Reinforcement Learning and Bandits **Spring 2019**

Exploration in Contextual Bandits

Instructors: Alekh Agarwal, Sham Kakade *Lecture 8-9*

In the last lecture we studied a contextual bandit setting where the learning algorithm has no influence over the data collection strategy. In this lecture, we will study a different setting where the learning algorithm actively engages in an *explore-exploit* tradeoff, and controls the choice of actions in response to the contexts it observes. We will focus attention on the i.i.d. setting, that is the contexts and rewards are drawn from a fixed distribution. Statistically, our goal is to match the optimal regret guarantee of $\widetilde{\mathcal{O}}(\sqrt{KT \ln |\Pi|})$ in the worst case.[1] Computationally, our goal is to develop algorithms which can be efficiently implemented.

Continuing from last lecture, we will assume access to a cost-sensitive classification algorithm over policies in Π. We formalize this assumption through the notion of an *ArgMax Oracle* ($\mathcal{AMO}$) below.

Definition 1 (ArgMax Oracle ($\mathcal{AMO}$))**.** Given a policy class Π, an ArgMax Oracle ($\mathcal{AMO}$) for short is an algorithm which takes in an arbitrary dataset $\{(x_s, r_s)\}_{s=1,2,\ldots,t}$ with $x_s \in \mathcal{X}$ and $r_s \in [0,1]^K$, and computes

$$\arg\max_{\pi \in \Pi} \sum_{s=1}^{t} r_s(\pi(x_s)).$$

In words, an $\mathcal{AMO}$ is a reward maximization or equivalently cost minimization algorithm for the class Π.

Remark: Note that the $\mathcal{AMO}$ itself makes no assumption on the contexts and rewards coming from a distribution, and this will be crucial to our algorithms. It is assumed to return the right answer for any dataset, even adversarially generated.

Given such an oracle, we can clearly implement the τ-GREEDY algorithm using just one call to the $\mathcal{AMO}$ for computing the GREEDY policy after τ rounds. However, this gives an algorithm with a suboptimal regret bound scaling as $T^{2/3}$, instead of the optimal $\sqrt{T}$ dependence that we seek.

In this lecture, we will work towards attanining statistical optimality without sacrificing computational feasibility. To build intuition though, we will start from an algorithm which is computationally inefficient. We will then see a sophisticated variant of it which can indeed be implemented efficiently given an $\mathcal{AMO}$.

Before describing the algorithms, we need a few additional pieces of notation. Recall that we use $V(\pi)$ to denote the expected reward of a policy. After the contextual bandit algorithm has run for t rounds, it has an empirical dataset of the form $(x_s, a_s, p_s, r_s)_{s=1}^t$, where p_s is the probability with which the action a_s was chosen at round s. Given such a dataset, we can also use the off-policy techniques from the previous lecture to define an empirical estimate for the reward of any policy π. Specifically, we will use the `IPS` estimator in this lecture for simplicity even though any other estimator can be alternatively used. With this in mind, we define the empirical estimate of π's value using t samples as

$$\widehat{V}_t(\pi) = \frac{1}{t} \sum_{s=1}^{t} r_s \frac{\mathbf{1}(a_s = \pi(x_s))}{p_s}. \tag{1}$$

With these expected and empirical values of a policy π, we also define corresponding shorthands for regret as:

[1] Here, and throughout we use the $\widetilde{\mathcal{O}}$ notation to suppress polylogarithmic factors in K, T and $\ln(|\Pi|/\delta)$.

$$\mathrm{Reg}(\pi) = V(\pi^\star) - V(\pi) = \max_{\pi' \in \Pi} V(\pi') - V(\pi)$$
$$\widehat{\mathrm{Reg}}_t(\pi) = \widehat{V}_t(\widehat{\pi}_t) - \widehat{V}_t(\pi) = \max_{\pi' \in \Pi} \widehat{V}_t(\pi') - \widehat{V}_t(\pi). \tag{2}$$

Note that while $\widehat{V}_t(\pi)$ is an unbiased estimator for $V(\pi)$ under the conditions we discussed in previous lecture, $\widehat{\mathrm{Reg}}_t(\pi)$ is typically a biased estimate for $\mathrm{Reg}(\pi)$ since we base it on the empirical best policy $\widehat{\pi}_t$ instead of $\pi^\star$. However, this bias is usually typically small and can be bounded.

We will use $Q \in \Delta(\Pi)$ to denote distributions over Π, that is $Q(\pi) \geq 0$, $\sum_{\pi \in \Pi} Q(\pi) = 1$. Given a context x, we will also abuse notation and use $Q(a \mid x)$ to be the induced distribution over actions, given the context. That is,

$$Q(a|x) = \sum_{\pi \in \Pi \,:\, \pi(x) = a} Q(\pi). \tag{3}$$

Given a desired minimum probability for each action, we also use the shorthand $Q^\gamma(a|x)$ to be the mixture of Q with the uniform distribution over actions as

$$Q^\gamma(a|x) = (1 - K\gamma) \sum_{\pi \in \Pi \,:\, \pi(x) = a} Q(\pi) + \gamma. \tag{4}$$

Note that we subtract $K\gamma$ instead of just γ, since the uniform distribution adds a probability γ over each of K actions.

1 Policy Elimination

Notice that the EXP4 algorithm, which gives the optimal regret guarantee, does so by maintaining and updating a distribution over policies in Π, with larger weight being placed on policies which obtain a high reward. We will now see an algorithm which uses a similar intuition in the i.i.d. setting.

1.1 The Algorithm

The algorithm, called POLICYELIMINATION [Dudík et al., 2011], is an iterative elimination based algorithm. At each iteration, the algorithm evaluates every surviving policy according to the `IPS` estimator on the data collected so far. Policies which have a low empirical regret according to these estimates are retained, while the rest are eliminated. The algorithm then constructs a probability distribution for exploration at the next round over the surviving policies.

The main design question in the algorithm is how to construct a distribution over the policies, and for this we derive guidance from the results on off-policy evaluation. Specifically, we would like that the set of policies retained based on low-empirical regret resembles the set of policies with low expected regret. This would be true if we have good estimates of the reward for each policy. Since the quality of our off-policiy estimates depends on the distribution of propensity scores, the algorithm constructs a distribution for exploration that results in well-behaved propensity scores on average. This intution is formalized in Algorithm 1.

The two main steps in the algorithm are those of finding a distribution respecting the constraints (5), and that of eliminating bad policies in the last step. The constraint (5) corresponding to π captures the variance in evaluating the reward of the policy π, if we were to collect samples using Q_t. The variance depends, based on last lecture, inversely on the probability $Q^\gamma(\pi(x)|x)$ for the context x. Taking expectation over this quantity results in the constraints of Algorithm 1.

Algorithm 1 POLICYELIMINATION algorithm for i.i.d. contextual bandits

Require: Failure probability δ. number of rounds T.
Initialize $\Pi_0 = \Pi$.
Define $\epsilon_t = 2\sqrt{\frac{2K\ln(NT/\delta)}{t}}$ and $\gamma = \min\left\{\frac{1}{2K}, \sqrt{\frac{\ln(NT\delta)}{2KT}}\right\}$.
for $t = 1, 2, \ldots, T$ **do**
Observe x_t and choose a distribution Q_t over Π_{t-1} such that

$$\mathbb{E}_{x\sim D}\left[\frac{1}{Q^\gamma(\pi(x)|x)}\right] \leq 2K, \qquad \forall \pi \in \Pi_{t-1}. \tag{5}$$

Choose action $a_t \sim Q_t^\gamma(a|x_t)$.
Observe reward r_t.
Update $\Pi_t = \{\pi \in \Pi_{t-1} \ : \ \widehat{\mathrm{Reg}}_t(\pi) \leq 2\epsilon_t\}$.
end for

Remark: Note that Algorithm 1 assumes the knowledge of the distribution D over contexts (that is, the marginal distribution over x, without the rewardS) as well as the number of rounds T. Both of these are avoidable, but assuming these allows us to present simpler proofs without losing any of the key ideas. We refer interested readers to Dudík et al. [2011] for the formal arguments in the more general setting.

Statistically, there are two properties of the algorithm which are important to understand. One is that the distribution Q_t which can satsify the constraint 5 must exist for the algorithm to work. Secondly, we would like to certify that the algorithm incurs a low regret. For the regret, it seems sufficient to argue that our empirical estimates $\widehat{V}_t(\pi)$ are close to $V(\pi)$ for all $\pi \in \Pi_t$ and for all t. In the previous lectures, we saw arguments for doing this both in an off-policy setting as well as for the τ-GREEDY algorithm using Hoeffding's or Bernstein's inequality. Specifically, consider the random variable

$$Y_t(\pi) = r_t \frac{\mathbf{1}(a_t = \pi(x_t))}{p_t} - V(\pi).$$

Then it is easily seen that

$$\widehat{V}_t(\pi) - V(\pi) = \frac{1}{t}\sum_{s=1}^{t} Y_s(\pi). \tag{6}$$

In previous lectures, we argued that $Y_t(\pi)$ is a sequence of i.i.d. and mean-zero random variables, so that their average is close to zero with high probability.

However, we cannot continue to use the same idea here. In both the off-policy and τ-GREEDY settings, the action distribution is *non-adaptive*. That is, (x_t, a_t, p_t, r_t) as a tuple form an i.i.d. collection of random variables. In the off-policy setting, we use a fixed policy μ, while our exploration is uniform in τ-GREEDY. In the POLICYELIMINATION algorithm, on the other hand, the probability $p_t = Q_t^\gamma(a_t|x_t)$ is heavily dependent on the previous actions, contexts and rewards. Consequently, $Y_t(\pi)$ are not i.i.d. anymore and our earlier bounds from Hoeffding's or Bernstein's inequality do not apply. In order to analyze such adaptive algorithms, we recall the notion of *martingales* from the first lecture.

1.2 Digression: Martingale concentration

Definition 2 (Martingale)**.** Suppose $Z_1, \ldots, Z_n$ are mean zero random variables, which additionally satisfy $\mathbb{E}[Z_i|Z_1, \ldots, Z_{i-1}] = 0$ for all $i = 2, 3, \ldots, n$, then the sequence Z_i is called a martingale difference sequence.[2]

[2]This definition is only a special case of the general formulation of martingales. This particular form of a martingale is often called a Doob martingale.

Martingales are useful objects in probability theory as they satisfy many inequalities analogous to i.i.d. random variables. It turns out they are also very relevant in the study of adaptive algorithms like POLICYELIMINATION. Specifically, we can show that the sequence $Y_t(\pi)$ is a martingale difference sequence. Letting $Z_t = Y_t(\pi)$, we notice that conditioning on the previous Z_i is analogous to conditioning on the previous samples (x_i, a_i, p_i, r_i), since they describe all the randomness contained in $Y_i(\pi)$. For brevity, we introduce the following two shorthands given any random variable Y:

$$\mathbb{E}_t[Y] = \mathbb{E}[Y|(x_i, a_i, p_i, r_i)_{i=1}^{t-1}] \quad \text{and} \quad \mathrm{Var}_t[Y] = \mathrm{Var}[Y|(x_i, a_i, p_i, r_i)_{i=1}^{t-1}].$$

Given this notation, checking that $Y_t(\pi)$ form a martigale difference sequence is the same as verifying $\mathbb{E}_t[Y_t(\pi)] = 0$. This is done in our next result.

Lemma 3. *The sequence $Y_t(\pi)$ is a martingale difference sequence.*

Proof: As mentioned above, it suffices to verify that $\mathbb{E}_t[Y_t(\pi)] = 0$. Notice that

$$\begin{aligned}
\mathbb{E}_t\left[r_t \frac{\mathbf{1}(a_t = \pi(x_t))}{p_t}\right] &= \mathbb{E}_t\left[\mathbb{E}\left[r_t \frac{\mathbf{1}(a_t = \pi(x_t))}{p_t}|x_t,\right]\right] \\
&\overset{(a)}{=} \mathbb{E}_t\left[\mathbb{E}\left[\widehat{v}_{\texttt{IPS}}(x_t, \pi(x_t))|x_t, Q_t^\gamma(x_t)\right]\right] \\
&\overset{(b)}{=} \mathbb{E}_t\left[\mathbb{E}\left[v(x_t, \pi(x_t))|x_t, Q_t^\gamma(x_t)\right]\right] \\
&\overset{(c)}{=} V(\pi).
\end{aligned}$$

Here, we have used the defintion of $\widehat{v}_{\texttt{IPS}}$ in equality (a), and we recall that $p_t = Q_t^\gamma(a_t|x_t)$. Then step (b) follows by Lemma 7.3 in the previous lecture, since the expectation is only over the random choice of a_t according to its distribution, and the random choice of rewards from their conditional distribution given x_t and $\pi(x_t)$. In particular, observe that a_t is conditionally independent of the past given x_t and the distribution $Q_t^\gamma(x_t)$ over actions. Similarly r_t is conditionally independent of the past given x_t and a_t, whence the equality (b) follows. Finally, equality (c) uses the fact that x_t is independent of the past and drawn from D, so that the expectation is simply the value of π. ■

Having shown that $Y_t(\pi)$ form a martingale difference sequence, we next describe a Bernstein-like inequality which they follow. This inequality will be crucial in showing that our estimates $\widehat{V}_t(\pi)$ and $V(\pi)$ are close.

Lemma 4 (Freedman-style inequality)**.** *Let $Z_1, \ldots, Z_n$ be a martingale difference sequence with $Z_i \leq R$ for all i. Let $V_n = \sum_{i=1}^n \mathrm{Var}_n[Z_i]$. For any $\delta \in (0,1)$ and any $\lambda \in [0, 1/R]$, with probability at least $1-\delta$*

$$\sum_{i=1}^n Z_i \leq (e-2)\lambda V_n + \frac{\ln(1/\delta)}{\lambda}.$$

Remark: The inequality looks quite different from Bernstein's at the first glance. However, this really is just a more general form convenient for some of our proofs, and of course applies to martingales. We encourage the reader to check that using $\lambda = \min(\sqrt{V_n/\ln(1/\delta)}, 1/R)$ yields a Bernstein-like bound for martingales.

With this digression, we now have all the tools for a statistical analysis of Algorithm 1.

1.3 Regret analysis

We begin by showing that the distribution Q_t posited in Equation 5 does exist.

Theorem 5 (Feasibility)**.** *Let Π be any finite policy class. For any $\gamma \in (0, 1/K]$ and all joint distributions D over (x, r), we have*

$$\min_{Q \in \Delta(\Pi)} \max_{\pi \in \Pi} \mathbb{E}_{x \sim D} \frac{1}{Q^\gamma(\pi(x)|x)} \leq \frac{K}{1 - K\gamma}.$$

Remark: Applying the theorem to the policy set Π_{t-1} and noting that $\gamma \leq 1/2K$ in Algorithm 1, we observe that there exists a distribution $Q_t \in \Delta(\Pi_{t-1})$ such that the maximum LHS of constraints (5) is at most $K/1 - K\gamma \leq 2K$ using $\gamma \leq 1/2K$.

Proof: The result is a consequence of a minimax theorem. To begin, we verify some regularity conditions on the various objects in our problem. Note that $\Delta(\Pi)$ is the convex hull of a finite set. We further note that

$$\begin{aligned}
\max_{\pi \in \Pi} \mathbb{E}_{x \sim D} \frac{1}{Q^\gamma(\pi(x)|x)} &= \max_{P \in \Delta(\Pi)} \mathbb{E}_{x \sim D} \left[\sum_{\pi \in \Pi} P(\pi) \frac{1}{Q^\gamma(\pi(x)|x)} \right] \\
&= \max_{P \in \Delta(\Pi)} \mathbb{E}_{x \sim D} \left[\sum_{a=1}^{K} \sum_{\pi \in \Pi \,:\, \pi(x) = a} P(\pi) \frac{1}{Q^\gamma(a|x)} \right] \\
&= \max_{P \in \Delta(\Pi)} \mathbb{E}_{x \sim D} \left[\sum_{a=1}^{K} P(a|x) \frac{1}{Q^\gamma(a|x)} \right] \\
&= \max_{P \in \Delta(\Pi)} \mathbb{E}_{x \sim D} \mathbb{E}_{a \sim P(\cdot|x)} \frac{1}{Q^\gamma(a|x)}.
\end{aligned}$$

Here the first equality is clearly an upper bound since we enlarged the maximization to a clearly larger set $\Delta(\Pi)$, which in particular includes each policy π by choosing a distribution which puts its entire mass on π. However, the inequality is really an equality because the objective of maximization is linear in P. So the maximization over all distributions P simply returns a point mass over the policy π for which the objective is that largest.

This allows us to instead consider the problem

$$\min_{Q \in \Delta(\Pi)} \max_{P \in \Delta(\Pi)} \mathbb{E}_{x \sim D} \mathbb{E}_{a \sim P(\cdot|x)} \frac{1}{Q^\gamma(a|x)}.$$

Both P and Q are chosen from the convex hull of a finite set here. Furthermore, the objective is linear (and hence concave) in P. Furthermore, it is easily seen to be convex in Q (from the convexity of $f(x) = 1/x$). Under these conditions, we can invoke Sion's minimax theorem [Sion, 1958], which allows us to swap the order of maximization and minimization. That is,

$$\min_{Q \in \Delta(\Pi)} \max_{P \in \Delta(\Pi)} \mathbb{E}_{x \sim D} \mathbb{E}_{a \sim P(\cdot|x)} \frac{1}{Q^\gamma(a|x)} = \max_{P \in \Delta(\Pi)} \min_{Q \in \Delta(\Pi)} \mathbb{E}_{x \sim D} \mathbb{E}_{a \sim P(\cdot|x)} \frac{1}{Q^\gamma(a|x)}.$$

We now proceed to upper bound the RHS of this equality. Since we are taking a minimum over Q, using any other distribution gives a valid upper bound, and in particular we choose $Q = P$. This yields

$$\begin{aligned}
\max_{P\in\Delta(\Pi)}\min_{Q\in\Delta(\Pi)}\mathbb{E}_{x\sim D}\mathbb{E}_{a\sim P(\cdot|x)}\frac{1}{Q^\gamma(a|x)} &\le \max_{P\in\Delta(\Pi)}\mathbb{E}_{x\sim D}\mathbb{E}_{a\sim P(\cdot|x)}\frac{1}{P^\gamma(a|x)}\\
&= \max_{P\in\Delta(\Pi)}\mathbb{E}_{x\sim D}\sum_{a=1}^{K}P(a|x)\frac{1}{(1-K\gamma)P(a|x)+\gamma}\\
&\le \max_{P\in\Delta(\Pi)}\mathbb{E}_{x\sim D}\sum_{a=1}^{K}P(a|x)\frac{1}{(1-K\gamma)P(a|x)}\\
&= \frac{K}{1-K\gamma}.
\end{aligned}$$

■

Thus, we have proved that a distribution Q_t satisfying the conditions (5) of Algorithm 1 exists. Next, we would like to establish that if we play according to such a distribution at each round, then we incur low regret.

We will establish the following proposition, from which the regret guarantee will follow as a corollary.

Proposition 6 (Concentration of regret estimates)**.** With probability at least $1-2\delta$, we have for all t

$$\max_{\pi\in\Pi_t}|\widehat{V}_t(\pi)-V(\pi)|\le\epsilon_t.$$

Proof: Fix a time step t and a policy $\pi\in\Pi_t$. Recall Equation 6 and Lemma 3 which collectively show that $\widehat{V}_t(\pi)-V(\pi)$ is a sample-average of a martingale difference sequence. Furthermore, each $Y_t(\pi)$ satisfies $|Y_t(\pi)|\le 1/\gamma$, since the rewards are bounded in $[0,1]$ and $p_t=Q_t^\gamma(\cdot|x_t)\ge\gamma$. Furthermore, we have

$$\begin{aligned}
\mathrm{Var}_t[Y_t(\pi)] &\le \mathbb{E}_t\left[r_t^2\frac{\mathbf{1}(a_t=\pi(x_t))}{p_t^2}\right]\\
&\overset{(a)}{\le} \mathbb{E}_t\left[\frac{\mathbf{1}(a_t=\pi(x_t))}{p_t^2}\right]\\
&= \mathbb{E}_t\mathbb{E}_{x_t\sim D}\left[\sum_{a=1}^{K}Q_t^\gamma(a|x_t)\frac{\mathbf{1}(a=\pi(x_t))}{Q_t^\gamma(a|x_t)^2}\right]\\
&= \mathbb{E}_t\mathbb{E}_{x_t\sim D}\left[\frac{1}{Q_t^\gamma(\pi(x_t)|x_t)}\right]\\
&\overset{(b)}{\le} 2K.
\end{aligned}$$

Here inequality (a) follows as $r_t\in[0,1]$ and (b) is a result of the constraint (5), since Q_t satisfies these constraints and $\pi\in\Pi_t\subseteq\Pi_{t-1}$. By the nested structure of Π_t, any $\pi\in\Pi_t$ is contained in all the previous Π_s so that the variance bound applies to each $Y_s(\pi)$.

Thus we have a bound on the range and conditional variances of the Y_t, and we can invoke Lemma 4 to bound their sample average. Applying the lemma twice, once for $Z_t=Y_t$ once for $Z_t=-Y_t$, we observe that with probability at least $1-2\delta'$, we have for any $\lambda\in[0,\gamma]$

$$\left|\sum_{s=1}^{t}Y_t(\pi)\right|\le(e-2)\lambda\,2Kt+\frac{\ln(1/\delta')}{\lambda}$$

We use the inequality with $\delta'=\delta/NT$ and $\lambda=\gamma$, which yields that with probability at least $1-2\delta/NT$, we have

$$\left|\sum_{s=1}^{t} Y_t(\pi)\right| \leq 2(e-2)\,\gamma K t + \frac{\ln(NT/\delta)}{\gamma} \tag{7}$$

To simplify further, we assume that T is large enough so that $\gamma = \sqrt{\frac{\ln(NT/\delta)}{2KT}}$. If not, then we have

$$\sqrt{\frac{\ln(NT/\delta)}{2KT}} \geq \frac{1}{2K} \quad \Rightarrow \quad \epsilon_t \geq 1 \ \ , t = 1, 2, \ldots, T.$$

Thus, the propisition is trivially true in the other case and we can focus on the desired setting of γ. Plugging this setting into Equation 7, we obtain that with probability at least $1 - 2\delta/NT$

$$\left|\sum_{s=1}^{t} Y_t(\pi)\right| \leq 2(e-2)\gamma K t + \frac{\ln(NT/\delta)}{\gamma} \leq \epsilon_t, \tag{8}$$

where the last inequality uses $e - 1 \leq 2$. Taking a union bound over all policies and rounds of the algorithm completes the proof. ■

Remark: We observe that the constraints (5) provide a direct bound on the variance of our $Y_t(\pi)$ random variables. As a result, we will often refer to these as the *variance constraints*.

Given the proposition, we can now state the main regret bound for Algorithm 1.

Theorem 7. *With probability at least $1 - 2\delta$, for all t, we have:*

1. *$\pi^\star \in \Pi_t$, that is, $\pi^\star$ is never eliminated.*
2. *$V(\pi) \geq V(\pi^\star) - 4\epsilon_t$ for all $\pi \in \Pi_t$.*

Consequently, the regret of Algorithm 1 is bounded with probability at least $1 - 2\delta$ by $17\sqrt{2TK \ln \frac{TN}{\delta}}$.

Proof: To simplify our handling of probabilities, let $\mathcal{E}$ refer to the event that the conclusion of Proposition 6 holds for all π, t. The proposition guarantees that $\mathbb{P}(\mathcal{E}^C) \leq 2\delta$.

We begin with the first part of the theorem. Let us inductively assume that $\pi^\star \in \Pi_s$ for all $s = 0, 1, 2, \ldots, t$. The base case is clearly true as $\pi^\star \in \Pi = \Pi_0$. We will now show that $\pi^\star \in \Pi_{t+1}$. We apply Proposition 6 after round t twice. Once with $\pi^\star$ and once with $\widehat{\pi}_t$. Our inductive assumption gives that $\pi^\star \in \Pi_t$ and $\widehat{\mathrm{Reg}}_t(\widehat{\pi}_t) = 0$ by definition so that it is also in Π_t. Consequently, the preconditions of Proposition 6 are satisfied for both policies, and we get under the event $\mathcal{E}$,

$$\widehat{V}_t(\pi^\star) \geq V(\pi^\star) - \epsilon_t, \quad \text{and} \quad V(\widehat{\pi}_t) \geq \widehat{V}_t(\widehat{\pi}_t) - \epsilon_t.$$

Adding the two inequalities, we obtain that

$$\widehat{V}_t(\pi^\star) \geq \widehat{V}_t(\widehat{\pi}_t) + V(\pi^\star) - V(\widehat{\pi}_t) - 2\epsilon_t \geq \widehat{V}_t(\widehat{\pi}_t) - 2\epsilon_t, \tag{9}$$

where the second inequality follows since $V(\pi^\star) \geq V(\widehat{\pi}_t)$ by definition of $\pi^\star$. This demonstrates that $\pi^\star$ is not eliminated at round $t + 1$ and the induction is complete.

By rearranging the terms in Equation 9 a bit differently, we also see that under $\mathcal{E}$, we have

$$V(\widehat{\pi}_t) \geq V(\pi^\star) - 2\epsilon_t.$$

Similarly, we observe that under $\mathcal{E}$, for any $\pi \in \Pi_t$

$$\widehat{V}_t(\pi) \geq \widehat{V}_t(\widehat{\pi}_t) - 2\epsilon_t \geq \widehat{V}_t(\pi^\star) - 2\epsilon_t \geq V(\pi^\star) - 3\epsilon_t.$$

Further adding the deviation between $\widehat{V}_t(\pi)$ and $V(\pi)$ completes the proof.

Finally, for the regret bound note that the algorithm always plays according to a distribution over Π_t at round t, except for the $K\gamma$ uniform random distribution which is added in. Hence, the cumulative regret of the algorithm is at most

$$\begin{aligned}\sum_{t=1}^{T}(1-K\gamma)4\epsilon_t + K\gamma T &\leq \sum_{t=1}^{T} 8\sqrt{\frac{2K\ln(NT/\delta)}{t}} + KT\sqrt{\frac{\ln(NT\delta)}{2KT}} \\ &\leq 16\sqrt{2KT\ln(NT/\delta)} + \sqrt{\frac{KT\ln(NT\delta)}{2}}.\end{aligned}$$

■

Remark: Here we analyze regret as the expected reward of our chosen action, compared with the expected reward of the best @inproceedingssyrgkanis2016improved, title=Improved regret bounds for oracle-based adversarial contextual bandits, author=Syrgkanis, Vasilis and Luo, Haipeng and Krishnamurthy, Akshay and Schapire, Robert E, booktitle=Advances in Neural Information Processing Systems, pages=3135–3143, year=2016 policy. Recall a problem from Homework 2 to obtain a bound on the actual obtained reward in terms of the empirically best policy using this expected regret.

2 A computationally efficient approach

The POLICYELIMINATION algorithm so far has little difference from EXP4, in that it is statistically optimal (albeit in a more limited i.i.d. setting), but computationally impractical. It turns out though, that the structure of POLICYELIMINATION can provide crucial guidance towards the development of efficient algorithms for the i.i.d. setting.

The main computational bottlenecks of POLICYELIMINATION are the elimination step and the requirement to enforce the variance constraints (5) for each surviving policy. It is natural to wonder if we can avoid elimination altogether, enforce the variance constraints over all the policies and somehow encourage the distribution Q_t to not place mass over policies with a large empirical regret? The next example shows that we cannot hope to enforce the variance constraints for all $\pi \in \Pi$, and also obtain low regret.

Example 8. Let us consider a policy class Π consisting of two policies π_{good} and π_{bad}. Suppose further that we have only two actions a_1 and a_2 with $v(x, a_1) = 1$ and $v(x, a_2) = 0$ for each context x. Furthermore, π_{good} always chooses a_1 and π_{bad} always chooses a_2 indpendent of the context. Now the variance constraint (5), if enforced for policy π_{bad} implies

$$\begin{aligned}&\mathbb{E}_x\left[\frac{1}{Q^\gamma(\pi_{\text{bad}}(x)|x)}\right] \leq 2K \\ &\Rightarrow \mathbb{E}_x\left[\frac{1}{Q^\gamma(a_2|x)}\right] \leq 2K \\ &\overset{(a)}{\Rightarrow} \mathbb{E}_x\left[\frac{1}{(1-K\gamma)Q(\pi_{\text{bad}}) + \gamma}\right] \leq 2K \\ &\Rightarrow (1-K\gamma)Q(\pi_{\text{bad}}) + \gamma \geq \frac{1}{2K} \\ &\Rightarrow Q(\pi_{\text{bad}}) \geq \frac{1-2K\gamma}{2K(1-K\gamma)}.\end{aligned}$$

Here the implication (a) follows since π_{bad} is the only policy which takes a_2. If we choose $\gamma \leq 1/(4K)$, then the RHS above is at least $1/(3K)$. Noting that $K = 2$, we pick π_{bad} with probability at least $1/6$ each round, meaning that our cumulative reward is at most $5T/6$, leading to a regret of at least $T/6$ for the algorithm.

This example tells us that it is damaging to our goal of low-regret, if we insist of having the variance constraints also for the bad policies π. This makes intuitive sense too, we only want to focus on taking the actions which help us distinguish amongst the good (rather not provably bad) policies. We will now see how to motivate the design of a different, and computationally efficient algorithn.

2.1 A new optimization problem and its statistical analysis

The search for a distribution Q satisfying constraints (5) can be seen as an optimization problem. We are looking for a distribution which puts mass only over the policies in Π_{t-1} and satisfies all the constraints. We now present a related, but different optimization problem which was developed in Agarwal et al. [2014]. We will subsequently show that any algorithm which picks a distribution based on this new optimization problem also enjoys low regret. The next section will show an efficient algorithm to solve the problem.

Optimization Problem (OP)

Given samples $(x_s, a_s, p_s, r_s)_{s=1}^t$, minimum probability γ and with $b_t(\pi) = \frac{\widehat{\text{Reg}}_t(\pi)}{4(e-2)\gamma \ln T}$, find $Q \in \Delta(\Pi)$ such that

$$\sum_{\pi \in \Pi} Q(\pi) b_t(\pi) \leq 2K \qquad \text{and} \tag{10}$$

$$\forall \pi \in \Pi \ : \ \mathbb{E}_{x \sim D}\left[\frac{1}{Q^\gamma(\pi(x)|x)}\right] \leq 2K + b_t(\pi). \tag{11}$$

Figure 1: The optimization algorithm (OP) in the ILOVETOCONBANDITS algorithm of Agarwal et al. [2014].

Qualitatively, the optimization problem OP in Figure 2.1 is not so different from the constraints (5). We are still looking for a distribution such that the estimation variance for the policies π can be controlled. Crucially though, *the algorithm does not explicitly eliminate bad policies now*, and enforces the low-variance constraints for each policy π. To circumvent the bad cases such as that of Example 8, we do allow for different bounds on the variance for each policy π. Intuitively, if $b(\pi) = \mathcal{O}(K)$, meaning that $\widehat{\text{Reg}}_t(\pi) = \mathcal{O}(\sqrt{K/t \ln(TN/\delta)})$ by our setting of γ, then the overall RHS in the variance constraints in OP is still $\mathcal{O}(K)$ like in the POLICYELIMINATION algorithm. But for the policies which are empirically poor, we allow for much larger estimation variance, as large as $\mathcal{O}(\sqrt{t})$ if the empirical regret is a constant. This relaxed constraint on the estimation variance for bad policies means that we do not risk over-exploring in the cases like Example 8.

Additionally, since the distribution Q is over all the policies and not just the empirically good policies, we require a constraint saying that Q should have good empirical regret. Recalling the setting of $b(\pi)$, we observe that the empirical regret of picking a policy according to Q is $\mathcal{O}(K\gamma) = \mathcal{O}(\sqrt{K/t \ln(TN/\delta)})$. Since we have low estimation variance for all policies with regret at that level, our analysis will show that the actual regret of Q is not much worse either. That is, we posit having a small empirical regret and a small estimation variance, which collectively imply a small population regret.

We will now make this intuition formal in a few lemmas. We will show how to obtain the regret bound for the algorithm by putting together these lemmas, while deferring the more technical steps to an appendix. Our analysis focuses on the following intuitive algorithm:

1. At time t, find a distribution Q_t which satisfies OP using past samples, i.e. with $b_{t-1}(\pi)$.
2. Observe x_t, and play an action according to $Q_t^\gamma(a|x_t)$.
3. Observe r_t and incorporate (x_t, a_t, p_t, r_t) to the dataset.

We begin with the statement of the overall theorem we will prove about any algorithm which plays a distribution satisfying OP at each round.

Theorem 9. *Suppose an algorithm plays according to a distribution which is a solution to* OP *at each round t. Then with probability at least $1-\delta$, the algorithm incurs a regret no more than $\mathcal{O}\left(\sqrt{KT\ln\frac{TN}{\delta}} + K\ln\frac{TN}{\delta}\right)$.*

Remark: Note that we have no explicitly shown that the optimization problem OP is even feasible, but only asserted a bound on regret assuming that a distribution feasible according to it can be found. In the next section, we will show an explicit construction for distributions satisfying it, which implicitly yields feasibility too.

Remark: The theorem shows that the less stringent constraints in OP, compared with (5) do not cost us statistically. The next section will show how this creates the path to an efficient algorithm.

Remark: As before we assume that the constraints involve expectations under the actual distribution D over the contexts. This can be relaxed to an empirical average, as detailed in Appendix B.2 of Agarwal et al. [2014].

2.2 Some helper lemmas

We now present a couple of lemmas which can be easily put together to obtain the theorem. The first lemma can be seen as an analog of Proposition 6.

Lemma 10. *Under conditions of Theorem 9, with probability at least $1-\delta$, we have for all $t = 1, 2, \ldots, T$ all policies $\pi \in \Pi$ and $\lambda \in [0, \gamma]$*

$$\left|\widehat{V}_t(\pi) - V(\pi)\right| \le (e-2)\lambda\left(2K + \frac{1}{t}\sum_{s=1}^{t} b_{s-1}(\pi)\right) + \frac{\ln\frac{TN}{\delta}}{\lambda t}.$$

The lemma follows almost directly from combining Freedman's inequality (Lemma 4) along with our variance constraints in OP, similar to how we used the variance constraints in the POLICYELIMINATION algorithm.

Remark: In order to apply the lemma at round 1, we also define $b_0(\pi) = 0$, and $2K + b_0(\pi) = 2K$ is still an upper bound on the variance at round 1, since a distribution satisfying this bound for all policies π exists by the Lemma 5 applied to the first step of the POLICYELIMINATION algorithm.

We next show how the deviation bound on reward estimates implies concentration of regret.

Lemma 11. *Under conditions of Theroem 9, with probability at least $1-\delta$, we have for all $t = 1, 2, \ldots, T$ and all $\pi \in \Pi$:*

$$\mathrm{Reg}(\pi) \le 2\widehat{\mathrm{Reg}}_t(\pi) + \epsilon_t, \qquad \text{and} \qquad \widehat{\mathrm{Reg}}_t(\pi) \le 2\mathrm{Reg}(\pi) + \epsilon_t,$$

where $\epsilon_t = 4\sqrt{\frac{K}{t}\ln\frac{TN}{\delta}} + \frac{32}{\gamma t}\ln\frac{TN}{\delta}$.

Remark: We see that unlike in the POLICYELIMINATION algorithm, we are not guaranteeing that $\mathrm{Reg}(\pi)$ and $\widehat{\mathrm{Reg}}_t(\pi)$ are necessarily close (due to the factor of 2). However, we do still guarantee that if $\widehat{\mathrm{Reg}}_t(\pi) = \mathcal{O}(\epsilon_t)$, then $\mathrm{Reg}(\pi) \le \widehat{\mathrm{Reg}}_t(\pi) + c\,\epsilon_t$ for some constant c. Note that ϵ_t is quite similar to the elimination threshold we

used in the POLICYELIMINATION algorithm. So we are asserting that for all the *good policies*, we still have good concentration of empirical regret around its expectation, but the concentration is allowed to be worse for bad policies. This is a natural consequence of having a higher variance on bad policies.

The proof of the lemma effectively uses Lemma 10 twice, for π and $\pi^\star$ to bound the deviation of regret. We present a rough sketch here, with a detailed proof in Appendix A.

We begin with the observation:

$$\mathrm{Reg}(\pi) - \widehat{\mathrm{Reg}}_t(\pi) \leq V(\pi^\star) - \widehat{V}_t(\pi^\star) - V(\pi) + \widehat{V}_t(\pi),$$

and invoke Lemma 10 on each of the deviation terms. The main departure from our previous analysis is that the deviation now depends on the policy through the $b_{s-1}(\pi)$ terms in Lemma 10.

In order to reason about them, Lemma 11 is proved inductively. Since $b_{s-1}(\pi)$ is $\widehat{\mathrm{Reg}}_{s-1}(\pi)$ up to scaling factors, it is also related to $\mathrm{Reg}(\pi)$ under the inductive assumption. That is, we obtain using the inductive assumption:

$$\begin{aligned}
\mathrm{Reg}(\pi) - \widehat{\mathrm{Reg}}_t(\pi) &\leq \frac{2\ln\frac{TN}{\delta}}{\lambda t} + 4(e-2)\lambda K + (e-2)\lambda\left(\frac{1}{t}\sum_{s=1}^{t}(b_{s-1}(\pi) + b_{s-1}(\pi^\star))\right) \\
&\leq \frac{2\ln\frac{TN}{\delta}}{\lambda t} + 4(e-2)\lambda K + \frac{\lambda}{4\gamma\ln T}\left(\frac{1}{t}\sum_{s=1}^{t}(2\mathrm{Reg}(\pi) + 2\mathrm{Reg}(\pi^\star) + 2\epsilon_{s-1})\right).
\end{aligned}$$

Rearranging terms along with some algebra now gives the upper bound on $\mathrm{Reg}(\pi)$ in Lemma 11. Following a similar logic also yields the bound on $\widehat{\mathrm{Reg}}_t(\pi)$ in terms of $\mathrm{Reg}(\pi)$. A detailed proof of this result can be found in Appendix A.

Given Lemma 11, the proof of Theorem 9 is quite straightforward.

2.3 Proof of Theorem 9

Under the conditions of Theorem 9, at each round we play according to a distribution Q_t which satisfies the constraints in OP. This implies, in particular that the distribution Q has a low, empirical regret. Specifically,

$$\sum_{\pi\in\Pi} Q_t(\pi)\widehat{\mathrm{Reg}}_{t-1}(\pi) \leq 8K(e-2)\gamma\ln T.$$

Using Lemma 11 this immediately yields a bound on the expected regret of picking policies according to Q. Now we just have to factor in the additional regret we incur due to mixing in the uniform distribution, which is bounded by γT. Substituting the value of γ and simplifying yields the desired guarantee.

3 An efficient algorithm to solve OP

The approach of solving OP instead of the POLICYELIMINATION algorithm eliminates the need to explicitly maintain the set of good policies. However, checking the feasibility of a distribution Q for OP is still challenging since the set of constraints is extremely large, with one constraint for each policy π. Naïvely, it appears just as intractable as POLICYELIMINATION or EXP4 approaches. However, we will see that the structure of OP is amenable to efficient solutions by leveraging the existence of $\mathcal{AMO}$ for Π. We now present an algorithm to solve OP which can be

implemented by appropriately invoking the $\mathcal{AMO}$. We will also discuss the computational efficiency of this algorithm in finding a solution to OP. *Throughout, we will drop the time index t* since the entire analysis is focusing on an arbitrary round of the contextual bandit algorithm.

Algorithm 2 Coordinate descent algorithm for solving OP

Require: Initial Q_{init}. Q_{init} can be any elementwise non-negative vector with entries summing to no more than 1.

Intialize $Q := Q_{\text{init}}$.

loop

Define the following quantities for all $\pi \in \Pi$:

$$\text{Var}_\pi(Q) = \mathbb{E}_x\left[\frac{1}{Q^\gamma(\pi(x)|x)}\right], \quad S_\pi(Q) = \mathbb{E}_x\left[\frac{1}{\left(Q^\gamma(\pi(x)|x)\right)^2}\right], \text{ and } D_\pi(Q) = \text{Var}_\pi(Q) - (2K + b(\pi)) \quad (12)$$

if $\sum_{\pi\in\Pi} Q(\pi)(2K + b(\pi)) > 2K$ **then**

Replace Q by cQ where

$$c = \frac{2K}{\sum_{\pi\in\Pi} Q(\pi)(2K + b(\pi))} < 1.$$

end if

if There is a policy π for which $D_\pi(Q) > 0$ **then**

Update $Q(\pi') = Q(\pi') + \alpha$, if $\pi' = \pi$ and leave $Q(\pi')$ unchanged otherwise, where

$$\alpha = \frac{\text{Var}_\pi(Q) + D_\pi(Q)}{2(1 - K\gamma)S_\pi(Q)}.$$

else

Let $\theta = \sum_{\pi\in\Pi} Q(\pi)$.

Halt and output $Q + (1 - \theta)\mathbf{1}(\widehat{\pi})$, where $\mathbf{1}(\widehat{\pi})$ is a distribution which picks the empirically best policy $\widehat{\pi}$ with probability 1.

end if

end loop

The algorithm can be intuitively seen as trying to greedily satisfy the constraints in OP. The first check and renormalization of the distribution Q is clearly enforcing the low-regret constraint in OP. The quantity $D_\pi(Q)$ is simply the violation of the variance constraint for policy π in OP, and then algorithm increases the weight on π if this constraint is violated, which should decrease the constraint violation. Given these properties, the following lemma is immediate.

Lemma 12. *If Algorithm 2 halts and outputs a distribution Q, then it is feasible for* OP.

Remark: The algorithm tacitly assumes that the quantity $\theta < 1$ in the last line, which is also required for the above assertion of correctness. However, this follows since we are assured that $\sum_\pi Q(\pi)(2K + b(\pi)) \leq 2K$ when the algorithm terminates. Recalling the non-negativity of $b(\pi)$, this implies that $\sum_\pi Q(\pi) \leq 1$.

Given the correctness, we will now study the computational properties of Algorithm 2. We begin by showing how the algorithm can be implemented using an $\mathcal{AMO}$ for Π first. Subsequently, we will bound the number of iterations of Algorithm 2 before termination, which will yield a bound on the number of $\mathcal{AMO}$ calls.

Lemma 13. *Algorithm 2 can be implemented with one call to $\mathcal{AMO}$ before the loop is started, and one call for each iteration of the loop thereafter.*

Proof: Before starting the loop, we make a single call to the $\mathcal{AMO}$ to obtain the empirically best policy $\widehat{\pi}$ so far, which is required in order to evaluate $b(\pi)$ (or rather $\widehat{\text{Reg}}(\pi)$).

Now we have two main computational steps in the algorithm. There is the computation of $\sum_{\pi\in\Pi} Q(\pi)b(\pi)$ before we potentially renormalize the distribution. This computation is efficient even for large Π as long as the distribution Q has a sparse support. For instance, assuming Q_{init} is the all 0's vector, since Q adds a non-zero weight to only one policy at each iteration, the size of its support is bounded by the number of iterations.

The key computational burden then, is in the checking if there is a policy π satisfying $D_\pi(Q) > 0$, that is there is a policy whose variance constraint in OP is violated. Note that the $\mathrm{Var}_\pi(Q)$ term in $D_\pi(Q)$ is

$$\mathbb{E}_{x\sim D}\left[\frac{1}{Q^\gamma(\pi(x)|x)}\right] \approx \frac{1}{t}\sum_{s=1}^{t}\frac{1}{Q^\gamma(\pi(x_t)|x_t)},$$

where we have replaced the true expectation with a sample average involving the previously seen contexts. While this might seem like a heuristic argument, the paper [Agarwal et al., 2014] makes this rigorous by using sample averages in the variance constraints of OP, and bounding the error from doing this. For simplicitly, we will continue to assume that the variance terms are a sample average for this remainder of the computational analysis. Ignoring the constant $2K$ in $D_\pi(Q)$, the other π-dependent term is $b(\pi)$, which can be simplified as

$$b_t(\pi) = \frac{\widehat{\mathrm{Reg}}_t(\pi)}{\psi\gamma} = \frac{\widehat{V}_t(\widehat{\pi}_t)}{\psi\gamma} - \frac{1}{\psi\gamma t}\sum_{s=1}^{t}\widehat{v}_t(x_t,\pi(x_t)),$$

where $\psi = 4(e-2)\ln T$ and $\widehat{v}_t(x_t,a)$ is the `IPS` estimator for the reward of action a, given the tuple (x_t,a_t,p_t,r_t) observed by the algorithm.

In order to check whether a policy with $D_\pi(Q) > 0$ exists, it suffices to find $\arg\max_{\pi\in\Pi} D_\pi(Q)$ and compute the corresponding maximum value. By the above simplifications, we see that we need to compute

$$\arg\max_{\pi\in\Pi} D_\pi(Q) = \arg\max_{\pi\in\Pi} \mathrm{Var}_\pi(Q) - 2K - b(\pi) = \arg\max_{\pi\in\Pi} \frac{1}{t}\sum_{s=1}^{t}\frac{1}{Q^\gamma(\pi(x_t)|x_t)} + \frac{1}{\psi\gamma t}\sum_{s=1}^{t}\widehat{v}_t(x_t,\pi(x_t)),$$

where we dropped all the terms indepedent of π. Now, given any round t and action a, we define

$$\tilde{r_t}(a) = \widehat{v}_t(x_t,a) + \frac{\psi\gamma}{Q^\gamma(a|x_t)}.$$

We then create a dataset $(x_s,\tilde{r_s})_{s=1}^t$ and feed it to the $\mathcal{AMO}$. It is now easily seen that the output of the $\mathcal{AMO}$ can be used to check whether $D_\pi(Q) > 0$, which concludes the proof. ■

Thus, each iteration of Algorithm 2 is computationally feasible. It now remains to bound the number of iterations, which controls both the sparsity of the distribution Q as well as the number of calls to the $\mathcal{AMO}$. We present this bound in the next theorem.

Theorem 14. *Algorithm 2 with $Q_{init} = 0$ terminates in at most $\frac{4\ln(1/(K\gamma)}{\gamma}$ iterations and outputs a solution to* OP.

Remark: Recalling the setting of γ, we see that the algorithm terminates in at most $\widetilde{\mathcal{O}}(\sqrt{T/K\ln(TN/\delta)})$ iterations. That is, we make $\widetilde{\mathcal{O}}(\sqrt{T})$ calls to the $\mathcal{AMO}$ at every round, and a total of $\widetilde{\mathcal{O}}(T^{1.5})$ calls across all rounds. In Agarwal et al. [2014], a more sophisticated epoch and warm-starting strategy is used to bring the overall number of calls down to $\widetilde{\mathcal{O}}(\sqrt{T})$ *across all T rounds*. That is, we just need a sublinear number of $\mathcal{AMO}$ calls, and this is unavoidable for any algorithm which solves OP due to a corresponding lower bound in that paper.

Proof: The proof uses a potential function argument, and we will only present the ideas at a very high-level. The key insight of the algorithm is in finding a potential function $\Phi(Q)$, such that minimizing $\Phi(Q)$ leads to finding a solution to OP. Since the function is acting over the N-dimensional variable Q, and we also have N variance constraints in OP, the function Φ is designed such that its directional derivative along policy π is effectively the constraint violation

$D_\pi(Q)$. Since the derivative vanishes at optimum, we can have no violation of the variance constraints. The handling of the low-regret constraint requires additional care, but is ensured by the re-normalization step in Algorithm 2.

Specifically Agarwal et al. [2014] show that if $\sum_\pi Q(\pi)(2K + b(\pi)) > 2K$, then $\Phi(cQ) \leq \Phi(Q)$ for the constant c in Algorithm 2. This implies that the renormalization step never increases the potential.

They further show that everytime we find a π such that $D_\pi(Q) > 0$ and update the distribution to add α mass on π, this leads to a significant decrease in the potential.

The convergence result now follows since the potential function is chosen to be upper and lower bounded, and is decreased significantly everytime we find a variance constraint violation. Hence, after the number of updates postulated in the theorem statement, the function has to be reduced to its minimum value. ■

The above argument is rather vague, but intentionally so. We encourage the reader to study the exact form of the potential, and the details of the convergence analysis in the paper.

In summary, the whole exercise yields an algorithm which invokes the $\mathcal{AMO}$ at most $\widetilde{\mathcal{O}}(\sqrt{T})$ times over T rounds and enjoys the statistically optimal regret guarantee for i.i.d. contextual bandit problems. In contrast, τ-GREEDY requires exactly one $\mathcal{AMO}$ call but is suboptimal in regret, while EXP4 has no known implementation via the $\mathcal{AMO}$. Subsequent works [Syrgkanis et al., 2016a, Rakhlin and Sridharan, 2016, Syrgkanis et al., 2016b] have explored other trade-offs in terms of the computational and statistical efficiencies, as well as the distributional assumptions on the problem.

For practical implementation, τ-GREEDY or its ϵ-GREEDY version are still the most convenient and efficient to implement. While the approach presented here can be also implemented using an $\mathcal{AMO}$, doing so is non-trivial and expensive. In the paper [Agarwal et al., 2014], we discuss epoching strategies to ease the computational cost, as well as online approximations to develop a more practical algorithm. Having strong regret guarantees for an online algorithm remains a challenging open question in this area.

Appendix A Proof of Lemma 10

We give a more formal version of our proof sketch here. As stated earlier, we will prove the lemma by induction. Starting with the base case, we check that $\epsilon_1 > 1$. Since the $\mathrm{Reg}(\pi) \leq 1$, the upper bound on population regret easily follows. For the upper bound on empirical regret, we note that $\epsilon_1 \geq 1/\gamma$ as well, but the empirical regret is no more than $1/\gamma$ meaning that the base case is satisfied. We will now assume that the statement of the lemma holds for all $s = 1, 2, \ldots, t-1$ and all policies π and establish it at round t. To simplify our treatment of probabilities, we will assume that the $1 - \delta$ probability event in Lemma 10 holds.

We will only work out the upper bound on $\mathrm{Reg}(\pi)$ in terms of $\widehat{\mathrm{Reg}}_t(\pi)$ since the proof of the other bound follows similarly. Now we observe that

$$\begin{aligned}\mathrm{Reg}(\pi) - \widehat{\mathrm{Reg}}_t(\pi) &= V(\pi^\star) - V(\pi) - \widehat{V}_t(\widehat{\pi}_t) - \widehat{V}_t(\pi) \\ &\leq V(\pi^\star) - V(\pi) - \widehat{V}_t(\pi^\star) - \widehat{V}_t(\pi),\end{aligned}$$

where the inequality follows since $\widehat{V}_t(\widehat{\pi}_t) \geq \widehat{V}_t(\pi^\star)$ by the definition of $\widehat{\pi}_t$. Next we apply Lemma 10 twice, once with π and once with $\pi^\star$ to obtain

$$\begin{aligned}
\mathrm{Reg}(\pi) - \widehat{\mathrm{Reg}}_t(\pi) &\leq \frac{2\ln\frac{TN}{\delta}}{\lambda t} + 4(e-2)\lambda K + (e-2)\lambda\left(\frac{1}{t}\sum_{s=1}^{t}(b_{s-1}(\pi) + b_{s-1}(\pi^\star))\right) \\
&\leq \frac{2\ln\frac{TN}{\delta}}{\lambda t} + 4(e-2)\lambda K + \frac{\lambda}{4\gamma t\ln T}\left(\sum_{s=2}^{t}(\widehat{\mathrm{Reg}}_{s-1}(\pi) + \widehat{\mathrm{Reg}}_{s-1}(\pi^\star))\right) \\
&\leq \frac{2\ln\frac{TN}{\delta}}{\lambda t} + 4(e-2)\lambda K + \frac{\lambda}{4\gamma t\ln T}\left(\sum_{s=2}^{t}(2\mathrm{Reg}(\pi) + 2\mathrm{Reg}(\pi^\star) + 2\epsilon_{s-1})\right) \\
&= \frac{2\ln\frac{TN}{\delta}}{\lambda t} + 4(e-2)\lambda K + \frac{\lambda}{2\gamma\ln T}(\mathrm{Reg}(\pi) + \mathrm{Reg}(\pi^\star)) + \frac{\lambda}{2\gamma t\ln T}\left(\sum_{s=2}^{t}\epsilon_{s-1}\right). \qquad (13)
\end{aligned}$$

Here the first inequality uses the definition of $b_{s-1}(\pi)$ (and $b_0(\pi) = 0$), while the second inequality uses the inductive hypothesis twice, once for π and once for $\pi^\star$. In order to simplify further, we bound the sum of ϵ_{s-1} terms first. We have

$$\begin{aligned}
\sum_{s=2}^{t}\epsilon_{s-1} &= \sum_{s=1}^{t}\left(4\sqrt{\frac{K}{s-1}\ln\frac{TN}{\delta}} + \frac{32}{\gamma(s-1)}\ln\frac{TN}{\delta}\right) \\
&\leq 8\sqrt{Kt\ln\frac{TN}{\delta}} + \frac{64\ln t}{\gamma}\ln\frac{TN}{\delta},
\end{aligned}$$

since $\sum_{s=1}^{t} 1/\sqrt{s} \leq 2\sqrt{t}$ and $\sum_{s=1}^{t} 1/s \leq 2\ln t$. Plugging this into our earlier bound (13) and noting that $\mathrm{Reg}(\pi^\star) = 0$, we obtain

$$\mathrm{Reg}(\pi) - \widehat{\mathrm{Reg}}_t(\pi) \leq \frac{2\ln\frac{TN}{\delta}}{\lambda t} + 4(e-2)\lambda K + \frac{\lambda}{2\gamma\ln T}\mathrm{Reg}(\pi) + \frac{\lambda}{2\gamma t\ln T}\left(8\sqrt{Kt\ln\frac{TN}{\delta}} + \frac{64\ln t}{\gamma}\ln\frac{TN}{\delta}\right).$$

In order to collect terms, we pick $\lambda = \gamma/4 \in [0, \gamma]$ and obtain

$$\begin{aligned}
\mathrm{Reg}(\pi) - \widehat{\mathrm{Reg}}_t(\pi) &\leq \frac{8\ln\frac{TN}{\delta}}{\gamma t} + (e-2)\gamma K + \frac{1}{8\ln T}\mathrm{Reg}(\pi) + \frac{1}{8t\ln T}\left(8\sqrt{Kt\ln\frac{TN}{\delta}} + \frac{64\ln t}{\gamma}\ln\frac{TN}{\delta}\right) \\
&\leq \frac{16\ln\frac{TN}{\delta}}{\gamma t} + \frac{\mathrm{Reg}(\pi)}{2} + 2\sqrt{\frac{K}{t}\ln\frac{TN}{\delta}}.
\end{aligned}$$

In the last inequality, we have also used the definition of $\gamma = \sqrt{\frac{K}{t}\ln\frac{TN}{\delta}}$, along with algebraic simplifcations. Rearranging terms completes the proof.

References

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert E Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *ICML*, 2014.

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

Alexander Rakhlin and Karthik Sridharan. Bistro: An efficient relaxation-based method for contextual bandits. In *ICML*, 2016.

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

Vasilis Syrgkanis, Akshay Krishnamurthy, and Robert E Schapire. Efficient algorithms for adversarial contextual learning. In *ICML*, 2016a.

Vasilis Syrgkanis, Haipeng Luo, Akshay Krishnamurthy, and Robert E Schapire. Improved regret bounds for oracle-based adversarial contextual bandits. In *Advances in Neural Information Processing Systems*, pages 3135–3143, 2016b.