt{T/K})$. On the other hand, Lemma 7 shows that each time Q is updated by Algorithm 2 the potential decreases by at least $\tilde{\Omega}(1/K)$. Therefore, the total number of updates of the algorithm totaled over all T rounds is at most $\tilde{O}(\sqrt{KT})$. For instance, if we use $(\tau_1, \tau_2) := (3, 5)$ and $\tau_m := m^2$ for $m \geq 3$, then the weight vector Q is only updated about $\sqrt{T}$ times in T rounds, and on each of those rounds, Algorithm 2 requires $\tilde{O}(\sqrt{K})$ iterations, on average. This proves Lemma 3.

6 Experimental Evaluation

In this section we evaluate a variant of Algorithm 1 against several baselines. While Algorithm 1 is significantly more efficient than many previous approaches, the overall computational complexity is still at least $\tilde{O}(T^{1.5})$ excluding the running time of the oracle, since each invocation of the oracle at round t of Algorithm 1 needs $O(t)$ time in order to create t cost-sensitive classification (henceforth CSC) examples through the transformation (5). After further accounting for the running time of the oracle itself, this seems markedly larger than the complexity of an ordinary supervised learning problem where it is typically

possible to perform an $O(1)$ complexity update upon receiving a fresh example using online learning algorithms.

A natural solution is to use an "online" oracle which is stateful and accepts examples one by one. An online CSC oracle takes as input a weighted example and returns a predicted class (corresponding to one of K actions in our setting). Since the oracle is stateful, it remembers and uses examples from all previous calls in answering questions, thereby reducing the complexity of each oracle invocation to $O(1)$ as in supervised learning. Using several of these oracles, we can efficiently track a distribution over good policies and sample from it. The detailed pseduo-code describing this high-level idea (which we call Online Cover) is provided in Algorithm 5 in Appendix F. The algorithm maintains a uniform distribution over a fixed number n of policies where n is a parameter of the algorithm. Upon receiving a fresh example, it updates all n policies with the suitable CSC examples (5). The specific CSC algorithm we use is a reduction to squared regression described in Algorithms 4 and 5 of Beygelzimer and Langford (2009), which is amenable to online updates. Our implementation is public and will be referenced in the final draft.

Due to lack of public datasets for contextual bandit problems, we use a simple supervised-to-contextual-bandit transformation (Dudík et al., 2011b) on the CCAT document classification problem in RCV1 (Lewis et al., 2004). This is a dataset of $781,265$ examples, with a total of $d = 47,152$ TF-IDF features. In the data transformation, we treated the class labels as actions, and one minus 0/1-loss as the reward. Using this dataset, we provide a proof-of-concept that this approach can be empirically very effective. Our evaluation criteria is progressive validation (Blum et al., 1999) on 0/1 loss. We compare ourselves to several plausible baselines, whose results are summarized in Table 1. All baselines and our online algorithm take advantage of linear representations which are known to work well on this dataset. For each algorithm we report the result for the best parameter settings which are mentioned in Table 1.

1. ϵ-greedy (Sutton and Barto, 1998) explores randomly with probability ϵ and otherwise exploits.

2. Explore-first is a common variant where you explore uniformly over actions before switching to an exploit-only phase.

3. A less common but powerful baseline is based on bagging. In essence, multiple algorithms can be trained with examples sampled with replacement. The different predictions of these different learned predictors yield a distribution over actions from which we can sample effectively.

4. Another reasonable baseline is based on Thompson sampling (Chapelle and Li, 2011; Li, 2013) or linear UCB (Auer, 2002; Li et al., 2010), the latter being quite effective in past evaluations (Li et al., 2010; Chapelle and Li, 2011). It is impractical to run vanilla LinUCB on this problem due to the high-dimensionality which makes matrix inversions impractical. We report results for the algorihm run after doing a dimesionality reduction via random projections to 1000 dimensions. Even then the algorithm required 59 hours[6], and simpler variants such as using diagonal matrix approximations or infrequent updating performed substantially worse.

5. Finally, our algorithm achieves the best loss of 0.0530. Somewhat surprisingly, the minimum occurs for us with a cover set of size 1—apparently for this problem the small decaying amount of uniform random sampling imposed is adequate exploration. Prediction performance is similar with a larger covering set.

All baselines except for LinUCB are implemented as a simple modification of Vowpal Wabbit, an open source online learning system. All reported results use default parameters where not otherwise specified. The contextual bandit learning algorithms all operate in a doubly robust mode similar to the reward estimates formed by our algorithm.

Because this is actually a fully supervised dataset, we can apply a fully supervised online multiclass algorithm to solve it. We use a simple one-against-all implementation to reduce this to binary classification, yielding an error rate of 0.051 which is competitive with the best previously reported results. This result is effectively a lower bound on the quality of the solution we can hope to achieve with algorithms

[6]The linear algebra routines are based on Intel MKL package.

using only partial information. Our algorithm nearly achieves this lower bound with a running time only a factor of 2.5 slower. Hence on this dataset, very little further algorithmic improvement is possible.

7 Conclusions

In this paper we have presented the first practical algorithm to our knowledge that attains the statistically optimal regret guarantee and is computationally efficient in the setting of general policy classes. A remarkable feature of the algorithm is that the total number of oracle calls over all T rounds is sublinear—a remarkable improvement over previous works in this setting. We believe that the online variant of the approach which we implemented in our experiments has the right practical flavor for a scalable solution to the contextual bandit problem. In future work, it would be interesting to directly analyze the Online Cover algorithm.

References

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal of Computing*, 32(1):48–77, 2002.

Alina Beygelzimer and John Langford. The offset tree for learning with partial labels. In *KDD*, 2009.

Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandit algorithms with supervised learning guarantees. In *AISTATS*, 2011.

Avrim Blum, Adam Kalai, and John Langford. Beating the holdout: Bounds for k-fold and progressive cross-validation. In *COLT*, 1999.

Olivier Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. In *NIPS*, 2011.

Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandits with linear payoff functions. In *AISTATS*, 2011.

Miroslav Dudík, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. In *UAI*, 2011a.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *ICML*, 2011b.

David P. Helmbold and Robert E. Schapire. Predicting nearly as well as the best pruning of a decision tree. *Machine Learning*, 27(1):51–68, 1997.

John Langford. Interactive machine learning, January 2014. URL `http://hunch.net/~jl/projects/interactive/index.html`.

John Langford and Tong Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *NIPS*, 2007.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397, 2004.

Lihong Li. Generalized Thompson sampling for contextual bandits. *CoRR*, abs/1310.7163, 2013.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, 2010.

Algorithm 3 IPS(H)

input History $H \subseteq X \times A \times [0,1] \times [0,1]$.
output Data set $S \subseteq X \times \mathbb{R}_+^A$.

Initialize data set $S := \emptyset$.
for each $(x, a, r(a), p(a)) \in H$ **do**
  Create fictitious rewards $\hat{r} \in \mathbb{R}_+^A$ with $\hat{r}(a) = r(a)/p(a)$ and $\hat{r}(a') = 0$ for all $a' \in A \setminus \{a\}$.
  $S := S \cup \{(x, \hat{r})\}$.
end for
return S.

Algorithm 4 Sample($x, Q, \bar{\pi}, \mu$)

input Context $x \in X$, weights $Q \in \Delta^{\Pi}$, default policy $\bar{\pi} \in \Pi$, minimum probability $\mu \in [0, 1/K]$.
output Selected action $\bar{a} \in A$ and probability $\bar{p} \in [\mu, 1]$.

Let $\tilde{Q} := Q + (1 - \sum_{\pi \in \Pi} Q(\pi)) \mathbb{1}_{\bar{\pi}}$
(so $\sum_{\pi \in \Pi} \tilde{Q}(\pi) = 1$).
Randomly draw action $\bar{a} \in A$ using the distribution

$$\tilde{Q}^{\mu}(a|x) := (1 - K\mu) \sum_{\substack{\pi \in \Pi: \\ \pi(x) = a}} \tilde{Q}(\pi) + \mu, \quad \forall a \in A.$$

Let $\bar{p}(\bar{a}) := \tilde{Q}^{\mu}(\bar{a}|x)$.
return $(\bar{a}, \bar{p}(\bar{a}))$.

H. Brendan McMahan and Matthew Streeter. Tighter bounds for multi-armed bandits with expert advice. In *COLT*, 2009.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning, an introduction*. MIT Press, 1998.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3–4):285–294, 1933.

A Omitted Algorithm Details

Algorithm 3 and Algorithm 4 give the details of the inverse propensity scoring transformation IPS and the action sampling procedure Sample.

B Deviation Inequalities

B.1 Freedman's Inequality

The following form of Freedman's inequality for martingales is from Beygelzimer et al. (2011).

Lemma 9. *Let $X_1, X_2, \ldots, X_T$ be a sequence of real-valued random variables. Assume for all $t \in \{1, 2, \ldots, T\}$, $X_t \leq R$ and $\mathbb{E}[X_t | X_1, \ldots, X_{t-1}] = 0$. Define $S := \sum_{t=1}^{T} X_t$ and $V := \sum_{t=1}^{T} \mathbb{E}[X_t^2 | X_1, \ldots, X_{t-1}]$. For any $\delta \in (0,1)$ and $\lambda \in [0, 1/R]$, with probability at least $1 - \delta$,*

$$S \leq (e-2)\lambda V + \frac{\ln(1/\delta)}{\lambda}.$$

B.2 Variance Bounds

Fix the epoch schedule $0 = \tau_0 < \tau_1 < \tau_2 < \cdots$.

Define the following for any probability distribution P over Π, $\pi \in \Pi$, and $\mu \in [0, 1/K]$:

$$V(P, \pi, \mu) := \mathbb{E}_{x \sim \mathcal{D}_X}\left[\frac{1}{P^\mu(\pi(x)|x)}\right], \tag{8}$$

$$\widehat{V}_m(P, \pi, \mu) := \widehat{\mathbb{E}}_{x \sim H_{\tau_m}}\left[\frac{1}{P^\mu(\pi(x)|x)}\right]. \tag{9}$$

The proof of the following lemma is essentially the same as that of Theorem 6 from Dudík et al. (2011a).

Lemma 10. *Fix any $\mu_m \in [0, 1/K]$ for $m \in \mathbb{N}$. For any $\delta \in (0, 1)$, with probability at least $1 - \delta$,*

$$V(P, \pi, \mu_m) \le 6.4\widehat{V}_m(P, \pi, \mu_m) + \frac{75(1 - K\mu_m)\ln|\Pi|}{\mu_m^2 \tau_m} + \frac{6.3\ln(2|\Pi|^2 m^2/\delta)}{\mu_m \tau_m}$$

for all probability distributions P over Π, all $\pi \in \Pi$, and all $m \in \mathbb{N}$. In particular, if

$$\mu_m \ge \sqrt{\frac{\ln(2|\Pi|m^2/\delta)}{K\tau_m}}, \qquad \tau_m \ge 4K\ln(2|\Pi|m^2/\delta),$$

then

$$V(P, \pi, \mu_m) \le 6.4\widehat{V}_m(P, \pi, \mu_m) + 81.3K.$$

Proof sketch. By Bernstein's (or Freedman's) inequality and union bounds, for any choice of $N_m \in \mathbb{N}$ and $\lambda_m \in [0, \mu_m]$ for $m \in \mathbb{N}$, the following holds with probability at least $1 - \delta$:

$$V(P, \pi, \mu_m) - \widehat{V}_m(P, \pi, \mu_m) \le \frac{(e-2)\lambda_m V(P, \pi, \mu_m)}{\mu_m} + \frac{\ln(|\Pi|^{N_m+1}2m^2/\delta)}{\lambda_m \tau_m}$$

all N_m-point distributions P over Π, all $\pi \in \Pi$, and all $m \in \mathbb{N}$. Here, an N-point distribution over Π is a distribution of the form $\frac{1}{N}\sum_{i=1}^N \mathbb{1}_{\pi_i}$ for $\pi_1, \pi_2, \ldots, \pi_N \in \Pi$. We henceforth condition on this $\ge 1 - \delta$ probability event (for choices of N_m and λ_m to be determined).

Using the probabilistic method, it can be shown that for any probability distribution P over Π, any $\pi \in \Pi$, any $\mu_m \in [0, 1/K]$, and any $c_m > 0$, there exists an N_m-point distribution $\widetilde{P}$ over Π such that

$$\begin{aligned}\big(V(P, \pi, \mu_m) - V(\widetilde{P}, \pi, \mu_m)\big) + c_m\big(\widehat{V}_m(\widetilde{P}, \pi, \mu_m) - \widehat{V}_m(P, \pi, \mu_m)\big) \\ \le \gamma_{N_m, \mu_m}\big(V(P, \pi, \mu_m) + c_m\widehat{V}_m(P, \pi, \mu_m)\big)\end{aligned}$$

where $\gamma_{N,\mu} := \sqrt{(1 - K\mu)/(N\mu)} + 3(1 - K\mu)/(N\mu)$.

Combining the displayed inequalities (using $c_m := 1/(1 - (e-2)\lambda_m/\mu_m)$) and rearranging gives

$$V(P, \pi, \mu_m) \le \frac{1 + \gamma_{N_m,\mu_m}}{1 - \gamma_{N_m,\mu_m}} \cdot \frac{\widehat{V}_m(P, \pi, \mu_m)}{1 - (e-2)\frac{\lambda_m}{\mu_m}} + \frac{1}{1 - \gamma_{N_m,\mu_m}} \cdot \frac{1}{1 - (e-2)\frac{\lambda_m}{\mu_m}} \cdot \frac{\ln(|\Pi|^{N_m+1}2m^2/\delta)}{\lambda_m \tau_m}.$$

Using $N_m := \lceil 12(1 - K\mu_m)/\mu_m \rceil$ and $\lambda_m := 0.66\mu_m$ for all $m \in \mathbb{N}$ gives the claimed inequalities.

If $\mu_m \ge \sqrt{\ln(2|\Pi|m^2/\delta)/(K\tau_m)}$ and $\tau_m \ge 4K\ln(2|\Pi|m^2/\delta)$, then $\mu_m^2\tau_m \ge \ln(|\Pi|)/K$ and $\mu_m\tau_m \ge \ln(2|\Pi|^2m^2/\delta)$, and hence

$$\frac{75(1 - K\mu_m)\ln|\Pi|}{\mu_m^2\tau_m} + \frac{6.3\ln(2|\Pi|^2m^2/\delta)}{\mu_m\tau_m} \le (75 + 6.3)K = 81.3K. \qquad \square$$

B.3 Reward Estimates

Again, fix the epoch schedule $0 = \tau_0 < \tau_1 < \tau_2 < \cdots$. Recall that for any epoch $m \in \mathbb{N}$ and round t in epoch m,

- $Q_{m-1} \in \Delta^{\Pi}$ are the non-negative weights computed at the end of epoch $m-1$;
- $\widetilde{Q}_{m-1}$ is the probability distribution over Π obtained from Q_{m-1} and the policy π_{m-1} with the highest reward estimate through epoch $m-1$;
- $\widetilde{Q}_{m-1}^{\mu_{m-1}}(\cdot|x_t)$ is the probability distribution used to choose a_t.

Let

$$m(t) := \min\{m \in \mathbb{N} : t \leq \tau_m\} \tag{10}$$

be the index of the epoch containing round $t \in \mathbb{N}$, and define

$$\mathcal{V}_t(\pi) := \max_{0 \leq m \leq m(t)-1} \{V(\widetilde{Q}_m, \pi, \mu_m)\} \tag{11}$$

for all $t \in \mathbb{N}$ and $\pi \in \Pi$.

Lemma 11. *For any $\delta \in (0,1)$ and any choices of $\lambda_{m-1} \in [0, \mu_{m-1}]$ for $m \in \mathbb{N}$, with probability at least $1-\delta$,*

$$|\widehat{\mathcal{R}}_t(\pi) - \mathcal{R}(\pi)| \leq \mathcal{V}_t(\pi)\lambda_{m-1} + \frac{\ln(4t^2|\Pi|/\delta)}{t\lambda_{m-1}}$$

for all policies $\pi \in \Pi$, all epochs $m \in \mathbb{N}$, and all rounds t in epoch m.

Proof. Fix any policy $\pi \in \Pi$, epoch $m \in \mathbb{N}$, and round t in epoch m. Then

$$\widehat{\mathcal{R}}_t(\pi) - \mathcal{R}(\pi) = \frac{1}{t}\sum_{i=1}^{t} Z_i$$

where $Z_i := \hat{r}_i(\pi(x_i)) - r_i(\pi(x_i))$. Round i is in epoch $m(i) \leq m$, so

$$|Z_i| \leq \frac{1}{\widetilde{Q}_{m(i)-1}^{\mu_{m(i)-1}}(\pi(x_i)|x_i)} \leq \frac{1}{\mu_{m(i)-1}}$$

by the definition of the fictitious rewards. Because the sequences $\mu_1 \geq \mu_2 \geq \cdots$ and $m(1) \leq m(2) \leq \cdots$ are monotone, it follows that $Z_i \leq 1/\mu_{m-1}$ for all $1 \leq i \leq t$. Furthermore, $\mathbb{E}[Z_i|H_{i-1}] = 0$ and

$$\begin{aligned}\mathbb{E}[Z_i^2|H_{i-1}] &\leq \mathbb{E}[\hat{r}_i(\pi(x_i))^2|H_{i-1}] \\ &\leq V(\widetilde{Q}_{m(i)-1}, \pi, \mu_{m(i)-1}) \leq \mathcal{V}_t(\pi)\end{aligned}$$

for all $1 \leq i \leq t$. The first inequality follows because for $\mathrm{var}(X) \leq \mathbb{E}(X^2)$ for any random variable X; and the other inequalities follow from the definitions of the fictitious rewards, $V(\cdot,\cdot,\cdot)$ in Eq. (8), and $\mathcal{V}_t(\cdot)$ in Eq. (11). Applying Freedman's inequality and a union bound to the sums $(1/t)\sum_{i=1}^{t} Z_i$ and $(1/t)\sum_{i=1}^{t}(-Z_i)$ implies the following: for all $\lambda_{m-1} \in [0, \mu_{m-1}]$, with probability at least $1 - 2\cdot\delta/(4t^2|\Pi|)$,

$$\left|\frac{1}{t}\sum_{i=1}^{t} Z_i\right| \leq (e-2)\mathcal{V}_t(\pi)\lambda_{m-1} + \frac{\ln(4t^2|\Pi|/\delta)}{t\lambda_{m-1}}.$$

The lemma now follows by applying a union bound for all choices of $\pi \in \Pi$ and $t \in \mathbb{N}$, since

$$\sum_{\pi\in\Pi}\sum_{t\in\mathbb{N}} \frac{\delta}{2t^2|\Pi|} \leq \delta.$$

□

C Regret Analysis

Throughout this section, we fix the allowed probability of failure $\delta \in (0,1)$ provided as input to the algorithm, as well as the epoch schedule $0 = \tau_0 < \tau_1 < \tau_2 < \cdots$.

C.1 Definitions

Define, for all $t \in \mathbb{N}$,

$$d_t := \ln(16t^2|\Pi|/\delta), \tag{12}$$

and recall that,

$$\mu_m = \min\left\{\frac{1}{2K}, \sqrt{\frac{d_{\tau_m}}{K\tau_m}}\right\}.$$

Observe that d_t/t is non-increasing with $t \in \mathbb{N}$, and μ_m is non-increasing with $m \in \mathbb{N}$.

Let

$$m_0 := \min\left\{m \in \mathbb{N} : \frac{d_{\tau_m}}{\tau_m} \le \frac{1}{4K}\right\}.$$

Observe that $\tau_{m_0} \ge 2$.

Define

$$\rho := \sup_{m \ge m_0}\left\{\sqrt{\frac{\tau_m}{\tau_{m-1}}}\right\}.$$

Recall that we assume $\tau_{m+1} \le 2\tau_m$; thus $\rho \le \sqrt{2}$.

C.2 Deviation Control and Optimization Constraints

Let $\mathcal{E}$ be the event in which the following statements hold:

$$V(P,\pi,\mu_m) \le 6.4\widehat{V}_m(P,\pi,\mu_m) + 81.3K \tag{13}$$

for all probability distributions P over Π, all $\pi \in \Pi$, and all $m \in \mathbb{N}$ such that $\mu_m \ge \sqrt{d_{\tau_m}/(K\tau_m)}$ and $\tau_m \ge 4Kd_{\tau_m}$; and

$$|\widehat{\mathcal{R}}_t(\pi) - \mathcal{R}(\pi)| \le \mathcal{V}_t(\pi)\lambda_t + \frac{d_t}{t\lambda_t} \tag{14}$$

where

$$\lambda_t := \begin{cases} \sqrt{\frac{d_t}{2Kt}} & \text{if } m \le m_0, \\ \mu_{m-1} & \text{if } m > m_0. \end{cases}$$

for all all policies $\pi \in \Pi$, all epochs $m \in \mathbb{N}$, and all rounds t in epoch m. By Lemma 10, Lemma 11, and a union bound, $\Pr(\mathcal{E}) \ge 1 - \delta/2$.

For every epoch $m \in \mathbb{N}$, the weights Q_m computed at the end of the epoch (in round τ_m) as the solution to (OP) satisfy the constraints Eq. (2) and Eq. (3): they are, respectively:

$$\sum_{\pi\in\Pi} Q_m(\pi)\widehat{\mathrm{Reg}}_{\tau_m}(\pi) \le \psi \cdot 2K\mu_m \tag{15}$$

and, for all $\pi \in \Pi$,

$$\widehat{V}_m(Q_m,\pi,\mu_m) \le 2K + \frac{\widehat{\mathrm{Reg}}_{\tau_m}(\pi)}{\psi\cdot\mu_m}. \tag{16}$$

Recall that $\psi = 100$ (as defined in (OP), assuming $\rho \le \sqrt{2}$). Define $\theta_1 := 94.1$ and $\theta_2 := \psi/6.4$ (which come from Lemma 12); the proof of Lemma 13 requires that $\theta_2 \ge 8\rho$, and hence $\psi \ge 6.4 \cdot 8\rho$; this is true with our setting of ψ since $\rho \le \sqrt{2}$.

C.3 Proof of Theorem 2

We now give the proof of Theorem 2, following the outline in Section 4.

The following lemma shows that if $\mathcal{V}_t(\pi)$ is large—specifically, much larger than K—then the estimated regret of π was large in some previous round.

Lemma 12. *Assume event $\mathcal{E}$ holds. Pick any round $t \in \mathbb{N}$ and any policy $\pi \in \Pi$, and let $m \in \mathbb{N}$ be the epoch achieving the* max *in the definition of $\mathcal{V}_t(\pi)$. Then*

$$\mathcal{V}_t(\pi) \leq \begin{cases} 2K & \text{if } \mu_m = 1/(2K), \\ \theta_1 K + \dfrac{\widehat{\mathrm{Reg}}_{\tau_m}(\pi)}{\theta_2 \mu_m} & \text{if } \mu_m < 1/(2K). \end{cases}$$

Proof. Fix a round $t \in \mathbb{N}$ and policy $\pi \in \Pi$. Let $m \leq m(t) - 1$ be the epoch achieving the max in the definition of $\mathcal{V}_t(\pi)$ from Eq. (11), so $\mathcal{V}_t(\pi) = V(\widetilde{Q}_m, \pi, \mu_m)$. If $\mu_m = 1/(2K)$, then $V(\widetilde{Q}_m, \pi, \mu_m) \leq 2K$. So assume instead that $1/(2K) > \mu_m = \sqrt{d_{\tau_m}/(K\tau_m)}$. This implies that $\tau_m > 4Kd_{\tau_m}$. By Eq. (13), which holds in event $\mathcal{E}$,

$$V(\widetilde{Q}_m, \pi, \mu_m) \leq 6.4\widehat{V}_m(\widetilde{Q}_m, \pi, \mu_m) + 81.3K.$$

The probability distribution $\widetilde{Q}_m$ satisfies the inequalities

$$\widehat{V}_m(\widetilde{Q}_m, \pi, \mu_m) \leq \widehat{V}_m(Q_m, \pi, \mu_m) \leq 2K + \frac{\widehat{\mathrm{Reg}}_{\tau_m}(\pi)}{\psi\mu_m}.$$

Above, the first inequality follows because the value of $\widehat{V}_m(Q_m, \pi, \mu_m)$ decreases as the value of $Q_m(\pi_{\tau_m})$ increases, as it does when going from Q_m to $\widetilde{Q}_m$; the second inequality is the constraint Eq. (16) satisfied by Q_m. Combining the displayed inequalities from above proves the claim. □

In the next lemma, we compare $\mathrm{Reg}(\pi)$ and $\widehat{\mathrm{Reg}}_t(\pi)$ for any policy π by using the deviation bounds for estimated rewards together with the variance bounds from Lemma 12. Define $t_0 := \min\{t \in \mathbb{N} : d_t/t \leq 1/(4K)\}$.

Lemma 13. *Assume event $\mathcal{E}$ holds. Let $c_0 := 4\rho(1 + \theta_1)$. For all epochs $m \geq m_0$, all rounds $t \geq t_0$ in epoch m, and all policies $\pi \in \Pi$,*

$$\mathrm{Reg}(\pi) \leq 2\widehat{\mathrm{Reg}}_t(\pi) + c_0 K\mu_m;$$
$$\widehat{\mathrm{Reg}}_t(\pi) \leq 2\,\mathrm{Reg}(\pi) + c_0 K\mu_m.$$

Proof. The proof is by induction on m. As the base case, consider $m = m_0$ and $t \geq t_0$ in epoch m. By definition of m_0, $\mu_m = 1/(2K)$ for all $m < m_0$, so $\mathcal{V}_t(\pi) \leq 2K$ for all $\pi \in \Pi$ by Lemma 12. By Eq. (14), which holds in event $\mathcal{E}$, for all $\pi \in \Pi$,

$$|\widehat{\mathcal{R}}_t(\pi) - \mathcal{R}(\pi)| \leq 2K\lambda + \frac{d_t}{t\lambda}$$

for all $\pi \in \Pi$, where $\lambda = \sqrt{d_t/(2Kt)}$. This implies

$$|\widehat{\mathcal{R}}_t(\pi) - \mathcal{R}(\pi)| \leq 2\sqrt{\frac{2Kd_t}{t}}$$

and therefore $|\widehat{\mathrm{Reg}}_t(\pi) - \mathrm{Reg}(\pi)| \leq 4\sqrt{2Kd_t/t}$ by the triangle inequality and optimality of π_t and $\pi_\star$. Since $t > \tau_{m_0-1}$ and $c_0 \geq 4\sqrt{2}\rho$, it follows that $|\widehat{\mathrm{Reg}}_t(\pi) - \mathrm{Reg}(\pi)| \leq 4\sqrt{2}\rho K\mu_{m_0} \leq c_0 K\mu_{m_0}$.

For the inductive step, fix some epoch $m > m_0$. We assume as the inductive hypothesis that for all epochs $m' < m$, all rounds t' in epoch m', and all $\pi \in \Pi$,

$$\mathrm{Reg}(\pi) \leq 2\widehat{\mathrm{Reg}}_{t'}(\pi) + c_0 K\mu_{m'};$$
$$\widehat{\mathrm{Reg}}_{t'}(\pi) \leq 2\,\mathrm{Reg}(\pi) + c_0 K\mu_{m'}.$$

We first show that

$$\mathrm{Reg}(\pi) \le 2\widehat{\mathrm{Reg}}_t(\pi) + c_0 K \mu_m \tag{17}$$

for all rounds t in epoch m and all $\pi \in \Pi$. So fix such a round t and policy π; by Eq. (14) (which holds in event $\mathcal{E}$),

$$\begin{aligned}
\mathrm{Reg}(\pi) - \widehat{\mathrm{Reg}}_t(\pi) &= \big(\mathcal{R}(\pi_\star) - \mathcal{R}(\pi)\big) - \big(\widehat{\mathcal{R}}_t(\pi_t) - \widehat{\mathcal{R}}_t(\pi)\big) \\
&\le \big(\mathcal{R}(\pi_\star) - \mathcal{R}(\pi)\big) - \big(\widehat{\mathcal{R}}_t(\pi_\star) - \widehat{\mathcal{R}}_t(\pi)\big) \\
&\le \big(\mathcal{V}_t(\pi) + \mathcal{V}_t(\pi_\star)\big)\mu_{m-1} + \frac{2d_t}{t\mu_{m-1}}.
\end{aligned} \tag{18}$$

Above, the first inequality follows from the optimality of π_t. By Lemma 12, there exist epochs $i, j < m$ such that

$$\begin{aligned}
\mathcal{V}_t(\pi) &\le \theta_1 K + \frac{\widehat{\mathrm{Reg}}_{\tau_i}(\pi)}{\theta_2 \mu_i} \cdot \mathbb{1}\{\mu_i < 1/(2K)\}, \\
\mathcal{V}_t(\pi_\star) &\le \theta_1 K + \frac{\widehat{\mathrm{Reg}}_{\tau_j}(\pi_\star)}{\theta_2 \mu_j} \cdot \mathbb{1}\{\mu_j < 1/(2K)\}.
\end{aligned}$$

Suppose $\mu_i < 1/(2K)$, so $m_0 \le i < m$: in this case, the inductive hypothesis implies

$$\frac{\widehat{\mathrm{Reg}}_{\tau_i}(\pi)}{\theta_2 \mu_i} \le \frac{2\,\mathrm{Reg}(\pi) + c_0 K \mu_i}{\theta_2 \mu_i} \le \frac{c_0 K}{\theta_2} + \frac{2\,\mathrm{Reg}(\pi)}{\theta_2 \mu_{m-1}}$$

where the second inequality uses the fact that $i \le m-1$. Therefore,

$$\mathcal{V}_t(\pi)\mu_{m-1} \le \left(\theta_1 + \frac{c_0}{\theta_2}\right) K \mu_{m-1} + \frac{2}{\theta_2}\,\mathrm{Reg}(\pi). \tag{19}$$

Now suppose $\mu_j < 1/(2K)$, so $m_0 \le j < m$: as above, the inductive hypothesis implies

$$\frac{\widehat{\mathrm{Reg}}_{\tau_j}(\pi_\star)}{\theta_2 \mu_j} \le \frac{2\,\mathrm{Reg}(\pi_\star) + c_0 K \mu_j}{\theta_2 \mu_j} = \frac{c_0}{\theta_2} K$$

since $\mathrm{Reg}(\pi_\star) = 0$. Therefore,

$$\mathcal{V}_t(\pi_\star)\mu_{m-1} \le \left(\theta_1 + \frac{c_0}{\theta_2}\right) K \mu_{m-1}. \tag{20}$$

Combining Eq. (18), Eq. (19), and Eq. (20), and rearranging gives

$$\mathrm{Reg}(\pi) \le \frac{1}{1 - \frac{2}{\theta_2}} \left(\widehat{\mathrm{Reg}}_t(\pi) + 2\left(\theta_1 + \frac{c_0}{\theta_2}\right) K \mu_{m-1} + \frac{2d_t}{t\mu_{m-1}}\right).$$

Since $m \ge m_0 + 1$, it follows that $\mu_{m-1} \le \rho \mu_m$ by definition of ρ. Moreover, since $t > \tau_{m-1}$, $(d_t/t)/\mu_{m-1} \le K\mu_{m-1}^2/\mu_{m-1} \le \rho K \mu_m$ Applying these inequalities to the above display, and simplifying, yields Eq. (17) because $c_0 \ge 4\rho(1+\theta_1)$ and $\theta_2 \ge 8\rho$.

We now show that

$$\widehat{\mathrm{Reg}}_t(\pi) \le 2\,\mathrm{Reg}(\pi) + c_0 K \mu_m \tag{21}$$

for all $\pi \in \Pi$. Again, fix an arbitrary $\pi \in \Pi$, and by Eq. (14),

$$\begin{aligned}
\widehat{\mathrm{Reg}}_t(\pi) - \mathrm{Reg}(\pi) &= \big(\widehat{\mathcal{R}}_t(\pi_t) - \widehat{\mathcal{R}}_t(\pi)\big) - \big(\mathcal{R}(\pi_\star) - \mathcal{R}(\pi)\big) \\
&\le \big(\widehat{\mathcal{R}}_t(\pi_t) - \widehat{\mathcal{R}}_t(\pi)\big) - \big(\mathcal{R}(\pi_t) - \mathcal{R}(\pi)\big) \\
&\le \big(\mathcal{V}_t(\pi) + \mathcal{V}_t(\pi_t)\big)\mu_{m-1} + \frac{2d_t}{t\mu_{m-1}}
\end{aligned} \tag{22}$$

where the first inequality follows from the optimality of $\pi_\star$. By Lemma 12, there exists an epoch $j < m$ such

$$\mathcal{V}_t(\pi_t) \le \theta_1 K + \frac{\widehat{\mathrm{Reg}}_{\tau_j}(\pi_t)}{\theta_2 \mu_j} \cdot \mathbb{1}\{\mu_j < 1/(2K)\}.$$

Suppose $\mu_j < 1/(2K)$, so $m_0 \le j < m$: in this case the inductive hypothesis and Eq. (17) imply

$$\frac{\widehat{\mathrm{Reg}}_{\tau_j}(\pi_t)}{\theta_2 \mu_j} \le \frac{2\,\mathrm{Reg}(\pi_t) + c_0 K \mu_j}{\theta_2 \mu_j} \le \frac{2\Big(2\widehat{\mathrm{Reg}}_t(\pi_t) + c_0 K \mu_m\Big) + c_0 K \mu_j}{\theta_2 \mu_j} = \frac{3c_0}{\theta_2} K$$

(the last equality follows because $\widehat{\mathrm{Reg}}_t(\pi_t) = 0$). Thus

$$\mathcal{V}_t(\pi_t)\mu_{\tau(t)-1} \le \left(\theta_1 + \frac{3c_0}{\theta_2}\right) K \mu_{m-1}. \tag{23}$$

Combining Eq. (22), Eq. (23), and Eq. (19) gives

$$\widehat{\mathrm{Reg}}_t(\pi) \le \left(1 + \frac{2}{\theta_2}\right) \mathrm{Reg}(\pi) + \left(2\theta_1 + \frac{4c_0}{\theta_2}\right) K \mu_{m-1} + \frac{2d_t}{t\mu_{m-1}}.$$

Again, applying the inequalities $\mu_{m-1} \le \rho\mu_m$ and $(d_t/t)/\mu_{m-1} \le K\mu_m$ to the above display, and simplifying, yields Eq. (21) because $c_0 \ge 4\rho(1+\theta_1)$ and $\theta_2 \ge 8\rho$. This completes the inductive step, and thus proves the overall claim. □

The next lemma shows that the "low estimated regret guarantee" of Q_{t-1} (optimization constraint Eq. (15)) also implies a "low regret guarantee", via the comparison of $\widehat{\mathrm{Reg}}_t(\cdot)$ to $\mathrm{Reg}(\cdot)$ from Lemma 13.

Lemma 14. *Assume event $\mathcal{E}$ holds. For every epoch $m \in \mathbb{N}$,*

$$\sum_{\pi \in \Pi} \widetilde{Q}_{m-1}(\pi)\,\mathrm{Reg}(\pi) \le (4\psi + c_0) K \mu_{m-1}$$

where c_0 is defined in Lemma 13.

Proof. Fix any epoch $m \in \mathbb{N}$. If $m \le m_0$, then $\mu_{m-1} = 1/(2K)$, in which case the claim is trivial. Therefore assume $m \ge m_0 + 1$. Then

$$\begin{aligned}
\sum_{\pi \in \Pi} \widetilde{Q}_{m-1}(\pi)\,\mathrm{Reg}(\pi) &\le \sum_{\pi \in \Pi} \widetilde{Q}_{m-1}(\pi)\big(2\widehat{\mathrm{Reg}}_{\tau_{m-1}}(\pi) + c_0 K \mu_{m-1}\big) \\
&= \left(2 \sum_{\pi \in \Pi} Q_{m-1}(\pi) \widehat{\mathrm{Reg}}_{\tau_{m-1}}(\pi)\right) + c_0 K \mu_{m-1} \\
&\le \psi \cdot 4K\mu_{m-1} + c_0 K \mu_{m-1}.
\end{aligned}$$

The first step follows from Lemma 13, as all rounds in an epoch $m \ge m_0 + 1$ satisfy $t \ge t_0$; the second step follows from the fact that $\widetilde{Q}_{m-1}$ is a probability distribution, that $\widetilde{Q}_{m-1} = Q_{m-1} + \alpha \mathbb{1}_{\pi_{\tau_{m-1}}}$ for some $\alpha \ge 0$, and that $\widehat{\mathrm{Reg}}_{\tau_{m-1}}(\pi_{\tau_{m-1}}) = 0$; and the last step follows from the constraint Eq. (15) satisfied by Q_{m-1}. □

Finally, we straightforwardly translate the "low regret guarantee" from Lemma 14 to a bound on the cumulative regret of the algorithm. This involves summing the bound in Lemma 14 over all rounds t (Lemma 15 and Lemma 16) and applying a martingale concentration argument (Lemma 17).

Lemma 15. *For any $T \in \mathbb{N}$,*

$$\sum_{t=1}^{T} \mu_{m(t)} \le 2\sqrt{\frac{d_{\tau_{m(T)}} \tau_{m(T)}}{K}}.$$

Proof. We break the sum over rounds into the epochs, and bound the sum within each epoch:

$$\begin{aligned}
\sum_{t=1}^{T} \mu_{m(t)} &\le \sum_{m=1}^{m(T)} \sum_{t=\tau_{m-1}+1}^{\tau_m} \mu_m \\
&\le \sum_{m=1}^{m(T)} \sum_{t=\tau_{m-1}+1}^{\tau_m} \sqrt{\frac{d_{\tau_m}}{K\tau_m}} \\
&\le \sqrt{\frac{d_{\tau_{m(T)}}}{K}} \sum_{m=1}^{m(T)} \frac{\tau_m - \tau_{m-1}}{\sqrt{\tau_m}} \\
&\le \sqrt{\frac{d_{\tau_{m(T)}}}{K}} \sum_{m=1}^{m(T)} \int_{\tau_{m-1}}^{\tau_m} \frac{dx}{\sqrt{x}} = \sqrt{\frac{d_{\tau_{m(T)}}}{K}} \int_{\tau_0}^{\tau_{m(T)}} \frac{dx}{\sqrt{x}} = 2\sqrt{\frac{d_{\tau_{m(T)}}}{K}}\sqrt{\tau_{m(T)}}.
\end{aligned}$$

Above, the first step uses the fact that $m(1) = 1$ and $\tau_{m(t)-1} + 1 \le t \le \tau_{m(t)}$. The second step uses the definition of μ_m. The third step simplifies the sum over t and uses the bound $d_{\tau_{m-1}} \le d_{\tau_{m(T)}}$. The remaining steps use an integral bound which is then directly evaluated (recalling that $\tau_0 = 0$). □

Lemma 16. *For any $T \in \mathbb{N}$,*

$$\sum_{t=1}^{T} \mu_{m(t)-1} \le \frac{\tau_{m_0}}{2K} + \sqrt{\frac{8 d_{\tau_{m(T)}} \tau_{m(T)}}{K}}.$$

Proof. Under the epoch schedule condition $\tau_{m+1} \le 2\tau_m$, we have $\mu_{m(t)-1} \le \sqrt{2}\mu_{m(t)}$ whenever $m(t) > m_0$; also, $\mu_{m(t)-1} \le 1/(2K)$ whenever $m(t) \le m_0$. The conclusion follows by applying Lemma 15. □

Lemma 17. *For any $T \in \mathbb{N}$, with probability at least $1 - \delta$, the regret after T rounds is at most*

$$C_0\left(4K d_{\tau_{m_0-1}} + \sqrt{8K d_{\tau_{m(T)}} \tau_{m(T)}}\right) + \sqrt{8\log(2/\delta)}$$

where $C_0 := (4\psi + c_0)$ and c_0 is defined in Lemma 13.

Proof. Fix $T \in \mathbb{N}$. For each round $t \in \mathbb{N}$, let $Z_t := r_t(\pi_\star(x_t)) - r_t(a_t) - \sum_{\pi\in\Pi} \widetilde{Q}_{m(t)-1} \operatorname{Reg}(\pi)$. Since

$$\mathbb{E}[r_t(\pi_\star(x_t)) - r_t(a_t) | H_{t-1}] = \mathcal{R}(\pi_\star) - \sum_{\pi\in\Pi} \widetilde{Q}_{m(t)-1}(\pi)\mathcal{R}(\pi) = \sum_{\pi\in\Pi} \widetilde{Q}_{m(t)-1} \operatorname{Reg}(\pi),$$

it follows that $\mathbb{E}[Z_t | H_{t-1}] = 0$. Since $|Z_t| \le 2$, it follows by Azuma's inequality that

$$\sum_{t=1}^{T} Z_t \le 2\sqrt{2T\ln(2/\delta)}$$

with probability at least $1 - \delta/2$. By Lemma 10, Lemma 11, and a union bound, the event $\mathcal{E}$ holds with probability at least $1 - \delta/2$. Hence, by another union bound, with probability at least $1 - \delta$, event $\mathcal{E}$ holds and the regret of the algorithm is bounded by

$$\sum_{t=1}^{T} \sum_{\pi\in\Pi} \widetilde{Q}_{m(t)-1}(\pi) \operatorname{Reg}(\pi) + 2\sqrt{2\ln(2/\delta)}.$$

The double summation above is bounded by Lemma 14 and Lemma 16:

$$\sum_{t=1}^{T} \sum_{\pi\in\Pi} \widetilde{Q}_{m(t)-1}(\pi) \operatorname{Reg}(\pi) \le (4\psi + c_0)K \sum_{t=1}^{T} \mu_{m(t)-1} \le (4\psi + c_0)\left(\frac{\tau_{m_0}}{2} + \sqrt{8K d_{\tau_{m(T)}} \tau_{m(T)}}\right).$$

By the definition of m_0, $\tau_{m_0-1} \le 4K d_{\tau_{m_0-1}}$. Since $\tau_{m_0} \le 2\tau_{m_0-1}$ by assumption, it follows that $\tau_{m_0} \le 8K d_{\tau_{m_0-1}}$. □

Theorem 2 follows from Lemma 17 and the fact that $\tau_{m(T)} \le 2(T-1)$ whenever $\tau_{m(T)-1} \ge 1$.

There is one last result implied by Lemma 12 and Lemma 13 that is used elsewhere.

Lemma 18. *Assume event $\mathcal{E}$ holds, and t is such that $d_{\tau_{m(t)-1}}/\tau_{m(t)-1} \le 1/(4K)$. Then*

$$\widehat{\mathcal{R}}_t(\pi_t) \le \mathcal{R}(\pi_\star) + \left(\theta_1 + \frac{c_0}{\theta_2} + c_0 + 1\right) K\mu_{m(t)-1}.$$

Proof. Let $m' < m(t)$ achieve the max in the definition of $\mathcal{V}_t(\pi_\star)$. If $\mu_{m'} < 1/(2K)$, then $m' \ge m_0$, and

$$\begin{aligned}\mathcal{V}_t(\pi_\star) &\le \theta_1 K + \frac{\widehat{\mathrm{Reg}}_{\tau_{m'}}(\pi_\star)}{\theta_2 \mu_{m'}} \\ &\le \theta_1 K + \frac{2\,\mathrm{Reg}(\pi_\star) + c_0 K\mu_{m'}}{\theta_2\mu_{m'}} = cK\end{aligned}$$

for $c := \theta_1 + c_0/\theta_2$. Above, the second inequality follows by Lemma 13. If $\mu_{m'} = 1/(2K)$, then the same bound also holds. Using this bound, we obtain from Eq. (14),

$$\widehat{\mathcal{R}}_t(\pi_\star) - \mathcal{R}(\pi_\star) \le cK\mu_{m(t)-1} + \frac{d_t}{t\mu_{m(t)-1}}.$$

To conclude,

$$\begin{aligned}\widehat{\mathcal{R}}_t(\pi_{\tau_m}) &= \mathcal{R}(\pi_\star) + \left(\widehat{\mathcal{R}}_t(\pi_\star) - \mathcal{R}(\pi_\star)\right) + \widehat{\mathrm{Reg}}_t(\pi_\star) \\ &\le \mathcal{R}(\pi_\star) + cK\mu_{m(t)-1} + \frac{d_t}{t\mu_{m(t)-1}} + \widehat{\mathrm{Reg}}_t(\pi_\star) \\ &\le \mathcal{R}(\pi_\star) + cK\mu_{m(t)-1} + \frac{d_t}{t\mu_{m(t)-1}} + c_0 K\mu_{m(t)}\end{aligned}$$

where the last inequality follows from Lemma 13. The claim follows because $d_t/t \le d_{\tau_{m(t)-1}}/\tau_{m(t)-1}$ and $\mu_{m(t)} \le \mu_{m(t)-1}$. □

D Details of Optimization Analysis

D.1 Proof of Lemma 5

Following the execution of step 4, we must have

$$\sum_\pi Q(\pi)(2K + b_\pi) \le 2K. \tag{24}$$

This is because, if the condition in step 7 does not hold, then Eq. (24) is already true. Otherwise, Q is replaced by $Q' = cQ$, and for this set of weights, Eq. (24) in fact holds with equality. Note that, since all quantities are nonnegative, Eq. (24) immediately implies both Eq. (2), and that $\sum_\pi Q(\pi) \le 1$.

Furthermore, at the point where the algorithm halts at step 10, it must be that for all policies π, $D_\pi(Q) \le 0$. However, unraveling definitions, we can see that this is exactly equivalent to Eq. (3). □

D.2 Proof of Lemma 6

Consider the function

$$g(c) = B_0\Phi_m(cQ),$$

where, in this proof, $B_0 = 2K/(\tau\mu)$. Let $Q_c^\mu(a|x) = (1-K\mu)cQ(a|x) + \mu$. By the chain rule, the first derivative of g is:

$$\begin{aligned} g'(c) &= B_0 \sum_\pi Q(\pi) \frac{\partial G(cQ)}{\partial Q(\pi)} \\ &= \sum_\pi Q(\pi) \left((2K + b_\pi) - 2\widehat{\mathbb{E}}_{x\sim H_t}\left[\frac{1}{Q_c^\mu(\pi(x)|x)}\right]\right) \end{aligned} \tag{25}$$

To handle the second term, note that

$$\begin{aligned} \sum_\pi Q(\pi)\widehat{\mathbb{E}}_{x\sim H_t}\left[\frac{1}{Q_c^\mu(\pi(x)|x)}\right] &= \sum_\pi Q(\pi)\widehat{\mathbb{E}}_{x\sim H_t}\left[\sum_{a\in A}\frac{\mathbb{1}\{\pi(x)=a\}}{Q_c^\mu(a|x)}\right] \\ &= \widehat{\mathbb{E}}_{x\sim H_t}\left[\sum_{a\in A}\sum_\pi \frac{Q(\pi)\mathbb{1}\{\pi(x)=a\}}{Q_c^\mu(a|x)}\right] \\ &= \widehat{\mathbb{E}}_{x\sim H_t}\left[\sum_{a\in A}\frac{Q(a|x)}{Q_c^\mu(a|x)}\right] \\ &= \frac{1}{c}\widehat{\mathbb{E}}_{x\sim H_t}\left[\sum_{a\in A}\frac{cQ(a|x)}{(1-K\mu)cQ(a|x)+\mu}\right] \le \frac{K}{c}. \end{aligned} \tag{26}$$

To see the inequality in Eq. (26), let us fix x and define $q_a = cQ(a|x)$. Then $\sum_a q_a = c\sum_\pi Q(\pi) \le 1$ by Eq. (4). Further, the expression inside the expectation in Eq. (26) is equal to

$$\begin{aligned} \sum_a \frac{q_a}{(1-K\mu)q_a + \mu} &= K \cdot \frac{1}{K}\sum_a \frac{1}{(1-K\mu) + \mu/q_a} \\ &\le K \cdot \frac{1}{(1-K\mu) + K\mu/\sum_a q_a} \qquad (27) \\ &\le K \cdot \frac{1}{(1-K\mu)+K\mu} = K. \qquad (28) \end{aligned}$$

Eq. (27) uses Jensen's inequality, combined with the fact that the function $1/(1-K\mu+\mu/x)$ is concave (as a function of x). Eq. (28) uses the fact that the function $1/(1-K\mu+K\mu/x)$ is nondecreasing (in x), and that the q_a's sum to at most 1.

Thus, plugging Eq. (26) into Eq. (25) yields

$$g'(c) \ge \sum_\pi Q(\pi)(2K + b_\pi) - \frac{2K}{c} = 0$$

by our definition of c. Since g is convex, this means that g is nondecreasing for all values exceeding c. In particular, since $c < 1$, this gives

$$B_0\Phi_m(Q) = g(1) \ge g(c) = B_0\Phi_m(cQ),$$

implying the lemma since $B_0 > 0$. □

D.3 Proof of Lemma 7

We first compute the change in potential for general α. Note that $Q'^\mu(a|x) = Q^\mu(a|x)$ if $a \neq \pi(x)$, and otherwise

$$Q'^\mu(\pi(x)|x) = Q^\mu(\pi(x)|x) + (1-K\mu)\alpha.$$

Thus, most of the terms defining $\Phi_m(Q)$ are left unchanged by the update. In particular, by a direct calculation:

$$\begin{aligned}
\frac{2K}{\tau\mu}(\Phi_m(Q) - \Phi_m(Q')) &= \frac{2}{1-K\mu}\widehat{\mathbb{E}}_{x\sim H_t}\left[\ln\left(1+\frac{\alpha(1-K\mu)}{Q^\mu(\pi(x)|x)}\right)\right] - \alpha(2K+b_\pi) \\
&\geq \frac{2}{1-K\mu}\widehat{\mathbb{E}}_{x\sim H_t}\left[\frac{\alpha(1-K\mu)}{Q^\mu(\pi(x)|x)} - \frac{1}{2}\left(\frac{\alpha(1-K\mu)}{Q^\mu(\pi(x)|x)}\right)^2\right] \\
&\quad -\alpha(2K+b_\pi) \qquad (29) \\
&= 2\alpha V_\pi(Q) - (1-K\mu)\alpha^2 S_\pi(Q) - \alpha(2K+b_\pi) \\
&= \alpha(V_\pi(Q) + D_\pi(Q)) - (1-K\mu)\alpha^2 S_\pi(Q) \qquad (30) \\
&= \frac{(V_\pi(Q)+D_\pi(Q))^2}{4(1-K\mu)S_\pi(Q)}. \qquad (31)
\end{aligned}$$

Eq. (29) uses the bound $\ln(1+x) \geq x - x^2/2$ which holds for $x \geq 0$ (by Taylor's theorem). Eq. (31) holds by our choice of $\alpha = \alpha_\pi(Q)$, which was chosen to maximize Eq. (30). By assumption, $D_\pi(Q) > 0$, which implies $V_\pi(Q) > 2K$. Further, since $Q^\mu(a|x) \geq \mu$ always, we have

$$\begin{aligned}
S_\pi(Q) &= \widehat{\mathbb{E}}_{x\sim H_t}\left[\frac{1}{Q^\mu(\pi(x) \mid x)^2}\right] \\
&\leq \frac{1}{\mu}\cdot\widehat{\mathbb{E}}_{x\sim H_t}\left[\frac{1}{Q^\mu(\pi(x) \mid x)}\right] = \frac{V_\pi(Q)}{\mu}.
\end{aligned}$$

Thus,

$$\frac{(V_\pi(Q)+D_\pi(Q))^2}{S_\pi(Q)} \geq \frac{V_\pi(Q)^2}{S_\pi(Q)} = V_\pi(Q)\cdot\frac{V_\pi(Q)}{S_\pi(Q)} \geq 2K\mu.$$

Plugging into Eq. (31) completes the lemma. □

D.4 Proof of Lemma 8

We break the potential of Eq. (6) into pieces and bound the total change in each separately. Specifically, by straightforward algebra, we can write

$$\Phi_m(Q) = \phi_m^a(Q) + \phi_m^b + \phi_m^c(Q) + \phi_m^d(Q)$$

where

$$\begin{aligned}
\phi_m^a(Q) &= \frac{\tau_m\mu_m}{K(1-K\mu_m)}\widehat{\mathbb{E}}_{x\sim H_t}\left[-\sum_a \ln Q^\mu(a|x)\right] \\
\phi_m^b &= \frac{\tau_m\mu_m\ \ln K}{1-K\mu_m} \\
\phi_m^c(Q) &= \tau_m\mu_m\left(\sum_\pi Q(\pi) - 1\right) \\
\phi_m^d(Q) &= \frac{\tau_m\mu_m}{2K}\sum_\pi Q(\pi)b_\pi.
\end{aligned}$$

We assume throughout that $\sum_\pi Q(\pi) \leq 1$ as will always be the case for the vectors produced by Algorithm 2. For such a vector Q,

$$\phi_{m+1}^c(Q) - \phi_m^c(Q) = (\tau_{m+1}\mu_{m+1} - \tau_m\mu_m)\left(\sum_\pi Q(\pi) - 1\right) \leq 0$$

since $\tau_m \mu_m$ is nondecreasing. This means we can essentially disregard the change in this term.

Also, note that ϕ_m^b does not depend on Q. Therefore, for this term, we get a telescoping sum:

$$\sum_{m=1}^{M} (\phi_{m+1}^b - \phi_m^b) = \phi_{M+1}^b - \phi_1^b \le \phi_{M+1}^b \le 2\sqrt{\frac{Td_T}{K}} \ln K$$

since $K\mu_{M+1} \le 1/2$, and where d_T, used in the definition of μ_m, is defined in Eq. (12).

Next, we tackle ϕ_m^a:

Lemma 19.

$$\sum_{m=1}^{M} (\phi_{m+1}^a(Q_m) - \phi_m^a(Q_m)) \le 6\sqrt{\frac{Td_T}{K}} \ln(1/\mu_{M+1}).$$

Proof. For the purposes of this proof, let

$$C_m = \frac{\mu_m}{1 - K\mu_m}.$$

Then we can write

$$\phi_m^a(Q) = -\frac{C_m}{K} \sum_{t=1}^{\tau_m} \sum_a \ln Q^{\mu_m}(a|x_t).$$

Note that $C_m \ge C_{m+1}$ since $\mu_m \ge \mu_{m+1}$. Thus,

$$\begin{aligned}
\phi_{m+1}^a(Q) - \phi_m^a(Q) &\le \frac{C_{m+1}}{K}\Bigg[\sum_{t=1}^{\tau_m} \sum_a \ln Q^{\mu_m}(a|x_t) \\
&\quad - \sum_{t=1}^{\tau_{m+1}} \sum_a \ln Q^{\mu_{m+1}}(a|x_t)\Bigg] \\
&= \frac{C_{m+1}}{K}\Bigg[\sum_{t=1}^{\tau_m} \sum_a \ln\left(\frac{Q^{\mu_m}(a|x_t)}{Q^{\mu_{m+1}}(a|x_t)}\right) \\
&\quad - \sum_{t=\tau_m+1}^{\tau_{m+1}} \sum_a \ln Q^{\mu_{m+1}}(a|x_t).\Bigg] \\
&\le C_{m+1}[\tau_m \ln(\mu_m/\mu_{m+1}) - (\tau_{m+1} - \tau_m) \ln \mu_{m+1}]. \qquad (32)
\end{aligned}$$

Eq. (32) uses $Q^{\mu_{m+1}}(a|x) \ge \mu_{m+1}$, and also

$$\frac{Q^{\mu_m}(a|x)}{Q^{\mu_{m+1}}(a|x)} = \frac{(1 - K\mu_m)Q(a|x) + \mu_m}{(1 - K\mu_{m+1})Q(a|x) + \mu_{m+1}} \le \frac{\mu_m}{\mu_{m+1}}.$$

A sum over the two terms appearing in Eq. (32) can now be bounded separately. Starting with the one on the left, since $\tau_m < \tau_{m+1} \le T$ and $K\mu_m \le 1/2$, we have

$$C_{m+1}\tau_m \le 2\tau_m \mu_{m+1} \le 2\tau_{m+1}\mu_{m+1} \le 2\sqrt{\frac{Td_T}{K}}.$$

Thus,

$$\begin{aligned}
\sum_{m=1}^{M} C_{m+1}\tau_m \ln(\mu_m/\mu_{m+1}) &\le 2\sqrt{\frac{Td_T}{K}} \sum_{m=1}^{M} \ln(\mu_m/\mu_{m+1}) \\
&= 2\sqrt{\frac{Td_T}{K}} \ln(\mu_1/\mu_{M+1})) \\
&\le 2\sqrt{\frac{Td_T}{K}}(-\ln(\mu_{M+1})). \qquad (33)
\end{aligned}$$

For the second term in Eq. (32), using $\mu_{m+1} \geq \mu_{M+1}$ for $m \leq M$, and definition of C_m, we have

$$\begin{aligned}
\sum_{m=1}^{M} -C_{m+1}(\tau_{m+1} - \tau_m) \ln \mu_{m+1} &\leq -2(\ln \mu_{M+1}) \sum_{m=1}^{M} (\tau_{m+1} - \tau_m)\mu_{m+1} \\
&\leq -2(\ln \mu_{M+1}) \sum_{t=1}^{T} \mu_{m(t)} \\
&\leq -4\sqrt{\frac{Td_T}{K}}(\ln \mu_{M+1}) \qquad (34)
\end{aligned}$$

by Lemma 15. Combining Eqs. (32), (33) and (34) gives the statement of the lemma. □

Finally, we come to $\phi_m^d(Q)$, which, by definition of b_π, can be rewritten as

$$\phi_m^d(Q) = B_1 \tau_m \sum_{\pi} Q(\pi) \widehat{\mathrm{Reg}}_{\tau_m}(\pi)$$

where $B_1 = 1/(2K\psi)$ and ψ is the same as appears in optimization problem (OP). Note that, conveniently,

$$\tau_m \widehat{\mathrm{Reg}}_{\tau_m}(\pi) = \widehat{\mathcal{S}}_m(\pi_m) - \widehat{\mathcal{S}}_m(\pi),$$

where $\widehat{\mathcal{S}}_m(\pi)$ is the cumulative empirical importance-weighted reward through round τ_m:

$$\widehat{\mathcal{S}}_m(\pi) = \sum_{t=1}^{\tau_m} \hat{r}_t(\pi(x_t)) = \tau_m \widehat{\mathcal{R}}_{\tau_m}(\pi).$$

From the definition of $\tilde{Q}$, we have that

$$\begin{aligned}
\phi_m^d(\tilde{Q}) &= \phi_m^d(Q) \\
&\quad + B_1 \left(1 - \sum_{\pi} Q(\pi)\right) \tau_m \widehat{\mathrm{Reg}}_{\tau_m}(\pi_m) \\
&= \phi_m^d(Q)
\end{aligned}$$

since $\widehat{\mathrm{Reg}}_{\tau_m}(\pi_m) = 0$. And by a similar computation, $\phi_{m+1}^d(\tilde{Q}) \geq \phi_{m+1}^d(Q)$ since $\widehat{\mathrm{Reg}}_{\tau_{m+1}}(\pi)$ is always nonnegative.

Therefore,

$$\begin{aligned}
\phi_{m+1}^d(Q_m) - \phi_m^d(Q_m) &\leq \phi_{m+1}^d(\tilde{Q}_m) - \phi_m^d(\tilde{Q}_m) \\
&= B_1 \sum_{\pi} \tilde{Q}_m(\pi) \left[\left(\widehat{\mathcal{S}}_{m+1}(\pi_{m+1}) - \widehat{\mathcal{S}}_{m+1}(\pi) \right) - \left(\widehat{\mathcal{S}}_m(\pi_m) - \widehat{\mathcal{S}}_m(\pi) \right) \right] \\
&= B_1 \left(\widehat{\mathcal{S}}_{m+1}(\pi_{m+1}) - \widehat{\mathcal{S}}_m(\pi_m) \right) \\
&\quad - B_1 \left(\sum_{t=\tau_m+1}^{\tau_{m+1}} \sum_{\pi} \tilde{Q}_m(\pi) \hat{r}_t(\pi(x_t)) \right). \qquad (35)
\end{aligned}$$

We separately bound the two parenthesized expressions in Eq. (35) when summed over all epochs. Beginning with the first one, we have

$$\sum_{m=1}^{M} \left(\widehat{\mathcal{S}}_{m+1}(\pi_{m+1}) - \widehat{\mathcal{S}}_m(\pi_m) \right) = \widehat{\mathcal{S}}_{M+1}(\pi_{M+1}) - \widehat{\mathcal{S}}_1(\pi_1) \leq \widehat{\mathcal{S}}_{M+1}(\pi_{M+1}).$$

But by Lemma 18 (and under the same assumptions),

$$\begin{aligned}\widehat{\mathcal{S}}_{M+1}(\pi_{M+1}) &= \tau_{M+1}\widehat{\mathcal{R}}_{\tau_{M+1}}(\pi_{M+1}) \\ &\leq \tau_{M+1}(\mathcal{R}(\pi_\star) + D_0 K\mu_M) \\ &\leq \tau_{M+1}\mathcal{R}(\pi_\star) + D_0\sqrt{KTd_T},\end{aligned}$$

where D_0 is the constant appearing in Lemma 18.

For the second parenthesized expression of Eq. (35), let us define random variables

$$Z_t = \sum_\pi \tilde{Q}_{\tau(t)}(\pi)\hat{r}_t(\pi(x_t)).$$

Note that Z_t is nonnegative, and if $m = \tau(t)$, then

$$\begin{aligned}Z_t &= \sum_\pi \tilde{Q}_m(\pi)\hat{r}_t(\pi(x_t)) \\ &= \sum_a \tilde{Q}_m(a|x_t)\hat{r}_t(a) \\ &= \sum_a \tilde{Q}_m(a|x_t)\frac{r_t(a)\mathbb{1}\{a = a_t\}}{\tilde{Q}^{\mu_m}(a|x_t)} \\ &\leq \frac{r_t(a_t)}{1 - K\mu_m} \leq 2\end{aligned}$$

since $\tilde{Q}^{\mu_m}(a|x) \geq (1 - K\mu_m)\tilde{Q}_m(a|x)$, and since $r_t(a_t) \leq 1$ and $K\mu_m \leq 1/2$. Therefore, by Azuma's inequality, with probability at least $1 - \delta$,

$$\sum_{t=1}^{\tau_{M+1}} Z_t \geq \sum_{t=1}^{\tau_{M+1}} \mathbb{E}[Z_t|H_{t-1}] - \sqrt{2\tau_{M+1}\ln(1/\delta)}.$$

The expectation that appears here can be computed to be

$$\mathbb{E}[Z_t|H_{t-1}] = \sum_\pi \tilde{Q}_m(\pi)\mathcal{R}(\pi)$$

so

$$\begin{aligned}\mathcal{R}(\pi_\star) - \mathbb{E}[Z_t|H_{t-1}] &= \sum_\pi \tilde{Q}_m(\pi)(\mathcal{R}(\pi_\star) - \mathcal{R}(\pi)) \\ &= \sum_\pi \tilde{Q}_m(\pi)\,\mathrm{Reg}(\pi) \\ &\leq (4\psi + c_0)K\mu_m\end{aligned}$$

by Lemma 14 (under the same assumptions, and using the same constants). Thus, with high probability,

$$\begin{aligned}\sum_{t=1}^{\tau_{M+1}}(\mathcal{R}(\pi_\star) - Z_t) &\leq (4\psi + c_0)K\sum_{t=1}^{\tau_{M+1}}\mu_{m(t)} + \sqrt{2\tau_{M+1}\ln(1/\delta)} \\ &\leq (4\psi + c_0)\sqrt{8KTd_T} + \sqrt{2T\ln(1/\delta)}\end{aligned}$$

by Lemma 16.

Putting these together, and applying the union bound, we find that with probability at least $1 - 2\delta$, for all T (and corresponding M),

$$\sum_{m=1}^{M}(\phi_m^d(Q_m) - \phi_{m+1}^d(Q_m)) \leq O\left(\sqrt{\frac{T}{K}\ln(T/\delta)}\right).$$

Combining the bounds on the separate pieces, we get the bound stated in the lemma.

E Proof of Theorem 4

Recall the earlier definition of the low-variance distribution set

$$\mathcal{Q}_m = \{Q \in \Delta^\Pi : Q \text{ satisfies Eq. (3) in round } \tau_m\}.$$

Fix $\delta \in (0,1)$ and the epoch sequence, and assume M is large enough so $\mu_m = \sqrt{\ln(16\tau_m^2|\Pi|/\delta)/\tau_m}$ for all $m \in \mathbb{N}$ with $\tau_m \geq \tau_M/2$. The low-variance constraint Eq. (3) gives, in round $t = \tau_m$,

$$\widehat{\mathbb{E}}_{x\sim H_t}\left[\frac{1}{Q^{\mu_m}(\pi(x)|x)}\right] \leq 2K + \frac{\widehat{\mathrm{Reg}}_{\tau_m}(\pi)}{\psi\mu_m}, \quad \forall \pi \in \Pi.$$

Below, we use a policy class Π where every policy $\pi \in \Pi$ has no regret ($\mathrm{Reg}(\pi) = 0$), in which case Lemma 13 implies

$$\widehat{\mathbb{E}}_{x\sim H_t}\left[\frac{1}{Q^{\mu_m}(\pi(x)|x)}\right] \leq 2K + \frac{c_0 K\mu_m}{\psi\mu_m} = K\left(2 + \frac{c_0}{\psi}\right), \quad \forall \pi \in \Pi.$$

Applying Lemma 10 (and using our choice of μ_m) gives the following constraints: with probability at least $1-\delta$, for all $m \in \mathbb{N}$ with $\tau_m \geq \tau_M/2$, for all $\pi \in \Pi$,

$$\mathbb{E}_{x\sim\mathcal{D}_X}\left[\frac{1}{\widetilde{Q}^{\mu_m}(\pi(x)|x)}\right] \;\leq\; 81.3K + 6.4K\left(2 + \frac{c_0}{\psi}\right) =: cK \tag{36}$$

(to make Q into a probability distribution $\widetilde{Q}$, the leftover mass can be put on any policy, say, already in the support of Q). That is, with high probability, for every relevant epoch m, every $Q \in \mathcal{Q}_m$ satisfies Eq. (36) for all $\pi \in \Pi$.

Next, we construct an instance with the property that these inequalities cannot be satisfied by a very sparse Q. An instance is drawn uniformly at random from N different contexts denoted as $\{1, 2, \ldots, N\}$ (where we set, with foresight, $N := 1/(2\sqrt{2}cK\mu_M)$). The reward structure in the problem will be extremely simple, with action K always obtaining a reward of 1, while all the other actions obtain a reward of 0, independent of the context. The distribution $\mathcal{D}$ will be uniform over the contexts (with these deterministic rewards). Our policy set Π will consist of $(K-1)N$ separate policies, indexed by $1 \leq i \leq N$ and $1 \leq j \leq K-1$. Policy π_{ij} has the property that

$$\pi_{ij}(x) = \begin{cases} j & \text{if } x = i, \\ K & \text{otherwise.} \end{cases}$$

In words, policy π_{ij} takes action j on context i, and action K on all other contexts. Given the uniform distribution over contexts and our reward structure, each policy obtains an identical reward

$$\mathcal{R}(\pi) = \left(1 - \frac{1}{N}\right)\cdot 1 + \frac{1}{N}\cdot 0 = 1 - \frac{1}{N}.$$

In particular, each policy has a zero expected regret as required.

Finally, observe that on context i, π_{ij} is the unique policy taking action j. Hence we have that $\widetilde{Q}(j|i) = \widetilde{Q}(\pi_{ij})$ and $\widetilde{Q}^{\mu_m}(j|i) = (1 - K\mu_m)\widetilde{Q}(\pi_{ij}) + \mu_m$. Now, let us consider the constraint Eq. (36) for the policy π_{ij}. The left-hand side of this constraint can be simplified as

$$\begin{aligned}
\mathbb{E}_{x\sim\mathcal{D}_X}\left[\frac{1}{\widetilde{Q}^{\mu_m}(\pi(x)|x)}\right] &= \frac{1}{N}\sum_{x=1}^{N}\frac{1}{\widetilde{Q}^{\mu_m}(\pi_{ij}(x)|x)} \\
&= \frac{1}{N}\sum_{x\neq i}\frac{1}{\widetilde{Q}^{\mu_m}(\pi_{ij}(x)|x)} + \frac{1}{N}\cdot\frac{1}{\widetilde{Q}^{\mu_m}(j|i)} \\
&\geq \frac{1}{N}\cdot\frac{1}{\widetilde{Q}^{\mu_m}(j|i)}.
\end{aligned}$$

If the distribution $\widetilde{Q}$ does not put any support on the policy π_{ij}, then $\widetilde{Q}^{\mu_m}(j|i) = \mu_m$, and thus

$$\mathbb{E}_{x\sim\mathcal{D}_X}\left[\frac{1}{\widetilde{Q}^{\mu_m}(\pi(x)|x)}\right] \geq \frac{1}{N}\cdot\frac{1}{\widetilde{Q}^{\mu_m}(j|i)} = \frac{1}{N\mu_m} \geq \frac{1}{\sqrt{2}N\mu_M} > cK$$

(since $N < 1/(\sqrt{2}cK\mu_M)$). Such a distribution $\widetilde{Q}$ violates Eq. (36), which means that every $Q \in \mathcal{Q}_m$ must have $\widetilde{Q}(\pi_{ij}) > 0$. Since this is true for each policy π_{ij}, we see that every $Q \in \mathcal{Q}_m$ has

$$|\mathrm{supp}(Q)| \geq (K-1)N = \frac{K-1}{2\sqrt{2}cK\mu_M} = \Omega\left(\sqrt{\frac{K\tau_M}{\ln(\tau_M|\Pi|/\delta)}}\right)$$

which completes the proof.

F Online Cover algorithm

This section describes the pseudocode of the precise algorithm use in our experiments. The minimum exploration probability μ was set as $0.05\ \min(1/K, 1/\sqrt{tK})$ for our evaluation.

Algorithm 5 Online Cover

input Cover size n, minimum sampling probability μ.

1: Initialize online cost-sensitive minimization oracles $O_1, O_2, \ldots, O_n$, each of which controls a policy $\pi_{(1)}, \pi_{(2)}, \ldots, \pi_{(n)}$; $U :=$ uniform probability distribution over these policies.
2: **for round** $t = 1, 2, \ldots$ **do**
3: Observe context $x_t \in X$.
4: $(a_t, p_t(a_t)) := \mathsf{Sample}(x_t, U, \emptyset, \mu)$.
5: Select action a_t and observe reward $r_t(a_t) \in [0, 1]$.
6: **for each** $i = 1, 2, \ldots, n$ **do**
7: $Q_i := (i-1)^{-1}\sum_{j<i}\mathbb{1}_{\pi_{(j)}}$.
8: $p_i(a) := Q^{\mu}(a|x_t)$.
9: Create cost-sensitive example (x_t, c) where $c(a) = 1 - \frac{r_t(a_t)}{p_t(a_t)}\mathbb{1}\{a = a_t\} - \frac{\mu}{p_i(a)}$.
10: Update $\pi_{(i)} = O_i(x, c)$
11: **end for**
12: **end for**

Two additional details are important in step 9:

1. We pass a cost vector rather than a reward vector to the oracle since we have a loss minimization rather than a reward maximization oracle.
2. We actually used a doubly robust estimate Dudík et al. (2011b) with a linear reward function that was trained in an online fashion.