Efficient Optimal Learning for Contextual Bandits

Miroslav Dudik mdudik@yahoo-inc.com
Daniel Hsu djhsu@rci.rutgers.edu
Satyen Kale skale@yahoo-inc.com
Nikos Karampatziakis nk@cs.cornell.edu
John Langford jl@yahoo-inc.com
Lev Reyzin lreyzin@cc.gatech.edu
Tong Zhang tzhang@stat.rutgers.edu

Abstract

We address the problem of learning in an online setting where the learner repeatedly observes features, selects among a set of actions, and receives reward for the action taken. We provide the first efficient algorithm with an optimal regret. Our algorithm uses a cost sensitive classification algorithm as an oracle and has a running time polylog(N), where N is the number of classifiers among which the oracle might choose. This is exponentially faster than all previous algorithms that achieve optimal regret in this setting. Our formulation also enables us to create an algorithm with regret that is additive rather than multiplicative in feedback delay as in all previous work.

1 INTRODUCTION

The contextual bandit setting consists of the following loop repeated indefinitely:

1. The world presents context information as features x.
2. The learning algorithm chooses an action a from K possible actions.
3. The world presents a reward r for the action.

The key difference between the contextual bandit setting and standard supervised learning is that *only* the reward of the chosen action is revealed. For example, after always choosing the same action several times in a row, the feedback given provides almost no basis to prefer the chosen action over another action. In essence, the contextual bandit setting captures the difficulty of exploration while avoiding the difficulty of credit assignment as in more general reinforcement learning settings.

The contextual bandit setting is a half-way point between standard supervised learning and full-scale reinforcement learning where it appears possible to construct algorithms with convergence rate guarantees similar to supervised learning. Many natural settings satisfy this half-way point, motivating the investigation of contextual bandit learning. For example, the problem of choosing interesting news articles or ads for users by internet companies can be naturally modeled as a contextual bandit setting. In the medical domain where discrete treatments are tested before approval, the process of deciding which patients are eligible for a treatment takes context into account. More generally, we can imagine that in a future with personalized medicine, new treatments are essentially equivalent to new actions in a contextual bandit setting.

In the i.i.d. setting, the world draws $(x, \vec{r})$ from some unknown distribution D, revealing x in step 1 and the reward $r(a)$ of the chosen action a in step 3. Given a set of policies $\Pi = \{\pi : X \to A\}$, the goal is to create an algorithm for step 2 which competes with the set of policies. We measure our success by comparing the algorithm's cumulative reward to the expected cumulative reward of the best policy in the set.

All existing algorithms for this setting either achieve a suboptimal regret (Langford and Zhang, 2007) or require computation linear in the number of policies (Auer et al., 2002b; Beygelzimer et al., 2010). In unstructured policy spaces, this computational complexity is the best one can hope for. On the other hand, in the case where the rewards of all actions are revealed, the problem is equivalent to cost-sensitive classification, and we know of algorithms to efficiently search the space of policies (classifiers) such as cost-sensitive logistic regression and support vector machines. In these cases, the space of classifiers is exponential in the number of features, but these problems can be efficiently solved using convex optimization.

Our goal here is to efficiently solve the contextual bandit problems for similarly large policy spaces. We do this by reducing the contextual bandit problem to cost-sensitive classification. Given a supervised cost-sensitive learning algorithm as an oracle (Beygelzimer et al., 2009), our algorithm runs in time only polylog(N) while achieving regret $O(\sqrt{TK \ln N})$, where N is the number of possible policies (classifiers), K is the number of actions (classes), and T is the number of time steps. This efficiency is achieved in a modular way, so any future improvement in cost sensitive learning immediately applies here.

1.1 PREVIOUS WORK AND MOTIVATION

All previous regret-optimal approaches are *measure* based—they work by updating a measure over policies, an operation which is linear in the number of policies. In contrast, regret guarantees scale only logarithmically in the number of policies. If not for this computational bottleneck, these regret guarantees imply that we could dramatically increase performance in contextual bandit settings using more expressive policies. We overcome the computational bottleneck using an optimization based algorithm which works by choosing optimal policies rather than keeping track of a measure over policies.

In a more difficult version of contextual bandits, an adversary chooses $(x, \vec{r})$ given knowledge of the learning algorithm (but not any random numbers). All known regret optimal solutions in this setting are variants of the EXP4 algorithm (Auer et al., 2002b). EXP4 achieves the same regret rate as our algorithm: $O\left(\sqrt{KT \ln N}\right)$, where T is the number of time steps, K is the number of actions available in each time step, and N is the number of policies.

Why not use EXP4 in the i.i.d. setting? For example, it is known that the algorithm can be modified to succeed with high probability (Beygelzimer et al., 2010), and also for VC classes when the adversary is constrained to i.i.d. sampling. There are two central benefits that we may hope to realize from an i.i.d. argument style.

1. Computational Tractability. Even when the reward vector is fully known, regrets scale as $O\left(\sqrt{\ln N}\right)$ while computation scales as $O(N)$ in general. One attempt to get around this is the follow-the-perturbed-leader algorithm (Kalai and Vempala, 2005) which provides a computationally tractable solution in certain special-case structures. This algorithm has no mechanism for efficient application to arbitrary policy spaces, even given an efficient optimization oracle over the policy space. An efficient optimization oracle has been shown effective in transductive settings (Kakade and Kalai, 2005). Aside from the drawback of requiring a transductive setting, the regret achieved here is substantially worse than for EXP4.
2. Improved Rates. When the world is not completely adversarial, it's possible to achieve substantially lower regrets than are possible with algorithms optimized for the adversarial setting. For example, in supervised learning, it's well known that regrets scaling as $O(\log(T))$ with a problem dependent constant are possible. When the feedback is delayed by τ rounds, lower bounds imply that the regret in the adversarial setting increases by a multiplicative $\sqrt{\tau}$ while in the i.i.d. setting, it's known that an additive regret of τ is possible.

In the i.i.d. analysis setting, the previous-best approach was given by ϵ-greedy and epoch greedy algorithms (Langford and Zhang, 2007) which have a regret scaling as $O(T^{2/3})$ in the worst case.

There have also been many special-case analyses. For example, in the context-free setting (Lai and Robbins, 1985; Auer et al., 2002a; Even-Dar et al., 2006) a theory is well understood. Similarly when rewards are known to be linear in features (Auer, 2002) or according to a Gaussian Process (Srinivas et al., 2010) good algorithms are known.

1.2 WHAT WE PROVE

In Section 3 we state the Policy_Elimination algorithm, and prove the following regret bound for it.

Theorem 4. For all distributions D over K actions and features, for all sets of N policies Π, with probability at least $1-\delta$, the regret of the Policy_Elimination algorithm (Algorithm 1) over T rounds is at most

$$16\sqrt{2TK \ln \frac{4T^2N}{\delta}}.$$

This result can easily be extended to deal with VC classes, as well as other special cases. It forms the simplest method we have of exhibiting the new analysis.

The new key element of this algorithm is a mechanism for constructing a distribution over actions $W(a|x)$ via a distribution over policies $P(\pi)$ which simultaneously achieves small expected regret and small variance in the estimated value of every policy. The key insight here boils down to a game between an adversary and

the algorithm. A minimax theorem *nonconstructively* shows the value of this game.

The Policy_Elimination algorithm is computationally intractable and also requires the learner to have knowledge of the unlabeled data distribution. We show how to address these issues in Section 4 using an algorithm we call Randomized_UCB. Namely, we prove the following theorem.

Theorem 5. For all distributions D over K actions and features, for all sets of N policies Π, with probability at least $1-\delta$, the regret of the Randomized UCB algorithm (Algorithm 2) over T rounds is at most

$$O\left(\sqrt{TK\log\left(TN/\delta\right)}+K\log(NK/\delta)\right).$$

Randomized_UCB's analysis is substantially more complex, with a key subroutine being an application of the ellipsoid algorithm on an optimization oracle (described in Section 5). The Randomized_UCB algorithm also works with its own existing history of unlabeled data points and, unlike Policy_Elimination, does not have access to the unlabeled data distribution. Modifying the proof in this manner requires a covering argument over the distributions over policies which uses the probabilistic method. The net result is an algorithm with a similar analysis that has only a logarithmic dependence on the number of policies given an optimization oracle.

Theorem 11. In each time step t, the Randomized UCB algorithm makes at most $O(\text{poly}(t,K,\log(1/\delta),\log N))$ calls to an optimization oracle, and requires additional $O(\text{poly}(t,K,\log N))$ processing time.

Another key advantage of this style of analysis is the ability to prove tighter results than for adversarial settings. We provide one example of this for the common setting where reward feedback is delayed by τ rounds in Section 6. Here, a straightforward modification of the Policy_Elimination algorithm yields a regret only τ larger than in the delay-free setting, namely we prove the following.

Theorem 12. For all distributions D over K actions and features, for all sets of N policies Π, and all delay factors τ, with probability at least $1-\delta$, the regret of the Delayed Policy Elimination algorithm (Algorithm 3) is at most

$$16\sqrt{2K\ln\frac{4T^2N}{\delta}}\left(\tau+\sqrt{T}\right).$$

We start next with precise settings and definitions in Section 2.

2 SETTING AND DEFINITIONS

2.1 THE SETTING

Let A be the set of K actions, let X be the domain of contexts x_t, and let D be an arbitrary joint distribution on $(x,\vec{r})$. We denote the marginal distribution of D over X by D_X.

We denote Π to be a finite set of policies $\{\pi : X \to A\}$, as in each policy π, predicts according to $\pi(x_t)$, where x_t is the context available in round t. The cardinality of Π is denoted by N. Let $\vec{r}_t \in [0,1]^K$ be the vector of rewards, where $r_t(a)$ is the reward of action a on round t.

In the i.i.d. setting, on each round $t = 1\ldots T$, the world chooses $(x_t,\vec{r}_t)$ i.i.d. according to D and reveals x_t to the learner. Then the learner, having access to Π, chooses action $a_t \in \{1,\ldots,K\}$. Finally, the world reveals reward $r_t(a_t)$ (which we call r_t for short) to the learner, and this game proceeds to the next round. The number of rounds T is not known in advance to the learner.

The goal of the learner is to minimize its regret to the best policy in Π. This notion is defined in Equation 2.1.

2.2 EXPECTED AND EMPIRICAL REWARDS

Let the expected instantaneous **reward** of a policy $\pi \in \Pi$ be denoted by

$$\eta_D(\pi) \doteq \mathop{\mathbb{E}}_{(x,\vec{r})\sim D}[r(\pi(x))].$$

The best policy $\pi_{\max} \in \Pi$ is that which maximizes $\eta_D(\pi)$. More formally,

$$\pi_{\max} \doteq \mathop{\text{argmax}}_{\pi\in\Pi} \eta_D(\pi).$$

We can define h_t to be the **history** at time t that the learner has seen. Specifically

$$h_t = \bigcup_{t'=1\ldots t}(x_{t'},a_{t'},r_{t'}(a_{t'}),p_{t'}),$$

where $p_{t'}$ is the probability of the algorithm choosing action $a_{t'}$ at time t'. Note that $a_{t'}$ and $p_{t'}$ are given by alogrithm while $x_{t'},r_{t'}$ are given by nature. We denote choosing x uniformly at random from the x's in history h, by $x \sim h$.

Using the history of past actions and probabilities with which they were taken, we can form an unbiased estimate of the policy value:

$$\eta_t(\pi) \doteq \frac{1}{t}\sum_{(x,a,r,p)\in h_t}\frac{rI(\pi(x)=a)}{p}.$$

The unbiasedness follows, because $\mathbb{E}_{a\sim p}\frac{rI(\pi(x)=a)}{p(a)} = \sum_a p(a)\frac{rI(\pi(x)=a)}{p(a)} = r_{\pi(x)}$.

And we denote π_t to be empirically the best policy at time t

$$\pi_t \doteq \operatorname*{argmax}_{\pi\in\Pi} \eta_t(\pi).$$

2.3 REGRET

The goal of this work is to obtain a learner that has small **regret** to the expected performance of $\pi_{\max}$ over T rounds, which is

$$\sum_{t=1}^{T} \left(\eta_D(\pi_{\max}) - r_t\right). \tag{2.1}$$

We say that the regret of the learner over T rounds is bounded by ϵ with probability at least $1-\delta$, if

$$\Pr\left[\sum_{t=1}^{T} \left(\eta_D(\pi_{\max}) - r_t(a_t)\right) \le \epsilon\right] \ge 1-\delta$$

where the probability is taken with respect to the random pairs $(x_t, \vec{r}_t) \sim D$ for $t = 1, \ldots, T$, as well as any internal randomness used by the learner.

We can also define notions of regret for policies π. $\forall \pi \in \Pi$, let

$$\Delta_D(\pi) = \eta_D(\pi_{\max}) - \eta_D(\pi).$$

and

$$\Delta_t(\pi) = \eta_t(\pi_t) - \eta_t(\pi).$$

Our algorithms work by choosing distributions over policies, which in turn then induce distributions over actions. For any distribution P over policies Π, let $W_P(x,a)$ denote the induced conditional distribution over actions a given the context x:

$$W_P(x,a) \doteq \sum_{\pi\in\Pi:\pi(x)=a} P(\pi). \tag{2.2}$$

In general, we shall use W, W' and Z as probability distributions over the actions A, namely

$$\{W, W', Z : X \times A \to p(A)\}.$$

We shall think of W' as a dampened version of W with a minimum action probability of μ (to be defined by the algorithm), such that $\forall x \in X, a \in A$

$$W'(x,a) = (1-K\mu)W(x,a) + \mu.$$

We can now define notions of regret also for probability distributions W (and W', Z, etc.). Let

$$\Delta_D(W) \doteq \eta_D(\pi_{\max}) - \eta_D(W)$$

Algorithm 1 Policy_Elimination(Π,δ,K,D_X)

Let $\Pi_0 = \Pi$ and history $h_0 = \emptyset$

Define: $\delta_t \doteq \delta \,/\, 4Nt^2$

Define: $b_t \doteq 2\sqrt{\dfrac{2K\ln(1/\delta_t)}{t}}$

Define: $\mu_t \doteq \min\left\{\dfrac{1}{2K}, \sqrt{\dfrac{\ln(1/\delta_t)}{2Kt}}\right\}$

For each timestep $t = 1 \ldots T$, observe x_t and do:

1. Choose distribution P_t over Π_{t-1} s.t. $\forall\, \pi \in \Pi_{t-1}$:
$$\mathop{\mathbb{E}}_{x\sim D_X}\left[\frac{1}{(1-K\mu_t)W_{P_t}(x,\pi(x)) + \mu_t}\right] \le 2K$$
2. Let $W'_t(a) = (1-K\mu_t)W_{P_t}(x_t,a) + \mu_t$ for all $a \in A$
3. Choose $a_t \sim W'_t$
4. Observe reward r_t
5. Let $\Pi_t = \Big\{\pi \in \Pi_{t-1} : \eta_t(\pi) \ge \Big(\max_{\pi'\in\Pi_{t-1}} \eta_t(\pi')\Big) - 2b_t\Big\}$
6. Let $h_t = h_{t-1} \cup (x_t, a_t, r_t, W'_t(a_t))$

and

$$\Delta_t(W) \doteq \eta_t(\pi_t) - \eta_t(W).$$

with

$$\eta_D(W) \doteq \mathop{\mathbb{E}}_{(x,\vec{r})\sim D}[\vec{r}\cdot W(x)]$$

and

$$\eta_t(W) \doteq \frac{1}{t}\sum_{(x,a,r,p)\in h_t} \frac{rW(x,a)}{p}.$$

3 POLICY ELIMINATION

In Algorithm 1, we present the algorithm Policy_Elimination, which demonstrates the basic ideas behind our approach.

The key insight which allows this algorithm to work is Step 1, which essentially finds a distribution over policies which induces low variance in the estimate of the value of all policies. We prove below that this is always possible using a minimax theorem. How to find this distribution is not specified here, although we discuss one method in Section 5 based on the ellipsoid algorithm.

Step 2 then projects this distribution over actions, and mixes with the uniform distribution over actions.

Finally, Step 5 eliminates the policies that have been determined to be suboptimal (with high probability).

In addition to proving a bound on regret, in our analysis to follow we address this point explicitly – that all policies within $4b_t$ of the optimal remain uneliminated after step 5.

ALGORITHM ANALYSIS

The following minimax theorem considers randomized policies. A randomized policy W is a map $W : X \times A \to [0,1]$ where $W(x,a)$ is the probability of choosing action a on a context x. Note that W_P from Eq. (2.2) is an example of a randomized policy.

Lemma 1. *Let $\mathcal{C}$ be a compact and convex set of randomized policies. Let $\mu \in (0, 1/K]$ and for any $W \in \mathcal{C}$, $W'(x,a) \doteq (1-K\mu)W(x,a) + \mu$. Then for all distributions D,*

$$\min_{W\in\mathcal{C}} \max_{Z\in\mathcal{C}} \mathop{\mathbb{E}}_{x\sim D_X} \mathop{\mathbb{E}}_{a\sim Z(x,\cdot)} \left[\frac{1}{W'(x,a)}\right] \le \frac{K}{1-K\mu} \ .$$

Proof. Let $f(W,Z) \doteq \mathbb{E}_{x\sim D_X} \mathbb{E}_{a\sim Z(x,\cdot)}[1/W'(x,a)]$ denote the inner expression of the minimax problem. Note that $f(W,Z)$ is:

- *everywhere defined*: Since $W'(x,a) \ge \mu$, we obtain that $1/W'(x,a) \in [0, 1/\mu]$, hence the expectations are defined for all W and Z.
- *linear in Z*: Linearity follows from rewriting $f(W,Z)$ as

$$f(W,Z) = \mathop{\mathbb{E}}_{x\sim D_X} \sum_{a\in A} \left[\frac{Z(x,a)}{W'(x,a)}\right].$$

- *convex in W*: Note that $1/W'(x,a)$ is convex in $W(x,a)$ by convexity of $1/(c_1 w + c_2)$ in $w \ge 0$, for $c_1 \ge 0$, $c_2 > 0$. Convexity of $f(W,Z)$ in W then follows by taking expectations over x and a.

Hence, by Theorem 14 (in Appendix B), min and max can be reversed without affecting the value:

$$\min_{W\in\mathcal{C}} \max_{Z\in\mathcal{C}} f(W,Z) = \max_{Z\in\mathcal{C}} \min_{W\in\mathcal{C}} f(W,Z) \ .$$

The right-hand side can be further upper-bounded by $\max_{Z\in\mathcal{C}} f(Z,Z)$, which is upper-bounded by

$$\begin{aligned} f(Z,Z) &= \mathop{\mathbb{E}}_{x\sim D_X} \sum_{a\in A} \left[\frac{Z(x,a)}{Z'(x,a)}\right] \\ &\le \mathop{\mathbb{E}}_{x\sim D_X} \sum_{a\in A} \left[\frac{Z(x,a)}{(1-K\mu)Z(x,a)}\right] = \frac{K}{1-K\mu} \end{aligned}$$

□

Corollary 2. *The set of distributions satisfying constraints of Step 1 is non-empty.*

Lemma 1 and Corollary 2 establish existence of a distribution P_t in Step 1. As we will see below, the constraints in Step 1 ensure low variance of the policy value estimator $\eta_h(\pi)$ for all $\pi \in \Pi_{t-1}$. The small variance is in turn used to ensure accuracy of policy elimination in Step 5 as quantified in the following lemma:

Lemma 3. *With probability at least $1-\delta$, for all t:*

1. $\pi_{\max} \in \Pi_t$
2. $\eta_D(\pi_{\max}) - \eta_D(\pi) \le 4b_t$ *for all* $\pi \in \Pi_t$

Proof. We will show that for any policy $\pi \in \Pi_{t-1}$, the probability that $\eta_t(\pi)$ deviates from $\eta_D(\pi)$ by more that b_t is at most δ_t. Taking the union bound over all policies and all time steps we find that with probability at least $1-\delta$,

$$|\eta_t(\pi) - \eta_D(\pi)| \le b_t \tag{3.1}$$

for all t and all $\pi \in \Pi_{t-1}$. Then:

1. By the triangle inequality, in each time step, $\eta_t(\pi_t) \le \eta_t(\pi_{\max}) + 2b_t$ for all $\pi \in \Pi_{t-1}$, yielding the first part of the lemma.
2. Also by the triangle inequality, if $\eta_D(\pi) < \eta_D(\pi_{\max}) + 4b_t$ for $\pi \in \Pi_{t-1}$, then $\eta_t(\pi) < \eta_t(\pi_{\max}) + 2b_t$. Hence the policy π is eliminated in Step 5, yielding the second part of the lemma.

It remains to show Eq. (3.1). We fix the policy $\pi \in \Pi$ time t, and show that the deviation bound is violated with probability at most δ_t. Our argument rests on Freedman's inequality (see Theorem 13 in Appendix A). Let

$$y_t = \frac{r_t \mathbb{I}(\pi(x_t) = a_t)}{W'_t(a_t)} \ ,$$

i.e., $\eta_t(\pi) = (\sum_{t'=1}^{t} y_{t'})/t$. Let $\mathbb{E}_t$ denote the conditional expectation $\mathbb{E}[\,\cdot \mid h_{t-1}]$. To use Freedman's inequality, we need to bound the range of y_t and its conditional second moment $\mathbb{E}_t[y_t^2]$.

Since $r_t \in [0,1]$ and $W'_t(a_t) \ge \mu_t$, we have the bound

$$0 \le y_t \le 1/\mu_t \doteq R_t \ .$$

Next,

$$\begin{aligned} \mathbb{E}_t[y_t^2] &= \mathop{\mathbb{E}}_{(x_t,\vec{r}_t)\sim D} \mathop{\mathbb{E}}_{a_t\sim W'_t} \left[y_t^2\right] \\ &= \mathop{\mathbb{E}}_{(x_t,\vec{r}_t)\sim D} \mathop{\mathbb{E}}_{a_t\sim W'_t} \left[\frac{r_t^2 \mathbb{I}(\pi(x_t) = a_t)}{W'_t(a_t)^2}\right] \\ &\le \mathop{\mathbb{E}}_{(x_t,\vec{r}_t)\sim D} \left[\frac{W'_t(\pi(x_t))}{W'_t(\pi(x_t))^2}\right] \qquad (3.2) \\ &= \mathop{\mathbb{E}}_{x_t\sim D} \left[\frac{1}{W'_t(\pi(x_t))}\right] \le 2K \ . \qquad (3.3) \end{aligned}$$

where Eq. (3.2) follows by boundedness of r_t and Eq. (3.3) follows from the constraints in Step 1. Hence,

$$\sum_{t'=1}^{t} \mathbb{E}_{t'}[y_{t'}^2] \leq 2Kt \doteq V_t \ .$$

Since $K \geq 2$ and $(\ln t)/t$ is decreasing for $t \geq 3$, we obtain that μ_t is non-decreasing. Let t_0 be the first t such that $\mu_t < 1/2K$. Note that $b_t \geq 4K\mu_t$, so for $t < t_0$, $b_t \geq 2$. and $\Pi_t = \Pi$. Hence, the deviation bound holds for $t < t_0$.

Let $t \geq t_0$. For $t' \leq t$, by the monotonicity of μ_t

$$R_{t'} = 1/\mu_{t'} \leq 1/\mu_t = \sqrt{\frac{2Kt}{\ln(1/\delta_t)}} = \sqrt{\frac{V_t}{\ln(1/\delta_t)}} \ .$$

Hence, the assumptions of Theorem 13 are satisfied, and

$$\Pr\left[|\eta_t(\pi) - \eta_D(\pi)| \geq b_t\right] \leq \delta_t \ .$$

The union bound over π and t then yields Eq. (3.1). □

This immediately implies that the cumulative regret is bounded by

$$\begin{aligned} \sum_{t=1}^{T} (\eta_D(\pi_{\max}) - r_t) &\leq 8\sqrt{2K \ln \frac{4NT^2}{\delta}} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \\ &\leq 16\sqrt{2TK \ln \frac{4T^2N}{\delta}} \end{aligned} \quad (3.4)$$

and gives us the following theorem.

Theorem 4. *For all distributions D over K actions and features, for all sets of N policies Π, with probability at least $1-\delta$, the regret of the Policy_Elimination algorithm (Algorithm 1) over T rounds is at most*

$$16\sqrt{2TK \ln \frac{4T^2N}{\delta}}.$$

4 THE RANDOMIZED UCB ALGORITHM

Policy Elimination is the simplest exhibition of the minimax argument, but it has some inherent drawbacks:

1. The algorithm keeps explicit track of the space of good policies (like a version space), and therefore it is computationally difficult to implement in general.
2. If the optimal policy is mistakenly eliminated by chance, the algorithm can never recover.
3. The algorithm requires perfect knowledge of the distribution D_X over contexts.

These difficulties are addressed by the Randomized UCB (RUCB) algorithm, which we present and analyze in this section. The essential idea here is to have a UCB style algorithm where instead of choosing the highest upper confidence bound, we randomize over choices according to the value of their empirical performance. The algorithm has the following properties:

1. The optimization required by the algorithm always considers the full set of policies (*i.e.*, explicit tracking of the set of good policies is avoided), and thus it can be efficiently implemented using an ERM-type oracle. We discuss this further in Section 5.
2. Suboptimal policies are implicitly used with de-

Algorithm 2 RUCB(Π,δ,K)

Let $h_0 \doteq \emptyset$ be the initial history.
Define the following quantities:

$$C_t \doteq 2\log\left(\frac{Nt}{\delta}\right) \quad \text{and} \quad \mu_t \doteq \min\left\{\frac{1}{2K}, \sqrt{\frac{C_t}{2Kt}}\right\}.$$

For each timestep $t = 1 \ldots T$, observe x_t and do:

1. Let P_t be a distribution over Π that approximately solves the optimization problem

$$\begin{aligned} &\min_P \sum_{\pi \in \Pi} P(\pi)\Delta_{t-1}(\pi) \\ &\text{s.t.} \quad \text{for all distributions } Q \text{ over } \Pi: \\ &\mathop{\mathbb{E}}_{\pi \sim Q}\left[\frac{1}{t-1}\sum_{i=1}^{t-1} \frac{1}{(1-K\mu_t)W_P(x_i,\pi(x_i)) + \mu_t}\right] \\ &\qquad \leq \max\left\{4K, \ \frac{(t-1)\Delta_{t-1}(W_Q)^2}{180C_{t-1}}\right\} \end{aligned} \quad (4.1)$$

   so that the objective value at P_t is within $\varepsilon_{\text{opt},t} = O(\sqrt{KC_t/t})$ of the optimal value, and so that each constraint is satisfied with slack $\leq K$.
2. Let W_t' be the distribution over A given by

$$W_t'(a) \doteq (1-K\mu_t)W_{P_t}(x_t, a) + \mu_t$$

   for all $a \in A$.
3. Choose $a_t \sim W_t'$.
4. Observe reward r_t.
5. Let $h_t \doteq h_{t-1} \cup (x, a_t, r_t, W_t'(a_t))$.

creasing frequency by using a non-uniform variance constraint that depends on a policy's estimated regret. A consequence of this is a bound on the value of the optimization, stated in Lemma 6 below.

3. The history of previously seen contexts is used as a surrogate for the distribution over contexts in the optimization. We discuss this in Subsection 4.2.

The Randomized UCB algorithm has the following performance guarantee.

Theorem 5. *For all distributions D over K actions and features, for all sets of N policies Π, with probability at least $1-\delta$, the regret of the Randomized UCB algorithm (Algorithm 2) over T rounds is at most*

$$O\left(\sqrt{TK\log\left(TN/\delta\right)} + K\log(NK/\delta)\right).$$

The proof is given in Appendix C (in the full version).

4.1 OVERVIEW OF THE ANALYSIS

Central to the analysis is the following lemma that bounds the value of the optimization in each round.

Lemma 6. *If OPT_t is the value of the optimization problem (4.1) in round t, then*

$$\mathrm{OPT}_t \;\leq\; O\left(\sqrt{\frac{KC_{t-1}}{t-1}}\right) \;=\; O\left(\sqrt{\frac{K\log(Nt/\delta)}{t}}\right).$$

This lemma implies that the algorithm is always able to select a distribution over the policies that focuses mostly on the policies with low estimated regret. Moreover, the variance constraints ensure that good policies never appear too bad, and that only bad policies are allowed to incur high variance in their reward estimates. Hence, minimizing the objective in (4.1) is an effective surrogate for minimizing regret.

The bulk of the analysis consists of analyzing the variance of the importance-weighted reward estimates $\eta_t(\pi)$, and showing how they relate to their actual expected rewards $\eta_D(\pi)$. The details are deferred to Appendix C (in the full version).

4.2 EMPIRICAL VARIANCE ESTIMATES

For a distribution P over policies Π and a particular policy $\pi \in \Pi$, define

$$V_{P,\pi,t} = \mathop{\mathbb{E}}_{x\sim D_X}\left[\frac{1}{(1-K\mu_t)W_P(x,\pi(x))+\mu_t}\right]$$

$$\widehat{V}_{P,\pi,t} = \frac{1}{t-1}\sum_{i=1}^{t-1}\frac{1}{(1-K\mu_t)W_P(x_i,\pi(x_i))+\mu_t}.$$

The first quantity $V_{P\pi,t}$ is (a bound on) the variance incurred by an importance-weighted estimate of reward in round t using the action distribution induced by P, and the second quantity $\widehat{V}_{P,\pi,t}$ is an empirical estimate of $V_{P,\pi,t}$ using the finite sample $\{x_1,\ldots,x_{t-1}\} \subseteq X$ drawn from D_X. We show that for all distributions P and all $\pi \in \Pi$, $\widehat{V}_{P,\pi,t}$ is close to $V_{P,\pi,t}$ with high probability.

Theorem 7. *For any $\epsilon \in (0,1)$, with probability at least $1-\delta$,*

$$V_{P,\pi,t} \leq (1+\epsilon)\cdot\widehat{V}_{P,\pi,t} + \frac{7500}{\epsilon^3}\cdot K$$

for all distributions P over Π, all $\pi \in \Pi$, and all $t \geq 16K\log(8KN/\delta)$.

The proof appears in Appendix D (in the full version).

5 USING AN ORACLE DIRECTLY

Assume that we have access to an arg-max oracle, which when supplied with a set of examples consisting of contexts and rewards, returns the policy that maximizes the expected reward:

Definition 1. There is an algorithm, $\mathcal{AMO}$, which when given any history $h = (X\times\mathbb{R}^k)^*$, computes

$$\mathcal{AMO}(h) \;:=\; \arg\max_{\pi\in\Pi}\mathop{\mathbb{E}}_{(x,\vec{r})\sim h}[r(\pi(x))].$$

Here, $(x,\vec{r}) \sim h$ is used to denote a uniform random draw from h.

We now show that there is an algorithm running in polynomial time *independent*[1] of the number of policies, which make queries to $\mathcal{AMO}$ to compute a distribution over policies suitable for the optimization step of algorithm 2.

This algorithm relies on the ellipsoid method. The ellipsoid method is a general technique for solving convex programs equipped with a separation oracle. A separation oracle is defined as follows:

Definition 2. Let S be a convex set in $\mathbb{R}^n$. A separation oracle for S is an algorithm that, given a point $x \in \mathbb{R}^n$, either declares correctly that $x \in S$, or produces a hyperplane H such that x and S are on opposite sides of H.

The following lemma gives a specification of the ellipsoid method. For a point $x \in \mathbb{R}^n$ and $r \geq 0$, we use the notation $B(x,r)$ to denote the ℓ_2 ball of radius r centered at x.

[1] Or rather dependent only on $\log N$, the representation complexity of a policy.

Lemma 8. *Suppose we are required to decide whether a convex set $S \subseteq \mathbb{R}^n$ is empty or not. Assume that we are given a separation oracle for S. Assume further that we are given two numbers R and r, such that $S \in B(0, R)$ and if S is non-empty, then there is a point $x^\star$ such that $S \supseteq B(x^\star, r)$. Then there is an iterative algorithm with at most $O(n^2 \log(\frac{R}{r}))$ iterations, each involving one call to the separation oracle and additional $O(n^2)$ processing time, that decides correctly if S is empty or not.*

We now write a convex program whose solution is the required distribution, and show how to solve it using the ellipsoid method by giving a separation oracle for its feasible set using $\mathcal{AMO}$.

Fix a time period t. Let $\mathcal{X}_{t-1}$ be the set of all contexts seen so far, i.e. $\mathcal{X}_{t-1} = \{x_1, x_2, \ldots, x_{t-1}\}$. We embed all policies $\pi \in \Pi$ in $\mathbb{R}^{(t-1)K}$, with coordinates identified with $(x, a) \in \mathcal{X}_{t-1} \times A$. With abuse of notation, a policy π is represented by the vector π with coordinate $\pi(x, a) = 1$ if $\pi(x) = a$ and 0 otherwise. Let $\mathcal{C}$ be the convex hull of all policy vectors π. Recall that a distribution P over policies corresponds to a point inside $\mathcal{C}$, i.e. $W_P(x, a) = \sum_{\pi:\pi(x)=a} P(\pi)$, and that $W'(x, a) = (1 - \mu_t K)W(x, a) + \mu_t$, where μ_t is as defined in Algorithm 2. Also define $\beta_t = \frac{t-1}{180 C_{t-1}}$. In the following, we use the notation $x \sim h_{t-1}$ to denote a context drawn uniformly at random from $\mathcal{X}_{t-1}$.

Consider the following convex program:

$$\begin{aligned}
&\min\ s \text{ s.t.} \\
&\Delta_{t-1}(W) \ \leq \ s && (5.1)\\
&W \ \in \ \mathcal{C} && (5.2)\\
&\forall Z \in \mathcal{C}: \\
&\mathop{\mathbb{E}}_{x \sim h_{t-1}} \left[\sum_a \frac{Z(x, a)}{W'(x, a)} \right] \leq \max\{4K, \beta_t \Delta_{t-1}(Z)^2\}, && (5.3)
\end{aligned}$$

We claim that this program is equivalent to the RUCB optimization problem (4.1), up to finding an explicit distribution over policies which corresponds to the optimal solution. This can be seen as follows. Since we require $W \in \mathcal{C}$, it can be interpreted as being equal to W_P for some distribution over policies P. The constraints (5.3) are equivalent to requiring that for all distributions Q over Π, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\pi \sim Q} \left[\frac{1}{t-1} \sum_{i=1}^{t-1} \frac{1}{(1 - K\mu_t) W_P(x_i, \pi(x_i)) + \mu_t} \right] \\
&\qquad \leq \max \left\{ 4K, \ \frac{(t-1)\Delta_{t-1}(W_Q)^2}{180 C_{t-1}} \right\}
\end{aligned}$$

This can be seen by setting $Z = W_Q$.

The above convex program can be solved by performing a binary search over s and testing feasibility of the constraints (5.1)–(5.3). For a fixed value of s, the feasibility problem defined by these constraints is denoted by $\mathcal{A}$.

We now give a sketch of how we construct a separation oracle for the feasible region of $\mathcal{A}$. The details of the algorithm are a bit complicated due to the fact that we need to ensure that the feasible region, when non-empty, has a non-negligible volume (recall the requirements of Lemma 8). This necessitates having a small error in satisfying the constraints of the program. We leave the details to Appendix E (in the full version). Modulo these details, the construction of the separation oracle essentially implies that we can solve $\mathcal{A}$.

Before giving the construction of the separation oracle, we first show that $\mathcal{AMO}$ allows us to do linear optimization over $\mathcal{C}$ efficiently:

Lemma 9. *Given a vector $w \in \mathbb{R}^{(t-1)K}$, we can compute $\arg\max_{Z \in \mathcal{C}} w \cdot Z$ using one invocation of $\mathcal{AMO}$.*

Proof. This follows directly from the following fact:

$$\begin{aligned}
&\arg\max_{Z \in \mathcal{C}} w \cdot Z = \arg\max_{\pi \in \Pi} w \cdot \pi \\
&= \arg\max_{\pi \in \Pi} \mathop{\mathbb{E}}_{x \sim h_{t-1}} [w(x, \pi(x))]. \square
\end{aligned}$$

We need another simple technical lemma which explains how to get a separating hyperplane for violations of convex constraints:

Lemma 10. *For $x \in \mathbb{R}^n$, let $f(x)$ be a convex function of x, and consider the convex set K defined by $K = \{x : \ f(x) \leq 0\}$. Suppose we have a point y such that $f(y) > 0$. Let $\nabla f(y)$ be a subgradient of f at y. Then the hyperplane $f(y) + \nabla f(y) \cdot (x - y) = 0$ separates y from K.*

Proof. Let $g(x) = f(y) + \nabla f(y) \cdot (x - y)$. By the convexity of f, we have $f(x) \geq g(x)$ for all x. Thus, for any $x \in K$, we have $g(x) \leq f(x) \leq 0$. Since $g(y) = f(y) > 0$, we conclude that $g(x) = 0$ separates y from K. $\square$

Now given a candidate point W, a separation oracle can be constructed as follows. We check whether W satisfies the constraints of $\mathcal{A}$. If any constraint is violated, then we find a hyperplane separating W from all points satisfying the constraint.

1. First, for constraint (5.1), note that $\eta_{t-1}(W)$ is linear in W, and so we can compute $\max_\pi \eta_{t-1}(\pi)$ via $\mathcal{AMO}$ as in Lemma 9. We can then compute $\eta_{t-1}(W)$ and check if the constraint is satisfied. If

not, then the constraint, being linear, automatically yields a separating hyperplane.

2. Next, we consider constraint (5.2). To check if $W \in \mathcal{C}$, we use the perceptron algorithm. We shift the origin to W, and run the perceptron algorithm with all points $\pi \in \Pi$ being positive examples. The perceptron algorithm aims to find a hyperplane putting all policies $\pi \in \Pi$ on one side. In each iteration of the perceptron algorithm, we have a candidate hyperplane (specified by its normal vector), and then if there is a policy π that is on the wrong side of the hyperplane, we can find it by running a linear optimization over $\mathcal{C}$ in the negative normal vector direction as in Lemma 9.

   If $W \notin \mathcal{C}$, then in a bounded number of iterations (depending on the distance of W from $\mathcal{C}$, and the maximum magnitude $\|\pi\|$) we obtain a separating hyperplane. In passing we also note that if $W \in \mathcal{C}$, the same technique also allows us to explicitly compute an approximate convex combination of policies in Π that yields W. This is done by running the perceptron algorithm as before and stopping after the bound on the number of iterations has been reached. Then we collect all the policies we have found in the run of the perceptron algorithm, and we are guaranteed that W is close in distance to their convex hull. We can then find the closest point in the convex hull of these policies by solving a simple quadratic program.

3. Finally, we consider constraint (5.3). We first rewrite $\eta_{t-1}(W)$ as $\eta_{t-1}(W) = w \cdot W$, where w is a vector defined as $w(x,a) = \frac{1}{t-1}\sum_{(x',a',r,p)\in h_t:\ x'=x,a'=a} \frac{r}{p}$. Thus, $\Delta_{t-1}(Z) = v - w \cdot Z$, where $v = \max_{\pi'} \eta_{t-1}(\pi') = \max_{\pi'} w \cdot \pi'$ which can be computed by using $\mathcal{AMO}$ once.

   Next, using the candidate point W, compute the vector u defined as $u(x,a) = \frac{n_x/t}{W'(x,a)}$, where n_x is the number of times x appears in h_{t-1}, so that $\mathbb{E}_{x\sim h_{t-1}}\left[\sum_a \frac{Z(x,a)}{W'(x,a)}\right] = u \cdot Z$. Now, the problem reduces to finding a policy $Z \in \mathcal{C}$ which violates the constraint

$$u \cdot Z \le \max\{4K, \beta_t (w \cdot Z - v)^2\}.$$

   Define $f(Z) = \max\{4K, \beta_t(w\cdot Z - v)^2\} - u\cdot Z$. Note that f is convex function of Z. Finding a point Z that violates the above constraint is equivalent to solving the following (convex) program:

$$f(Z) \ \le \ 0 \tag{5.4}$$
$$Z \ \in \ \mathcal{C} \tag{5.5}$$

   To do this, we again apply the ellipsoid method. For this, we need a separation oracle for the program. A separation oracle for the constraints (5.5) can be constructed as in step 2 above. For the constraints (5.4), if the candidate solution Z has $f(Z) > 0$, then we can construct a separating hyperplane as in Lemma 10. Thus we can solve the program by the ellipsoid method.

   Suppose that after solving the program, we get a point $Z \in \mathcal{C}$ such that $f(Z) \le 0$, i.e. W violates the constraint (5.3) for Z. Then since constraint (5.3) is convex in W, we can construct a separating hyperplane as in Lemma 10. This completes the description of the separation oracle.

Working out all the details carefully, this gives us the following theorem, proved in Appendix E (in the full version):

Theorem 11. *There is an iterative algorithm with at most $O(t^5K^4\log^2(\frac{tK}{\delta}))$ iterations, each involving one call to $\mathcal{AMO}$ and $O(t^2K^2)$ processing time, that either outputs an explicit distribution P over policies in Π such that W_P satisfies*

$$\forall Z \in \mathcal{C}:$$
$$\mathop{\mathbb{E}}_{x\sim h_{t-1}}\left[\sum_a \frac{Z(x,a)}{W'_P(x,a)}\right] \le \max\{4K, \beta_t \Delta_{t-1}(Z)^2\} + 5\epsilon$$
$$\Delta_{t-1}(W) \ \le \ s + 2\gamma,$$

where $\epsilon = \frac{8\delta}{\mu_t^2}$ and $\gamma = \frac{\delta}{\mu_t}$, or declares correctly that $\mathcal{A}$ is infeasible.

6 THE DELAYED FEEDBACK SETTING

In a delayed feedback setting, we observe rewards with a τ step delay according to:

1. The world presents features x_t.
2. The learning algorithm chooses an action $a_t \in \{1, ..., K\}$.
3. The world presents a reward $r_{t-\tau}$ for the action $a_{t-\tau}$ given the features $x_{t-\tau}$.

We can deal with the delay setting by suitably modifying Algorithm 1 to incorporate the factor τ. This gives us Algorithm 3.

Now we can prove the following theorem, which shows the delay has an additive effect on regret.

Theorem 12. *For all distributions D over K actions and features, for all sets of N policies Π, and all delay*

Algorithm 3 Delayed_PE (Π,δ,K,D_X,τ)

Let $\Pi_0 = \Pi$ and history $h_0 = \emptyset$

Define: $\delta_t \doteq \delta \,/\, 4Nt^2$

Define: $b_t \doteq 2\sqrt{\dfrac{2K\ln(1/\delta_t)}{t}}$

Define: $\mu_t \doteq \min\left\{\dfrac{1}{2K}\,,\ \sqrt{\dfrac{\ln(1/\delta_t)}{2Kt}}\right\}$

For each timestep $t = 1 \ldots T$, observe x_t and do:

1. Let $t' = \max(t-\tau, 1)$.
2. Choose distribution P_t over Π_{t-1} s.t. $\forall\ \pi \in \Pi_{t-1}$:
$$\mathop{\mathbb{E}}_{x\sim D_X}\left[\frac{1}{(1-K\mu_{t'})W_{P_t}(x,\pi(x)) + \mu_{t'}}\right] \le 2K$$
3. $\forall\ a \in A$, Let $W'_t(a) = (1-K\mu_{t'})W_{P_t}(x_t, a) + \mu_{t'}$
4. Choose $a_t \sim W'_t$
5. Observe reward r_t.
6. Let $\Pi_t = \Big\{\pi \in \Pi_{t-1} : \eta_h(\pi) \ge \Big(\max_{\pi'\in\Pi_{t-1}} \eta_h(\pi')\Big) - 2b_{t'}\Big\}$
7. Let $h_t = h_{t-1} \cup (x_t, a_t, r_t, W'_t(a_t))$

factors τ, with probability at least $1-\delta$, the regret of the Delayed Policy Elimination algorithm (Algorithm 3) is at most

$$16\sqrt{2K\ln\frac{4T^2N}{\delta}}\left(\tau + \sqrt{T}\right).$$

Proof. We conduct the same regret analysis as in the proof of Theorem 4. The variance bound is unchanged because that depends purely on the unlabeled data distribution. We therefore only need to replace $\sum_{t-1}^{T}\frac{1}{\sqrt{t}}$ with $\tau + \sum_{t=\tau+1}^{T+\tau}\frac{1}{\sqrt{t-\tau}} = \tau + \sum_{t=1}^{T}\frac{1}{\sqrt{t}}$ in Equation 3.4, implying the given bound on regret. □

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, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002a.

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

Alina Beygelzimer, John Langford, and Pradeep Ravikumar. Error correcting tournaments. In *ALT*, 2009.

Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert E. Schapire. An optimal high probability algorithm for the contextual bandit problem. *CoRR*, abs/1002.4058, 2010.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

David A. Freedman. On tail probabilities for martingales. *Annals of Probability*, 3(1):100–118, 1975.

Sham M. Kakade and Adam Kalai. From batch to transductive online learning. In *NIPS*, 2005.

Adam Tauman Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, 2005.

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

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

Maurice Sion. On general minimax theorems. *Pacific J. Math.*, 8(1):171–176, 1958.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *ICML*, pages 1015–1022, 2010.

A Concentration Inequality

The following is an immediate corollary of Theorem 1 of (Beygelzimer et al., 2010). It can be viewed as a version of Freedman's Inequality (Freedman, 1975). Let $y_1, \ldots, y_T$ be a sequence of real-valued random variables. Let $\mathbb{E}_t$ denote the conditional expectation $\mathbb{E}[\,\cdot \mid y_1, \ldots, y_{t-1}]$ and $\mathbb{V}_t$ conditional variance.

Theorem 13 (Freedman-Style Inequality)**.** *Let $V, R \in \mathbb{R}$ such that $\sum_{t=1}^{T}\mathbb{V}_t[y_t] \le V$, and for all t, $y_t - \mathbb{E}_t[y_t] \le R$. Then for any $\delta > 0$ such that $R \le \sqrt{V/\ln(2/\delta)}$, with probability at least $1-\delta$,*

$$\left|\sum_{t=1}^{T} y_t - \sum_{t=1}^{T}\mathbb{E}_t[y_t]\right| \le 2\sqrt{V\ln(2/\delta)}\ .$$

B Minimax Theorem

The following is a continuous version of Sion's Minimax Theorem (Sion, 1958, Theorem 3.4). It follows immediately from the original result.

Theorem 14. *Let $\mathcal{W}$ and $\mathcal{Z}$ be compact and convex sets, and $f : \mathcal{W}\times\mathcal{Z} \to \mathbb{R}$ a function such that $f(W,Z)$ is convex and continuous in W for all $Z \in \mathcal{Z}$, and concave and continuous in Z for all $W \in \mathcal{W}$. Then*

$$\min_{W\in\mathcal{W}}\max_{Z\in\mathcal{Z}} f(W,Z) = \max_{Z\in\mathcal{Z}}\min_{W\in\mathcal{W}} f(W,Z)\ .$$

C Analysis of the Randomized UCB Algorithm

C.1 Preliminaries

First, we define the following constants.

- $\epsilon \in (0,1)$ is a fixed constant, and
- $\rho \doteq \frac{7500}{\epsilon^3}$ is the factor that appears in the bound from Theorem 7.
- $\theta \doteq (\rho+1)/(1-(1+\epsilon)/2) = \frac{2}{1-\epsilon}\left(1+\frac{7500}{\epsilon^3}\right) \geq 5$ is a constant central to Lemma 19, which bounds the variance of the optimal policy's estimated rewards.

Recall the algorithm-specific quantities

$$C_t \doteq 2\log\left(\frac{Nt}{\delta}\right)$$
$$\mu_t \doteq \min\left\{\frac{1}{2K}, \sqrt{\frac{C_t}{2Kt}}\right\}.$$

It can be checked that μ_t is non-increasing. We define the following time indices:

- t_0 is the first round t in which $\mu_t = \sqrt{C_t/(2Kt)}$. Note that $8K \leq t_0 \leq 8K\log(NK/\delta)$.
- $t_1 := \lceil 16K\log(8KN/\delta)\rceil$ is the round given by Theorem 7 such that, with probability at least $1-\delta$,

$$\mathop{\mathbb{E}}_{x_t\sim D_X}\left[\frac{1}{W_t'(\pi(x_t))}\right] \leq (1+\epsilon)\mathop{\mathbb{E}}_{x\sim h_{t-1}}\left[\frac{1}{W_{P_t,\mu_t}(x,\pi(x))}\right] + \rho K \quad \text{(C.1)}$$

for all $\pi \in \Pi$ and all $t \geq t_1$, where $W_{P,\mu}(x,\cdot)$ is the distribution over A given by

$$W_{P,\mu}(x,a) \doteq (1-K\mu)W_P(x,a) + \mu,$$

and the notation $\mathbb{E}_{x\sim h_{t-1}}$ denotes expectation with respect to the empirical (uniform) distribution over $x_1,\ldots,x_{t-1}$.

The following lemma shows the effect of allowing slack in the optimization constraints.

Lemma 15. *If P satisfies the constraints of the optimization problem* (4.1) *with slack K for each distribution Q over Π,* i.e.*,*

$$\mathop{\mathbb{E}}_{\pi\sim Q}\mathop{\mathbb{E}}_{x\sim h_{t-1}}\left[\frac{1}{(1-K\mu_t)W_P(x,\pi(x))+\mu_t}\right] \leq \max\left\{4K, \frac{(t-1)\Delta_{t-1}(W_Q)^2}{180C_{t-1}}\right\} + K$$

for all Q, then P satisfies

$$\mathop{\mathbb{E}}_{\pi\sim Q}\mathop{\mathbb{E}}_{x\sim h_{t-1}}\left[\frac{1}{(1-K\mu_t)W_P(x,\pi(x))+\mu_t}\right] \leq \max\left\{5K, \frac{(t-1)\Delta_{t-1}(W_Q)^2}{144C_{t-1}}\right\}$$

for all Q.

Proof. Let $b \doteq \max\left\{4K, \frac{(t-1)\Delta_{t-1}(\pi)^2}{180C_{t-1}}\right\}$. Note that $\frac{b}{4} \geq K$. Hence $b+K \leq \frac{5b}{4}$ which gives the stated bound. □

Note that the allowance of slack K is somewhat arbitrary; any $O(K)$ slack is tolerable provided that other constants are adjusted appropriately.

C.2 Deviation Bound for $\eta_t(\pi)$

For any policy $\pi \in \Pi$, define, for $1 \leq t \leq t_0$,

$$\bar{V}_t(\pi) \doteq K,$$

and for $t > t_0$,

$$\bar{V}_t(\pi) \doteq K + \mathop{\mathbb{E}}_{x_t\sim D_X}\left[\frac{1}{W_t'(\pi(x_t))}\right].$$

The $\bar{V}_t(\pi)$ bounds the variances of the terms in $\eta_t(\pi)$.

Lemma 16. *Assume the bound in* (C.1) *holds for all $\pi \in \Pi$ and $t \geq t_1$. For all $\pi \in \Pi$:*

1. *If $t \leq t_1$, then*
$$K \leq \bar{V}_t(\pi) \leq 4K.$$
2. *If $t > t_1$, then*
$$\bar{V}_t(\pi) \leq (1+\epsilon)\mathop{\mathbb{E}}_{x\sim h_{t-1}}\left[\frac{1}{(1-K\mu_t)W_{P_t}(x,\pi(x))+\mu_t}\right] + (\rho+1)K.$$

Proof. For the first claim, note that if $t < t_0$, then $\bar{V}_t(\pi) = K$, and if $t_0 \leq t < t_1$, then

$$\mu_t = \sqrt{\frac{\log(Nt/\delta)}{Kt}} \geq \sqrt{\frac{\log(Nt_0/\delta)}{16K^2\log(8KN/\delta)}} \geq \frac{1}{4K};$$

so $W_t'(a) \geq \mu_t \geq 1/(4K)$.

For the second claim, pick any $t > t_1$, and note that by definition of t_1, for any $\pi \in \Pi$ we have

$$\mathop{\mathbb{E}}_{x_t\sim D_X}\left[\frac{1}{W_t'(\pi(x_t))}\right] \leq (1+\epsilon)\mathop{\mathbb{E}}_{x\sim h_{t-1}}\left[\frac{1}{(1-\mu_t)W_{P_t}(x,\pi(x))+\mu_t}\right] + \rho K.$$

The last inequality above follows from the fact $\mu_t < 1/(3K)$. The stated bound on $\bar{V}_t(\pi)$ now follows from its definition. □

Let

$$\bar{V}_{\max,t}(\pi) \doteq \max\{\bar{V}_\tau(\pi),\ \tau = 1, 2, \ldots, t\}$$

The following lemma gives a deviation bound for $\eta_t(\pi)$ in terms of these quantities.

Lemma 17. *Pick any $\delta \in (0,1)$. With probability at least $1-\delta$, for all pairs $\pi, \pi' \in \Pi$ and $t \geq t_0$, we have*

$$\left|(\eta_t(\pi) - \eta_t(\pi')) - (\eta_D(\pi) - \eta_D(\pi'))\right| \leq 2\sqrt{\frac{(\bar{V}_{\max,t}(\pi) + \bar{V}_{\max,t}(\pi')) \cdot C_t}{t}}. \quad \text{(C.2)}$$

Proof. Fix any $t \geq t_0$ and $\pi, \pi' \in \Pi$. Let $\delta_t := \exp(-C_t)$. Pick any $\tau \leq t$. Let

$$Z_\tau(\pi) \doteq \frac{r_\tau(a_\tau)\mathbb{I}(\pi(x_\tau) = a_\tau)}{W'_\tau(a_\tau)}$$

so $\eta_t(\pi) = t^{-1}\sum_{\tau=1}^t Z_\tau(\pi)$. It is easy to see that

$$\mathop{\mathbb{E}}_{\substack{(x_\tau,\vec{r}_\tau)\sim D,\\ a_\tau \sim W'_\tau}}[Z_\tau(\pi) - Z_\tau(\pi')] = \eta_D(\pi) - \eta_D(\pi')$$

and

$$\begin{aligned}
&\sum_{\tau=1}^t \mathop{\mathbb{E}}_{\substack{(x_\tau,\vec{r}(\tau))\sim D,\\ a_\tau \sim W'_\tau}}\left[(Z_\tau(\pi) - Z_\tau(\pi'))^2\right] \\
&\leq \sum_{\tau=1}^t \mathop{\mathbb{E}}_{x_\tau \sim D_X}\left[\frac{1}{W'_\tau(\pi(x_\tau))} + \frac{1}{W'_\tau(\pi'(x_\tau))}\right] \\
&\leq t \cdot (\bar{V}_{\max,t}(\pi) + \bar{V}_{\max,t}(\pi')).
\end{aligned}$$

Moreover, with probability 1,

$$|Z_\tau(\pi) - Z_\tau(\pi')| \leq \frac{1}{\mu_\tau}.$$

Now, note that since $t \geq t_0$, $\mu_t = \sqrt{\frac{C_t}{2Kt}}$, so that $t = \frac{C_t}{2K\mu_t^2}$. Further, both $\bar{V}_{\max,t}(\pi)$ and $\bar{V}_{\max,t}(\pi')$ are at least K. Using these bounds we get

$$\begin{aligned}
&\sqrt{\frac{1}{\log(1/\delta_t)} \cdot t \cdot (\bar{V}_{\max,t}(\pi) + \bar{V}_{\max,t}(\pi'))} \\
&\geq \sqrt{\frac{1}{C_t} \cdot \frac{C_t}{2K\mu_t^2} \cdot 2K} = \frac{1}{\mu_t} \geq \frac{1}{\mu_\tau},
\end{aligned}$$

for all $\tau \leq t$, since the μ_τ's are non-increasing. Therefore, by Freedman's inequality (Theorem 13), we have

$$\begin{aligned}
&\Pr\left[\left|(\eta_t(\pi) - \eta_t(\pi')) - (\eta_D(\pi) - \eta_D(\pi'))\right|\right. \\
&\left. > 2\sqrt{\frac{(\bar{V}_{\max,t}(\pi) + \bar{V}_{\max,t}(\pi')) \cdot \log(1/\delta_t)}{t}}\right] \leq 2\delta_t.
\end{aligned}$$

The conclusion follows by taking a union bound over $t_0 < t \leq T$ and all pairs $\pi, \pi' \in \Pi$. □

C.3 Variance Analysis

We define the following condition, which will be assumed by most of the subsequent lemmas in this section.

Condition 1. The deviation bound (C.1) holds for all $\pi \in \Pi$ and $t \geq t_1$, and the deviation bound (C.2) holds for all pairs $\pi, \pi' \in \Pi$ and $t \geq t_0$.

The next two lemmas relate the $\bar{V}_t(\pi)$ to the $\Delta_t(\pi)$.

Lemma 18. *Assume Condition 1. For any $t \geq t_1$ and $\pi \in \Pi$, if $\bar{V}_t(\pi) > \theta K$, then*

$$\Delta_{t-1}(\pi) \geq \sqrt{\frac{72\bar{V}_t(\pi)C_{t-1}}{t-1}}.$$

Proof. By Lemma 16, the fact $\bar{V}_t(\pi) > \theta K$ implies that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{x\sim h_{t-1}}\left[\frac{1}{(1-K\mu_t)W_{P_t}(x,\pi(x)) + \mu_t}\right] \\
&\qquad > \frac{1}{1+\epsilon}\left(1 - \frac{\rho+1}{\theta}\right)\bar{V}_t(\pi) \geq \frac{1}{2}\bar{V}_t(\pi).
\end{aligned}$$

Since $\bar{V}_t(\pi) > \theta K \geq 5K$, Lemma 15 implies that in order for P_t to satisfy the optimization constraint in (4.1) corresponding to π (with slack $\leq K$), it must be the case that

$$\begin{aligned}
&\Delta_{t-1}(\pi) \\
&\geq \sqrt{\frac{144C_{t-1}}{t-1} \cdot \mathop{\mathbb{E}}_{x\sim h_{t-1}}\left[\frac{1}{(1-K\mu_t)W_{P_t}(x,\pi(x)) + \mu_t}\right]}.
\end{aligned}$$

Combining with the above, we obtain

$$\Delta_{t-1}(\pi) \geq \sqrt{\frac{72\bar{V}_t(\pi)C_{t-1}}{t-1}}.$$

□

Lemma 19. *Assume Condition 1. For all $t \geq 1$, $\bar{V}_{\max,t}(\pi_{\max}) \leq \theta K$ and $\bar{V}_{\max,t}(\pi_t) \leq \theta K$.*

Proof. By induction on t. The claim for all $t \le t_1$ follows from Lemma 16. So take $t > t_1$, and assume as the (strong) inductive hypothesis that $\bar{V}_{\max,\tau}(\pi_{\max}) \le \theta K$ and $\bar{V}_{\max,\tau}(\pi_\tau) \le \theta K$ for $\tau \in \{1, \dots, t-1\}$. Suppose for sake of contradiction that $\bar{V}_t(\pi_{\max}) > \theta K$. By Lemma 18,

$$\Delta_{t-1}(\pi_{\max}) \ge \sqrt{\frac{72\bar{V}_t(\pi_{\max})C_{t-1}}{t-1}}.$$

However, by the deviation bounds, we have

$$\begin{aligned}
&\Delta_{t-1}(\pi_{\max}) + \Delta_D(\pi_{t-1}) \\
&\le 2\sqrt{\frac{(\bar{V}_{\max,t-1}(\pi_{t-1}) + \bar{V}_{\max,t-1}(\pi_{\max}))C_{t-1}}{t-1}} \\
&\le 2\sqrt{\frac{2\bar{V}_t(\pi_{\max})C_{t-1}}{t-1}} < \sqrt{\frac{72\bar{V}_t(\pi_{\max})C_{t-1}}{t-1}}.
\end{aligned}$$

The second inequality follows from our assumption and the induction hypothesis:

$$\bar{V}_t(\pi_{\max}) > \theta K \ge \bar{V}_{\max,t-1}(\pi_{t-1}), \bar{V}_{\max,t-1}(\pi_{\max}).$$

Since $\Delta_D(\pi_{t-1}) \ge 0$, we have a contradiction, so it must be that $\bar{V}_t(\pi_{\max}) \le \theta K$. This proves that $\bar{V}_{\max,t}(\pi_{\max}) \le \theta K$.

It remains to show that $\bar{V}_{\max,t}(\pi_t) \le \theta K$. So suppose for sake of contradiction that the inequality fails, and let $t_1 < \tau \le t$ be any round for which $\bar{V}_\tau(\pi_t) = \bar{V}_{\max,t}(\pi_t) > \theta K$. By Lemma 18,

$$\Delta_{\tau-1}(\pi_t) \ge \sqrt{\frac{72\bar{V}_\tau(\pi_t)C_{\tau-1}}{\tau-1}}. \tag{C.3}$$

On the other hand,

$$\begin{aligned}
\Delta_{\tau-1}(\pi_t) &\le \Delta_D(\pi_{\tau-1}) + \Delta_{\tau-1}(\pi_t) + \Delta_t(\pi_{\max}) \\
&= \Big(\Delta_D(\pi_{\tau-1}) + \Delta_{\tau-1}(\pi_{\max})\Big) \\
&\quad + \Big(\eta_{\tau-1}(\pi_{\max}) - \eta_{\tau-1}(\pi_t) - \Delta_D(\pi_t)\Big) \\
&\quad + \Big(\Delta_D(\pi_t) + \Delta_t(\pi_{\max})\Big).
\end{aligned}$$

The parenthesized terms can be bounded using the deviation bounds, so we have

$$\begin{aligned}
&\Delta_{\tau-1}(\pi_t) \\
&\le 2\sqrt{\frac{(\bar{V}_{\max,\tau-1}(\pi_{\tau-1}) + \bar{V}_{\max,\tau-1}(\pi_{\max}))C_{\tau-1}}{\tau-1}} \\
&\quad + 2\sqrt{\frac{(\bar{V}_{\max,\tau-1}(\pi_t) + \bar{V}_{\max,\tau-1}(\pi_{\max}))C_{\tau-1}}{\tau-1}} \\
&\quad + 2\sqrt{\frac{(\bar{V}_{\max,t}(\pi_t) + \bar{V}_{\max,t}(\pi_{\max}))C_t}{t}} \\
&\le 2\sqrt{\frac{2\bar{V}_\tau(\pi_t)C_{\tau-1}}{\tau-1}} + 2\sqrt{\frac{2\bar{V}_\tau(\pi_t)C_{\tau-1}}{\tau-1}} \\
&\quad + 2\sqrt{\frac{2\bar{V}_\tau(\pi_t)C_t}{t}} \\
&< \sqrt{\frac{72\bar{V}_\tau(\pi_t)C_{\tau-1}}{\tau-1}}
\end{aligned}$$

where the second inequality follows from the following facts:

1. By induction hypothesis, we have $\bar{V}_{\max,\tau-1}(\pi_{\tau-1}), \bar{V}_{\max,\tau-1}(\pi_{\max}), \bar{V}_{\max,t}(\pi_{\max}) \le \theta K$, and $\bar{V}_\tau(\pi_t) > \theta K$,
2. $\bar{V}_\tau(\pi_t) \ge \bar{V}_{\max,t}(\pi_t)$, and
3. since τ is a round that achieves $\bar{V}_{\max,t}(\pi_t)$, we have $\bar{V}_\tau(\pi_t) \ge \bar{V}_{\tau-1}(\pi_t)$.

This contradicts the inequality in (C.3), so it must be that $\bar{V}_{\max,t}(\pi_t) \le \theta K$. □

Corollary 20. *Under the assumptions of Lemma 19,*

$$\Delta_D(\pi_t) + \Delta_t(\pi_{\max}) \le 2\sqrt{\frac{2\theta K C_t}{t}}$$

for all $t \ge t_0$.

Proof. Immediate from Lemma 19 and the deviation bounds from (C.2). □

The following lemma shows that if a policy π has large $\Delta_\tau(\pi)$ in some round τ, then $\Delta_t(\pi)$ remains large in later rounds $t > \tau$.

Lemma 21. *Assume Condition 1. Pick any $\pi \in \Pi$ and $t \ge t_1$. If $\bar{V}_{\max,t}(\pi) > \theta K$, then*

$$\Delta_t(\pi) > 2\sqrt{\frac{2\bar{V}_{\max,t}(\pi)C_t}{t}}.$$

Proof. Let $\tau \le t$ be any round in which $\bar{V}_\tau(\pi) =$

$\bar{V}_{\max,t}(\pi) > \theta K$. We have

$$\begin{aligned}
\Delta_t(\pi) &\geq \Delta_t(\pi) - \Delta_t(\pi_{\max}) - \Delta_D(\pi_{\tau-1}) \\
&= \Delta_{\tau-1}(\pi) + \Big(\eta_t(\pi_{\max}) - \eta_t(\pi) - \Delta_D(\pi)\Big) \\
&\quad + \Big(\eta_D(\pi_{\tau-1}) - \eta_D(\pi) - \Delta_{\tau-1}(\pi)\Big) \\
&\geq \sqrt{\frac{72\bar{V}_\tau(\pi)C_{\tau-1}}{\tau-1}} \\
&\quad - 2\sqrt{\frac{(\bar{V}_{\max,t}(\pi) + \bar{V}_{\max,t}(\pi_{\max}))C_t}{t}} \\
&\quad - 2\sqrt{\frac{(\bar{V}_{\max,\tau-1}(\pi) + \bar{V}_{\max,\tau-1}(\pi_{\tau-1}))C_{\tau-1}}{\tau-1}} \\
&> \sqrt{\frac{72\bar{V}_{\max,t}(\pi)C_{\tau-1}}{\tau-1}} - 2\sqrt{\frac{2\bar{V}_{\max,t}(\pi)C_t}{t}} \\
&\quad - 2\sqrt{\frac{2\bar{V}_{\max,t}(\pi)C_{\tau-1}}{\tau-1}} \\
&\geq 2\sqrt{\frac{2\bar{V}_{\max,t}(\pi)C_{\tau-1}}{\tau-1}} \geq 2\sqrt{\frac{2\bar{V}_{\max,t}(\pi)C_t}{t}}
\end{aligned}$$

where the second inequality follows from Lemma 18 and the deviation bounds, and the third inequality follows from Lemma 19 and the facts that $\bar{V}_\tau(\pi) = \bar{V}_{\max,t}(\pi) > \theta K \geq \bar{V}_{\max,t}(\pi_{\max}), \bar{V}_{\max,\tau-1}(\pi_{\tau-1})$, and $\bar{V}_{\max,t}(\pi) \geq \bar{V}_{\max,\tau-1}(\pi)$. □

C.4 Regret Analysis

We now bound the value of the optimization problem (4.1), which then leads to our regret bound. The next lemma shows the existence of a feasible solution with a certain structure based on the non-uniform constraints. Recall from Section 5, that solving the optimization problem $\mathcal{A}$, i.e. constraints (5.1, 5.2, 5.3), for the smallest feasible value of s is equivalent to solving the RUCB optimization problem (4.1). Recall that $\beta_t = \frac{t-1}{180C_{t-1}}$.

Lemma 22. *There is a point $W \in \mathbb{R}^{(t-1)K}$ such that*

$$\Delta_{t-1}(W) \;\leq\; 4\sqrt{\frac{K}{\beta_t}}$$

$$W \;\in\; \mathcal{C}$$

$$\forall Z \in \mathcal{C} : \mathop{\mathbb{E}}_{x\sim h_{t-1}}\left[\sum_a \frac{Z(x,a)}{W'(x,a)}\right] \leq \max\{4K, \beta_t\Delta_{t-1}(Z)^2\}$$

In particular, the value of the optimization problem (4.1), OPT_t*, is bounded by* $8\sqrt{\frac{K}{\beta_t}} \leq 110\sqrt{\frac{KC_{t-1}}{t-1}}$.

Proof. Define the sets $\{\mathcal{C}_i : i = 1, 2, \ldots\}$ such that

$$\mathcal{C}_i := \{Z \in \mathcal{C} : 2^{i+1}\kappa \leq \Delta_{t-1}(Z) \leq 2^{i+2}\kappa\},$$

where $\kappa = \sqrt{\frac{K}{\beta_t}}$. Note that since $\Delta_{t-1}(Z)$ is a linear function of Z, each $\mathcal{C}_i$ is a closed, convex, compact set. Also, define $\mathcal{C}_0 = \{Z \in \mathcal{C} : \Delta_{t-1}(Z) \leq 4\kappa\}$. This is also a closed, convex, compact set. Note that $\mathcal{C} = \bigcup_{i=0}^\infty \mathcal{C}_i$.

Let $I = \{i : \mathcal{C}_i \neq \emptyset\}$.For $i \in I \setminus \{0\}$, define $w_i = 4^{-i}$, and let $w_0 = 1 - \sum_{i\in I\setminus\{0\}} w_i$. Note that $w_0 \geq 2/3$.

By Lemma 1, for each $i \in I$, there is a point $W_i \in \mathcal{C}_i$ such that for all $Z \in \mathcal{C}_i$, we have

$$\mathop{\mathbb{E}}_{x\sim h_{t-1}}\left[\sum_a \frac{Z(x,a)}{W_i'(x,a)}\right] \leq 2K.$$

Here we use the fact that $K\mu_t \leq 1/2$ to upper bound $\frac{K}{1-K\mu_t}$ by $2K$. Now consider the point $W = \sum_{i\in I} w_i W_i$. Since $\mathcal{C}$ is convex, $W \in \mathcal{C}$.

Now fix any $i \in I$. For any (x,a), we have $W'(x,a) \geq w_i W_i'(x,a)$, so that for all $Z \in \mathcal{C}_i$, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim h_{t-1}}\left[\sum_a \frac{Z(x,a)}{W'(x,a)}\right] &\leq \frac{1}{w_i}2K \\
&\leq 4^{i+1}K \\
&\leq \max\{4K, \beta_t\Delta_{t-1}(Z)^2\},
\end{aligned}$$

so the constraint for Z is satisfied.

Finally, since for all $i \in I$, we have $w_i \leq 4^{-i}$ and $\Delta_{t-1}(W_i) \leq 2^{i+2}\kappa$, we get

$$\Delta_{t-1}(W) = \sum_{i\in I} w_i\Delta_{t-1}(W_i) \leq \sum_{i=0}^\infty 4^{-i}\cdot 2^{i+2}\kappa \leq 8\kappa.$$

□

The value of the optimization problem (4.1) can be related to the expected instantaneous regret of policy drawn randomly from the distribution P_t.

Lemma 23. *Assume Condition 1. Then*

$$\sum_{\pi\in\Pi} P_t(\pi)\Delta_D(\pi) \leq \Big(220 + 4\sqrt{2\theta}\Big)\cdot\sqrt{\frac{KC_{t-1}}{t-1}} + 2\varepsilon_{\mathrm{opt},t}$$

for all $t > t_1$.

Proof. Fix any $\pi \in \Pi$ and $t > t_1$. By the deviation bounds, we have

$$\begin{aligned}
&\Big(\eta_D(\pi_{t-1}) - \eta_D(\pi)\Big) \\
&\leq \Delta_{t-1}(\pi) + 2\sqrt{\frac{(\bar{V}_{\max,t-1}(\pi) + \bar{V}_{\max,t-1}(\pi_{t-1}))C_{t-1}}{t-1}} \\
&\leq \Delta_{t-1}(\pi) + 2\sqrt{\frac{\big(\bar{V}_{\max,t-1}(\pi) + \theta K\big)\,C_{t-1}}{t-1}},
\end{aligned}$$

by Lemma 19. By Corollary 20 we have

$$\Delta_D(\pi_{t-1}) \le 2\sqrt{\frac{2\theta K C_{t-1}}{t-1}}$$

Thus, we get

$$\begin{aligned}\Delta_D(\pi) &\le \Big(\eta_D(\pi_{t-1}) - \eta_D(\pi)\Big) + \Delta_D(\pi_{t-1}) \\ &\le \Delta_{t-1}(\pi) + 2\sqrt{\frac{\left(\bar{V}_{\max,t-1}(\pi) + \theta K\right) C_{t-1}}{t-1}} \\ &\quad + 2\sqrt{\frac{2\theta K C_{t-1}}{t-1}}.\end{aligned}$$

If $\bar{V}_{\max,t-1}(\pi) \le \theta K$, then we have

$$\Delta_D(\pi) \le \Delta_{t-1}(\pi) + 4\sqrt{\frac{2\theta K C_{t-1}}{t-1}}.$$

Otherwise, Lemma 21 implies that

$$\bar{V}_{\max,t-1}(\pi) \le \frac{(t-1)\cdot\Delta_{t-1}(\pi)^2}{8C_{t-1}},$$

so

$$\begin{aligned}\Delta_D(\pi) &\le \Delta_{t-1}(\pi) + 2\sqrt{\frac{\Delta_{t-1}(\pi)^2}{8} + \frac{\theta K C_{t-1}}{t-1}} \\ &\quad + 2\sqrt{\frac{2\theta K C_{t-1}}{t-1}} \\ &\le 2\Delta_{t-1}(\pi) + 4\sqrt{\frac{2\theta K C_{t-1}}{t-1}}.\end{aligned}$$

Therefore

$$\begin{aligned}&\sum_{\pi\in\Pi} P_t(\pi)\Delta_D(\pi) \\ &\le 2\sum_{\pi\in\Pi} P_t(\pi)\Delta_{t-1}(\pi) + 4\sqrt{\frac{2\theta K C_{t-1}}{t-1}} \\ &\le 2\left(\mathrm{OPT}_t + \varepsilon_{\mathrm{opt},t}\right) + 4\sqrt{\frac{2\theta K C_{t-1}}{t-1}}\end{aligned}$$

where OPT_t is the value of the optimization problem (4.1). The conclusion follows from Lemma 22. □

We can now finally prove the main regret bound for RUCB.

Proof of Theorem 5. The regret through the first t_1 rounds is trivially bounded by t_1. In the event that Condition 1 holds, we have for all $t \ge t_1$,

$$\begin{aligned}\sum_{a\in A} W_t(a) r_t(a) &\ge \sum_{a\in A} (1 - K\mu_t) W_{P_t}(x_t, a) r_t(a) \\ &\ge \sum_{a\in A} W_{P_t}(x_t, a) r_t(a) - K\mu_t \\ &= \sum_{\pi\in\Pi} P_t(\pi) r_t(\pi(x_t)) - K\mu_t,\end{aligned}$$

and therefore

$$\begin{aligned}&\mathop{\mathbb{E}}_{\substack{(x_t,\vec{r}(t))\sim D \\ a_t\sim W'_t}} [r_t(a_t)] \\ &= \mathop{\mathbb{E}}_{(x_t,\vec{r}(t))\sim D}\left[\sum_{a\in A} W'_t(a) r_t(a)\right] \\ &\ge \sum_{\pi\in\Pi} P_t(\pi)\eta_D(\pi) - K\mu_t \\ &\ge \eta_D(\pi_{\max}) - O\left(\sqrt{\frac{KC_{t-1}}{t-1}} + \varepsilon_{\mathrm{opt},t}\right)\end{aligned}$$

where the last inequality follows from Lemma 23. Summing the bound from $t = t_1 + 1, \dots, T$ gives

$$\begin{aligned}&\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\substack{(x_t,\vec{r}(t))\sim D \\ a_t\sim W'_t}} [\eta_D(\pi_{\max}) - r_t(a_t)] \\ &\qquad\qquad \le t_1 + O\left(\sqrt{TK\log\left(NT/\delta\right)}\right).\end{aligned}$$

By Azuma's inequality, the probability that $\sum_{t=1}^{T} r_t(a_t)$ deviates from its mean by more than $O(\sqrt{T\log(1/\delta)})$ is at most δ. Finally, the probability that Condition 1 does not hold is at most 2δ by Lemma 17, Theorem 7, and a union bound. The conclusion follows by a final union bound. □

D Finite Sample Complexity Analysis

In this section we prove Theorem 7.

We first show uniform convergence for a certain class of policy distributions (Lemma 24), and argue that each distribution P is close to some distribution $\widetilde{P}$ from this class, in the sense that $V_{P\pi,t}$ is close to $V_{\widetilde{P},\pi,t}$ and $\widehat{V}_{P,\pi,t}$ is close to $\widehat{V}_{\widetilde{P},\pi,t}$ (Lemma 25). Together, they imply the main uniform convergence result in Theorem 7.

For each positive integer m, let $\mathsf{Sparse}[m]$ be the set of distributions $\widetilde{P}$ over Π that can be written as

$$\widetilde{P}(\pi) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(\pi = \pi_i)$$

(*i.e.*, the average of m delta functions) for some $\pi_1, \dots, \pi_m \in \Pi$. In our analysis, we approximate an arbitrary distribution P over Π by a distribution $\widetilde{P} \in \mathsf{Sparse}[m]$ chosen randomly by independently drawing $\pi_1, \dots, \pi_m \sim P$; we denote this process by $\widetilde{P} \sim P^m$.

Lemma 24. *Fix positive integers $(m_1, m_2, \dots)$. With probability at least $1 - \delta$ over the random samples*

$(x_1, x_2, \dots)$ *from* D_X,

$$V_{\widetilde{P},\pi,t} \le (1+\lambda)\cdot \widehat{V}_{\widetilde{P},\pi,t} + \left(5+\frac{1}{2\lambda}\right)\cdot \frac{(m_t+1)\log N + \log\frac{2t^2}{\delta}}{\mu_t\cdot(t-1)}$$

for all $\lambda > 0$, *all* $t \ge 1$, *all* $\pi \in \Pi$, *and all distributions* $\widetilde{P} \in \mathsf{Sparse}[m_t]$.

Proof. Let

$$Z_{\widetilde{P},\pi,t}(x) \doteq \frac{1}{(1-K\mu_t)W_{\widetilde{P}}(x,\pi(x)) + \mu_t}$$

so $V_{\widetilde{P},\pi,t} = \mathbb{E}_{x\sim D_X}[Z_{\widetilde{P},\pi,t}(x)]$ and $\widehat{V}_{\widetilde{P},\pi,t} = (t-1)^{-1}\sum_{i=1}^{t-1} Z_{\widetilde{P},\pi,t}(x_i)$. Also let

$$\varepsilon_t \doteq \frac{\log(|\mathsf{Sparse}[m_t]|N2t^2/\delta)}{\mu_t\cdot(t-1)} = \frac{((m_t+1)\log N + \log\frac{2t^2}{\delta})}{\mu_t\cdot(t-1)}.$$

We apply Bernstein's inequality and union bounds over $P \in \mathsf{Sparse}[m_t]$, $\pi \in \Pi$, and $t \ge 1$ so that with probability at least $1-\delta$,

$$V_{\widetilde{P},\pi,t} \le \widehat{V}_{\widetilde{P},\pi,t} + \sqrt{2V_{\widetilde{P},\pi,t}\varepsilon_t} + (2/3)\varepsilon_t$$

all $t \ge 1$, all $\pi \in \Pi$, and all distributions $P \in \mathsf{Sparse}[m_t]$. The conclusion follows by solving the quadratic inequality for $V_{\widetilde{P},\pi,t}$ to get

$$V_{\widetilde{P},\pi,t} \le \widehat{V}_{\widetilde{P},\pi,t} + \sqrt{2\widehat{V}_{\widetilde{P},\pi,t}\varepsilon_t} + 5\varepsilon_t$$

and then applying the AM/GM inequality. □

Lemma 25. *Fix any* $\gamma \in [0,1]$, *and any* $x \in X$. *For any distribution* P *over* Π *and any* $\pi \in \Pi$, *if*

$$m \doteq \left\lceil \frac{6}{\gamma^2\mu_t} \right\rceil,$$

then

$$\mathop{\mathbb{E}}_{\widetilde{P}\sim P^m}\left|\frac{1}{(1-K\mu_t)W_{\widetilde{P}}(x,\pi(x))+\mu_t} - \frac{1}{(1-K\mu_t)W_P(x,\pi(x))+\mu_t}\right| \le \frac{\gamma}{(1-K\mu_t)W_P(x,\pi(x))+\mu_t}.$$

This implies that for all distributions P *over* Π *and any* $\pi \in \Pi$, *there exists* $\widetilde{P} \in \mathsf{Sparse}[m]$ *such that for any* $\lambda > 0$,

$$\left(V_{P,\pi,t} - V_{\widetilde{P},\pi,t}\right) + (1+\lambda)\left(\widehat{V}_{\widetilde{P},\pi,t} - \widehat{V}_{P,\pi,t}\right) \le \gamma(V_{P,\pi,t} + (1+\lambda)\widehat{V}_{P,\pi,t}).$$

Proof. We randomly draw $\widetilde{P} \sim P^m$, with $\widetilde{P}(\pi') \doteq m^{-1}\sum_{i=1}^m \mathbb{I}(\pi' = \pi_i)$, and then define

$$z \doteq \sum_{\pi'\in\Pi} P(\pi')\cdot\mathbb{I}(\pi'(x)=\pi(x)) \quad \text{and}$$
$$\hat{z} \doteq \sum_{\pi'\in\Pi} \widetilde{P}(\pi')\cdot\mathbb{I}(\pi'(x)=\pi(x)).$$

We have $z = \mathbb{E}_{\pi'\sim P}[\mathbb{I}(\pi'(x)=\pi(x)]$ and $\hat{z} = m^{-1}\sum_{i=1}^m \mathbb{I}(\pi_i(x)=\pi(x))$. In other words, $\hat{z}$ is the average of m independent Bernoulli random variables, each with mean z. Thus, $\mathbb{E}_{\widetilde{P}\sim P^m}[(\hat{z}-z)^2] = z(1-z)/m$ and $\Pr_{\widetilde{P}\sim P^m}[\hat{z} \le z/2] \le \exp(-mz/8)$ by a Chernoff bound. We have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\widetilde{P}\sim P^m}\left|\frac{1}{(1-K\mu_t)\hat{z}+\mu_t} - \frac{1}{(1-K\mu_t)z+\mu_t}\right| \\
&\le \mathop{\mathbb{E}}_{\widetilde{P}\sim P^m}\frac{(1-K\mu_t)|\hat{z}-z|}{[(1-K\mu_t)\hat{z}+\mu_t][(1-K\mu_t)z+\mu_t]} \\
&\le \mathop{\mathbb{E}}_{\widetilde{P}\sim P^m}\frac{(1-K\mu_t)|\hat{z}-z|\mathbb{I}(\hat{z}\ge 0.5z)}{0.5[(1-K\mu_t)z+\mu_t]^2} \\
&\quad + \mathop{\mathbb{E}}_{\widetilde{P}\sim P^m}\frac{(1-K\mu_t)|\hat{z}-z|\mathbb{I}(\hat{z}\le 0.5z)}{\mu_t[(1-K\mu_t)z+\mu_t]} \\
&\le \frac{(1-K\mu_t)\sqrt{\mathbb{E}_{\widetilde{P}\sim P^m}|\hat{z}-z|^2}}{0.5[(1-K\mu_t)z+\mu_t]^2} \\
&\quad + \frac{(1-K\mu_t)z\Pr_{\widetilde{P}\sim P^m}(\hat{z}\le 0.5z)}{\mu_t[(1-K\mu_t)z+\mu_t]} \\
&\le \frac{(1-K\mu_t)\sqrt{z/m}}{0.5[2\sqrt{(1-K\mu_t)z\mu_t}][(1-K\mu_t)z+\mu_t]} \\
&\quad + \frac{(1-K\mu_t)z\exp(-mz/8)}{\mu_t[(1-K\mu_t)z+\mu_t]} \\
&\le \frac{\gamma\sqrt{1-K\mu_t}\sqrt{z/m}}{\sqrt{z(6/m)}[(1-K\mu_t)z+\mu_t]} \\
&\quad + \frac{(1-K\mu_t)\gamma^2 mz\exp(-mz/8)}{6[(1-K\mu_t)z+\mu_t]},
\end{aligned}$$

where the third inequality follows from Jensen's inequality, and the fourth inequality uses the AM/GM inequality in the denominator of the first term and the previous observations in the numerators. The final expression simplifies to the first desired displayed inequality by observing that $mz\exp(-mz/8) \le 3$ for all $mz \ge 0$ (the maximum is achieved at $mz = 8$). The second displayed inequality follows from the following facts:

$$\mathop{\mathbb{E}}_{\widetilde{P}\sim P^m}|V_{P,\pi,t} - V_{\widetilde{P},\pi,t}| \le \gamma V_{P,\pi,t},$$
$$\mathop{\mathbb{E}}_{\widetilde{P}\sim P^m}(1+\lambda)|\widehat{V}_{P,\pi,t} - \widehat{V}_{\widetilde{P},\pi,t}| \le \gamma(1+\lambda)\widehat{V}_{P,\pi,t}.$$

Both inequalities follow from the first displayed bound of the lemma, by taking expectation with respect to

the true (and empirical) distributions over x. The desired bound follows by adding the above two inequalities, which implies that the bound holds in expectation, and hence the existence of $\widetilde{P}$ for which the bound holds. □

Now, we can prove Theorem 7.

Proof of Theorem 7. Let

$$m_t \doteq \left\lceil \frac{6}{\lambda^2} \cdot \frac{1}{\mu_t} \right\rceil$$

(for some $\lambda \in (0, 1/5)$ to be determined) and condition on the $\geq 1-\delta$ probability event from Lemma 24 that

$$\begin{aligned} &V_{\widetilde{P},\pi,t} - (1+\lambda)\widehat{V}_{\widetilde{P},\pi,t} \\ &\leq K \cdot \left(5 + \frac{1}{2\lambda}\right) \cdot \frac{(m_t+1)\log(N) + \log(2t^2/\delta)}{K\mu_t \cdot (t-1)} \\ &\leq K \cdot 5\left(1 + \frac{1}{\lambda}\right) \cdot \frac{(m_t+1)\log(N) + \log(2t^2/\delta)}{K\mu_t \cdot t} \end{aligned}$$

for all $t \geq 2$, all $\widetilde{P} \in \mathsf{Sparse}[m_t]$, and all $\pi \in \Pi$. Using the definitions of m_t and μ_t, the second term is at most $(40/\lambda^2) \cdot (1+1/\lambda) \cdot K$ for all $t \geq 16K\log(8KN/\delta)$: the key here is that for $t \geq 16K\log(8KN/\delta)$, we have $\mu_t = \sqrt{\log(Nt/\delta)/(Kt)} \leq 1/(2K)$ and therefore

$$\frac{m_t \log(N)}{K\mu_t t} \leq \frac{6}{\lambda^2} \quad \text{and} \quad \frac{\log(N) + \log(2t^2/\delta)}{K\mu_t t} \leq 2.$$

Now fix $t \geq 16K\log(8KN/\delta)$, $\pi \in \Pi$, and a distribution P over Π. Let $\widetilde{P} \in \mathsf{Sparse}[m_t]$ be the distribution guaranteed by Lemma 25 with $\gamma = \lambda$ satisfying

$$V_{P,\pi,t} \leq \frac{V_{\widetilde{P},\pi,t} - (1+\lambda)\widehat{V}_{\widetilde{P},\pi,t} + (1+\lambda)^2\widehat{V}_{P,\pi,t}}{1-\lambda}.$$

Substituting the previous bound for $V_{\widetilde{P},\pi,t} - (1+\lambda)\widehat{V}_{\widetilde{P},\pi,t}$ gives

$$V_{P,\pi,t} \leq \frac{1}{1-\lambda}\left(\frac{40}{\lambda^2}(1+1/\lambda)K + (1+\lambda)^2\widehat{V}_{P,\pi,t}\right).$$

This can be bounded as $(1+\epsilon)\cdot\widehat{V}_{P,\pi,t} + (7500/\epsilon^3)\cdot K$ by setting $\lambda = \epsilon/5$. □

E Details of Oracle-based Algorithm

We show how to (approximately) solve $\mathcal{A}$ using the ellipsoid algorithm with $\mathcal{AMO}$. Fix a time period t. To avoid clutter, (only) in this section we drop the subscript $t-1$ from $\eta_{t-1}(\cdot)$, $\Delta_{t-1}(\cdot)$, and h_{t-1} so that they becomes $\eta(\cdot)$, $\Delta(\cdot)$, and h respectively.

In order to use the ellipsoid algorithm, we need to relax the program a little bit in order to ensure that the feasible region has a non-negligible volume. To do this, we need to obtain some perturbation bounds for the constraints of $\mathcal{A}$. The following lemma gives such bounds. For any $\delta > 0$, we define $\mathcal{C}_\delta$ to be the set of all points within a distance of δ from $\mathcal{C}$.

Lemma 26. *Let $\delta \leq b/4$ be a parameter. Let $U, W \in \mathcal{C}_{2\delta}$ be points such that $\|U - W\| \leq \delta$. Then we have*

$$|\Delta(U) - \Delta(W)| \;\leq\; \gamma \tag{E.1}$$

$$\forall Z \in \mathcal{C}_1 : \quad \left| \mathop{\mathbb{E}}_{x\sim h}\left[\sum_a \frac{Z(x,a)}{U'(x,a)}\right] - \mathop{\mathbb{E}}_{x\sim h}\left[\sum_a \frac{Z(x,a)}{W'(x,a)}\right] \right| \leq \epsilon \tag{E.2}$$

where $\epsilon = \frac{8\delta}{\mu_t^2}$ and $\gamma = \frac{\delta}{\mu_t}$.

Proof. First, we have

$$\begin{aligned} |\eta(U) - \eta(W)| &\leq \frac{1}{t-1}\sum_{(x,a,r,q)\in h} \frac{r}{p}|U(x,a) - W(x,a)| \\ &\leq \frac{\delta}{\mu_t} = \gamma, \end{aligned}$$

which implies (E.1).

Next, for any $Z \in \mathcal{C}_1$, we have

$$\begin{aligned} &\left|\sum_a \frac{Z(x,a)}{U'(x,a)} - \sum_a \frac{Z(x,a)}{W'(x,a)}\right| \\ &\leq \sum_a |Z(x,a)| \frac{|U'(x,a) - W'(x,a)|}{U'(x,a)W'(x,a)} \\ &\leq \frac{8\delta}{\mu_t^2} \;=\; \epsilon. \end{aligned}$$

In the last inequality, we use the Cauchy-Schwarz inequality, and use the following facts (here, $Z(x,\cdot)$ denotes the vector $\langle Z(x,a)\rangle_a$, etc.):

1. $\|Z(x,\cdot)\| \leq 2$ since $Z \in \mathcal{C}_1$,
2. $\|U'(x,\cdot) - W'(x,\cdot)\| \leq \|U(x,\cdot) - W(x,\cdot)\| \leq \delta$, and
3. $U'(x,a) \geq (1-bK)\cdot(-2\delta) + b \geq b/2$, for $\delta \leq b/4$, and similarly $W'(x,a) \geq b/2$.

This implies (E.2). □

We now consider the following relaxed form of $\mathcal{A}$. Here, $\delta \in (0, b/4)$ is a parameter. We want to find

a point $W \in \mathbb{R}^{(t-1)K}$ such that

$$\Delta(W) \;\leq\; s + \gamma \tag{E.3}$$

$$W \;\in\; \mathcal{C}_\delta \tag{E.4}$$

$$\forall Z \in \mathcal{C}_{2\delta}: \quad \mathop{\mathbb{E}}_{x\sim h}\left[\sum_a \frac{Z(x,a)}{W'(x,a)}\right] \;\leq \max\{4K, \beta_t \Delta(Z)^2\} + \epsilon, \tag{E.5}$$

where ϵ and γ are as defined in Lemma 26. Call this relaxed program $\mathcal{A}'$.

We apply the ellipsoid method to $\mathcal{A}'$ rather than $\mathcal{A}$. Recall the requirements of Lemma 8: we need an enclosing ball of bounded radius for the feasible region, and the radius of an enclosed ball in the feasible region. The following lemma gives this.

Lemma 27. *The feasible region for $\mathcal{A}'$ is contained in $B(0, \sqrt{t} + \delta)$, and if $\mathcal{A}$ is feasible, then it contains a ball of radius δ.*

Proof. Note that for any $W \in \mathcal{C}_\delta$, we have $\|W\| \leq \sqrt{t} + \delta$, so the feasible region lies in $B(0, \sqrt{t} + \delta)$.

Next, if $\mathcal{A}$ is feasible, let $W^\star \in \mathcal{C}$ be any feasible solution to $\mathcal{A}$. Consider the ball $B(W^\star, \delta)$. Let U be any point in $B(W^\star, \delta)$. Clearly $U \in \mathcal{C}_\delta$. By Lemma 26, assuming $\delta \leq 1/2$, we have for all $Z \in \mathcal{C}_{2\delta}$,

$$\begin{aligned}\mathop{\mathbb{E}}_{x\sim h}\left[\sum_a \frac{Z(x,a)}{U'(x,a)}\right] &\leq \mathop{\mathbb{E}}_{x\sim h}\left[\sum_a \frac{Z(x,a)}{U'(x,a)}\right] + \epsilon \\ &\leq \max\{4K, \beta_t \Delta(Z)^2\} + \epsilon.\end{aligned}$$

Also

$$\Delta(U) \leq \Delta(W^\star) + \gamma \leq s + \gamma.$$

Thus, U is feasible for $\mathcal{A}'$, and hence the entire ball $B(W^\star, \delta)$ is feasible for $\mathcal{A}'$. □

We now give the construction of a separation oracle for the feasible region of $\mathcal{A}'$ by checking for violations of the constraints. In the following, we use the word "iteration" to indicate one step of either the ellipsoid algorithm or the perceptron algorithm. Each such iteration involves one call to $\mathcal{AMO}$, and additional $O(t^2K^2)$ processing time.

Let $W \in \mathbb{R}^{(t-1)K}$ be a candidate point that we want to check for feasibility for $\mathcal{A}'$. We can check for violation of the constraint (E.3) easily, and since it is a linear constraint in W, it automatically yields a separating hyperplane if it is violated.

The harder constraints are (E.4) and (E.5). Recall that Lemma 9 shows that that $\mathcal{AMO}$ allows us to do linear optimization over $\mathcal{C}$ efficiently. This immediately gives us the following useful corollary:

Corollary 28. *Given a vector $w \in \mathbb{R}^{(t-1)K}$ and $\delta > 0$, we can compute $\arg\max_{Z\in\mathcal{C}_\delta} w \cdot Z$ using one invocation of $\mathcal{AMO}$.*

Proof. This follows directly from the following fact:

$$\arg\max_{Z\in\mathcal{C}_\delta} w \cdot Z \;=\; \frac{\delta}{\|w\|} w + \arg\max_{Z\in\mathcal{C}} w \cdot Z.$$

□

Now we show how to use $\mathcal{AMO}$ to check for constraint (E.4):

Lemma 29. *Suppose we are given a point W. Then in $O(\frac{t}{\delta^2})$ iterations, if $W \notin \mathcal{C}_{2\delta}$, we can construct a hyperplane separating W from $\mathcal{C}_\delta$. Otherwise, we declare correctly that $W \in \mathcal{C}_{2\delta}$. In the latter case, we can find an explicit distribution P over policies in Π such that W_P satisfies $\|W_P - W\| \leq 2\delta$.*

Proof. We run the perceptron algorithm with the origin at W and all points in $\mathcal{C}_\delta$ being positive examples. The goal of the perceptron algorithm then is to find a hyperplane going through W that puts all of $\mathcal{C}_\delta$ (strictly) on one side. In each iteration of the perceptron algorithm, we have a weight vector w that is the normal to a candidate hyperplane, and we need to find a point $Z \in \mathcal{C}_\delta$ such that $w \cdot (Z - W) \leq 0$ (note that we have shifted the origin to W). To do this, we use $\mathcal{AMO}$ as in Lemma 9 to find $Z^\star = \arg\max_{Z\in\mathcal{C}_\delta} -w \cdot Z$. If $w \cdot (Z^\star - W) \leq 0$, we use $Z^\star$ to update w using the perceptron update rule, $w \leftarrow w + (Z^\star - W)$. Otherwise, we have $w \cdot (Z - W) > 0$ for all $W \in \mathcal{C}_\delta$, and hence we have found our separating hyperplane.

Now suppose that $W \notin \mathcal{C}_{2\delta}$, i.e. the distance of W from $\mathcal{C}_\delta$ is more than δ. Since $\|Z - W\| \leq 2\sqrt{t} + 3\delta = O(\sqrt{t})$ for all $W \in \mathcal{C}_\delta$ (assuming $\delta = O(\sqrt{t})$), the perceptron convergence guarantee implies that in $O(\frac{t}{\delta^2})$ iterations we find a separating hyperplane.

If in $k = O(\frac{t}{\delta^2})$ iterations we haven't found a separating hyperplane, then $W \in \mathcal{C}_{2\delta}$. In fact the perceptron algorithm gives a stronger guarantee: if the k policies found in the run of the perceptron algorithm are $\pi_1, \pi_2, \ldots, \pi_k \in \Pi$, then W is within a distance of 2δ from their convex hull, $\mathcal{C}' = \text{conv}(\pi_1, \pi_2, \ldots, \pi_k)$. This is because a run of the perceptron algorithm on $\mathcal{C}'_{2\delta}$ would be identical to that on $\mathcal{C}_{2\delta}$ for k steps. We can then compute the explicit distribution over policies P by computing the Euclidean projection of W on $\mathcal{C}'$ in poly(k) time using a convex quadratic program:

$$\begin{aligned}\min \;\; & \|W - \textstyle\sum_{i=1}^k P_i \pi_i\|^2 \\ \sum_i P_i \;&=\; 1 \\ \forall i: \; P_i \;&\geq\; 0\end{aligned}$$

Solving this quadratic program, we get a distribution P over the policies $\{\pi_1, \pi_2, \ldots, \pi_k\}$ such that $\|W_P - W\| \leq 2\delta$. □

Finally, we show how to check constraint (E.5):

Lemma 30. *Suppose we are given a point W. In $O(\frac{t^3K^2}{\delta^2} \cdot \log(\frac{t}{\delta}))$ iterations, we can either find a point $Z \in \mathcal{C}_{2\delta}$ such that*

$$\mathop{\mathbb{E}}_{x\sim h}\left[\sum_a \frac{Z(x,a)}{W'(x,a)}\right] \geq \max\{4K, \beta_t\Delta(Z)^2\} + 2\epsilon,$$

or else we conclude correctly that for all $Z \in \mathcal{C}$, we have

$$\mathop{\mathbb{E}}_{x\sim h}\left[\sum_a \frac{Z(x,a)}{W'(x,a)}\right] \leq \max\{4K, \beta_t\Delta(Z)^2\} + 3\epsilon.$$

Proof. We first rewrite $\eta(W)$ as $\eta(W) = w \cdot \pi$, where w is a vector defined as

$$w(x,a) = \frac{1}{t-1} \sum_{(x',a',r,p)\in h:\ x'=x, a'=a} \frac{r}{p}.$$

Thus, $\Delta(Z) = v - w \cdot Z$, where $v = \max_{\pi'} \eta(\pi') = \max_{\pi'} w \cdot \pi'$ which can be computed by using $\mathcal{AMO}$ once.

Next, using the candidate point W, compute the vector u defined as $u(x,a) = \frac{n_x/t}{W'(x,a)}$, where n_x is the number of times x appears in h, so that $\mathbb{E}_{x\sim h}\left[\sum_a \frac{Z(x,a)}{W'(x,a)}\right] = u \cdot Z$. Now, the problem reduces to finding a point $R \in \mathcal{C}$ which violates the constraint

$$u \cdot Z \leq \max\{4K, \beta_t(w \cdot Z - v)^2\} + 3\epsilon.$$

Define

$$f(Z) = \max\{4K, \beta_t(w \cdot Z - v)^2\} + 3\epsilon - u \cdot Z.$$

Note that f is convex function of Z. Checking for violation of the above constraint is equivalent to solving the following (convex) program:

$$f(Z) \ \leq\ 0 \tag{E.6}$$
$$Z \ \in\ \mathcal{C} \tag{E.7}$$

To do this, we again apply the ellipsoid method, but on the relaxed program

$$f(Z) \ \leq\ \epsilon \tag{E.8}$$
$$Z \ \in\ \mathcal{C}_\delta \tag{E.9}$$

To run the ellipsoid algorithm, we need a separation oracle for the program. Given a candidate solution Z, we run the algorithm of Lemma 29, and if $Z \notin \mathcal{C}_{2\delta}$, we construct a hyperplane separating Z from $\mathcal{C}_\delta$.

Now suppose we conclude that $Z \in \mathcal{C}_{2\delta}$. Then we construct a separation oracle for (E.6) as follows. If $f(Z) > \epsilon$, then since f is a convex function of Z, we can construct a separating hyperplane as in Lemma 10.

Now we can run the ellipsoid algorithm with the starting ellipsoid being $B(0, \sqrt{t})$. If there is a point $Z^\star \in \mathcal{C}$ such that $f(Z^\star) \leq 0$, then consider the ball $B(Z^\star, \frac{4\delta}{5\sqrt{tK}\beta_t})$. For any $Y \in B(Z^\star, \frac{4\delta}{5\sqrt{tK}\beta_t})$, we have

$$|(u \cdot Z^\star) - (u \cdot Y)| \leq \|u\|\|Z^\star - Y\| \leq \frac{\epsilon}{2}$$

since $\|u\| \leq \frac{\sqrt{K}}{\mu_t}$. Also,

$$\begin{aligned}
&\beta_t|(w \cdot Z^\star - v)^2 - (w \cdot Y - v)^2| \\
&= \beta_t|(w \cdot Z^\star - w \cdot Y)(w \cdot Z^\star + w \cdot Y - 2v)| \\
&\leq \beta_t\|w\|\|Z^\star - Y\|(\|w\|(\|Z^\star\| + \|Y\|) + 2|v|) \leq \frac{\epsilon}{2},
\end{aligned}$$

since $\|w\| \leq \frac{1}{\mu_t}$, $\|Z^\star\| \leq \sqrt{t}$, $\|Y\| \leq \sqrt{t} + \delta \leq 2\sqrt{t}$, and $|v| \leq \|w\| \cdot \sqrt{t} \leq \frac{\sqrt{t}}{\mu_t}$.

Thus, $f(Y) \leq f(Z^\star) + \epsilon \leq \epsilon$, so the entire ball $B(Z^\star, \frac{4\delta}{5\sqrt{tK}\beta_t})$ is feasible for the relaxed program.

By Lemma 8, in $O(t^2K^2 \cdot \log(\frac{tK}{\delta}))$ iterations of the ellipsoid algorithm, we obtain one of the following:

1. we either find a point $Z \in \mathcal{C}_{2\delta}$ such that $f(Z) \leq \epsilon$, i.e.

$$\mathop{\mathbb{E}}_{x\sim h}\left[\sum_a \frac{Z(x,a)}{W'(x,a)}\right] \ \geq\ \max\{4K, \beta_t\Delta(Z)^2\} + 2\epsilon,$$

2. or else we conclude that the original convex program (E.6,E.7) is infeasible, i.e. for all $Z \in \mathcal{C}$, we have

$$\mathop{\mathbb{E}}_{x\sim h}\left[\sum_a \frac{Z(x,a)}{W'(x,a)}\right] \ \leq\ \max\{4K, \beta_t\Delta(Z)^2\} + 3\epsilon.$$

The total number of invocations of iterations is bounded by $O(t^2K^2 \cdot \log(\frac{tK}{\delta})) \cdot O(\frac{t}{\delta^2}) = O(\frac{t^3K^2}{\delta^2} \cdot \log(\frac{tK}{\delta}))$. □

Lemma 31. *Suppose we are given a point $Z \in \mathcal{C}_{2\delta}$ such that*

$$\mathop{\mathbb{E}}_{x\sim h}\left[\sum_a \frac{Z(x,a)}{W'(x,a)}\right] \ \geq\ \max\{4K, \beta_t\Delta(Z)^2\} + 2\epsilon.$$

Then we can construct a hyperplane separating W from all feasible points for $\mathcal{A}'$.

Proof. For notational convenience, define the function

$$f_Z(W) := \mathop{\mathbb{E}}_{x\sim h}\left[\sum_a \frac{Z(x,a)}{W'(x,a)}\right] - \max\{4K, \beta_t \Delta(Z)^2\} - 2\epsilon.$$

Note that it is a convex function of W. Note that for any point U that is feasible for $\mathcal{A}'$, we have $f_Z(U) \leq -\epsilon$, whereas $f_Z(W) \geq 0$. Thus, by Lemma 10, we can construct the desired separating hyperplane. □

We can finally prove Theorem 11:

Proof. **[Theorem 11.]** We run the ellipsoid algorithm starting with the ball $B(0, \sqrt{t} + \delta)$. At each point, we are given a candidate solution W for program $\mathcal{A}'$. We check for violation of constraint (E.3) first. If it is violated, the constraint, being linear, gives us a separating hyperplane. Else, we use Lemma 29 to check for violation of constraint (E.4). If $W \notin \mathcal{C}_{2\delta}$, then we can construct a separating hyperplane. Else, we use Lemmas 30 and 31 to check for violation of constraint (E.5). If there is a $Z \in \mathcal{C}$ such that $\mathbb{E}_{x\sim h}\left[\sum_a \frac{Z(x,a)}{W'(x,a)}\right] \geq \max\{4K, \beta_t \Delta(Z)^2\} + 3\epsilon$, then we can find a separating hyperplane. Else, we conclude that the current point W satisfies the following constraints:

$$\begin{aligned} \Delta(W) &\leq s + \gamma \\ \forall Z \in \mathcal{C} : \mathop{\mathbb{E}}_{x\sim h}\left[\sum_a \frac{Z(x,a)}{W'(x,a)}\right] &\leq \max\{4K, \beta_t \Delta(Z)^2\} + 3\epsilon \\ W &\in \mathcal{C}_{2\delta} \end{aligned}$$

We can then use the perceptron-based algorithm of Lemma 29 to "round" W to an explicit distribution P over policies in Π such that W_P satisfies $\|W_P - W\| \leq 2\delta$. Then Lemma 26 implies the stated bounds for W_P.

By Lemma 8, in $O(t^2K^2\log(\frac{t}{\delta}))$ iterations of the ellipsoid algorithm, we find the point W satisfying the constraints given above, or declare correctly that $\mathcal{A}$ is infeasible. In the worst case, we might have to run the algorithm of Lemma 30 in every iteration, leading to an upper bound of $O(t^2K^2\log(\frac{t}{\delta})) \times O(\frac{t^3K^2}{\delta^2} \cdot \log(\frac{tK}{\delta})) = O(t^5K^4\log^2(\frac{tK}{\delta}))$ on the number of iterations. □