Lecture 4: Contextual Bandits

Akshay Krishnamurthy
akshay@cs.umass.edu

February 14, 2022

1 The contextual bandit problem

So far, we have studying the exploration-exploitation tradeoff in relatively simple bandit settings, where there is little need for generalization. In particular we've focused on settings where there is a single reward function that is fixed over time, which may not be a good model when we have a system that is interacting with many different users or in many different scenarios. In this lecture, we'll extend the protocol to model decision making in a variety of scenarios, which will lead to the development of algorithms that can explore *and* generalize.

The protocol is known as *contextual bandits*. The main new feature is that on each round the learner observes some "contextual information" which it may use to inform its choice of action. For most of the lecture, and in most of the literature, the context will be an abstract object, which allows us to instantiate the protocol in many diverse problems. For example, in recommendation settings, the context might be the user coming to our system as well as a short list of items we may choose to display.

While there are many formulations with somewhat minor differences, a basic stochastic protocol is as follows. There is an abstract context space $\mathcal{X}$, action space $\mathcal{A}$, a distribution $\mathcal{D}$ supported on $\mathcal{X}$, and a reward function $R : \mathcal{X} \times \mathcal{A} \to \Delta([0,1])$. Let us define $f^\star := (x,a) \mapsto \mathbb{E}_{r\sim R(x,a)}[r]$ to be the mean reward function. The learning process proceeds for T rounds where in round t:

1. Nature samples a *context* $x_t \sim \mathcal{D}$ and presents it to the learner.
2. Learner examines the context and uses it to choose an action $a_t \in \mathcal{A}$.
3. Learner collects reward $r_t \sim R(x_t, a_t)$ which is also observed.

As usual we want the learner to accumulate a lot of reward, measured as $\sum_{t=1}^T r_t$ or $\sum_t f^\star(x_t, a_t)$. To do this, we will provide the learner with some function class so that we can incorporate inductive biases and generalize across contexts. There are two different formulations for the function class.

- *Policy class.* Here we give the learner a class $\Pi : \mathcal{X} \to \mathcal{A}$ (or $\mathcal{X} \to \Delta(\mathcal{A})$) that directly maps contexts to actions. This class will have a best policy $\pi^\star := \operatorname{argmax}_{\pi\in\Pi} \mathbb{E}_{x\sim\mathcal{D}}[f^\star(x, \pi(x))]$ and we will measure regret relative to this policy:

$$\mathrm{Regret}(T, \Pi) = \sum_{t=1}^T f^\star(x_t, \pi^\star(x_t)) - \sum_{t=1}^T f^\star(x_t, a_t).$$

  Note that this policy class could be quite arbitrary and in this formulation we haven't made any assumptions about the distribution $\mathcal{D}$ or the reward function R. In particular $\pi^\star$ may not actually get much reward. This formulation is analogous to *agnostic learning* in the supervised/statistical setting.

- *Value function class.* Here we give the learner a class $\mathcal{F} : (\mathcal{X} \times \mathcal{A}) \to [0,1]$ that can be used to score context-action pairs. Any "value function" f induces a "greedy" policy $\pi_f : x \mapsto \operatorname{argmax}_a f(x,a)$ that takes the highest scoring action. So from $\mathcal{F}$ we can derive a policy class $\Pi_\mathcal{F} := \{\pi_f : f \in \mathcal{F}\}$ and we can measure regret relative to the best policy in $\Pi_\mathcal{F}$.

Since it induces a policy class, the value function class provides more information to the learner, but it turns out that this information is not so useful unless we make further assumptions. The most standard assumption is *realizability* which is that that mean reward function $f^\star$ is actually in the function class $\mathcal{F}$ (This is analogous to our assumption that the rewards were linearly realizable in linear stochastic bandits). It is not hard to show that $\pi_{f^\star}$ is the globally (unconstrained) optimal policy, and since $f^\star \in \mathcal{F}$, we have $\pi_{f^\star} \in \Pi_{\mathcal{F}}$ so we are interested in competing with the globally optimal policy when we measure regret.

However, we are making a strong assumption. As you will see on the homework, this can be relaxed somewhat and this model inspires the design of algorithms that work quite well in practice, where realizability is certainly not going to hold.

Linear contextual bandits. A simple observation is that LinUCB can be used as is in the "value function class" setting. Unlike the "large actions" version we saw last time, let us instead consider a featurization of context-action pairs, that is $\phi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$. Then our function class is linear functions on top of this feature map, that is $\mathcal{F} = \{\phi \mapsto \langle \theta, \phi \rangle : \theta \in \mathbb{R}^d, \|\theta\|_2 \leq W\}$ and we assume linear realizability. Then at each round t we receive x_t and we have a new feature set $\{\phi(x_t, a)\}_{a \in \mathcal{A}}$ but we can still choose the UCB action $\operatorname{argmax}_{a \in \mathcal{A}} \left\langle \hat{\theta}, \phi(x_t, a) \right\rangle + \beta \|\phi(x_t, a)\|_{\Lambda^{-1}}$.

Intuitively, even though the features are changing from round to round, we can transfer information between rounds through our estimate (and confidence) on $\theta^\star$. Almost exactly the same proof we saw for LinUCB will apply in this setting and it will give a $\tilde{O}(d\sqrt{T})$ regret bound.

Oracle efficiency and reductions. Beyond the linear approach, much of the literature on contextual bandits has focused on what are known as oracle-efficient algorithms. I would say that this has also been the most successful approach for designing practically useful algorithms.

Let's start in the policy class variant of the problem. The key question that arises is how should we think about designing a computationally efficient algorithm? In fact, a variant of EXP3 can work in the contextual bandit setting and achieve $O(\sqrt{AT \log |\Pi|})$ regret, but it requires maintaining and updating a weight for each policy, so the running time is $O(|\Pi|T)$. The fact that the regret scales logarithmically with $|\Pi|$ but the runtime is linear is somewhat disappointing, because the running time prevents us from using a large function class.

The oracle-efficient approach is a way to avoid this by assuming (abstractly) that your function class is structured in some way. In the policy class version, the standard oracle assumption is that the class Π supports efficient optimization for "classification" problems. That is, given any dataset $D = \{(x_i, \vec{r}_i)\}_{i=1}^n$ of full information context-reward vectors, we can *compute*

$$\operatorname*{argmax}_{\pi \in \Pi} \sum_{i=1}^{n} r_i(x_i, \pi(x_i))$$

This can be seen as a weighted classification problem. Even though there is little theory for this, we do solve such problems fairly routinely in practice using highly non-linear function classes. So the oracle-efficient approach is a modular way to compose optimization heuristics that work well in practice with exploration. It is important to note that this is a purely computational assumption, since we could implement this oracle simply by enumerating over the policy class. However it does provide a nontrivial restriction when designing algorithms.

In the value functions variant of the problem, it is more natural to assume that we can solve regression problems over the function class $\mathcal{F}$. Let us turn to one instantiation of this oracle.

2 Contextual bandits with regression oracles

Recently, Foster and Rakhlin developed a very clean reduction from contextual bandits to *online regression*. They consider the value function version of the problem with function class $\mathcal{F}$, and they assume access to an online regression oracle SqAlg that operates in a full-information regression protocol. At each round t of this protocol: (1) nature chooses an input $z_t = (x_t, a_t)$ and an outcome y_t, (2) the algorithm sees z_t and makes a prediction $\hat{y}_t \in \mathbb{R}$,

(3) algorithm observes y_t and suffers loss $(\hat{y}_t - y_t)^2$. The oracle assumption is that for any sequence $\{(z_t, y_t)\}_{t=1}^T$:

$$\sum_{t=1}^{T}(\hat{y}_t - y_t)^2 - \min_{f\in\mathcal{F}}\sum_{t=1}^{T}(f(z_t) - y_t)^2 \leq \mathrm{Reg}_{\mathrm{Sq}}(T).$$

The algorithm can also be asked at any time for predictions on any $z \in \mathcal{Z}$. You can think of this as freezing the internal state of the algorithm passing it z, getting $\hat{y}(z)$, but then not doing an update.

The online learning community has developed algorithms that typically get $\mathrm{Reg}_{\mathrm{Sq}}(T) \asymp \log(T)$. For example if $\mathcal{F}$ is linear functions as in the LinUCB setting, there is an efficient algorithm that gets $\mathrm{Reg}_{\mathrm{Sq}}(T) \leq O(d\log(T/d))$. However, it is important to note that this is not purely a computational assumption as we are also assuming that the function class supports some online learning guarantee.

Using the online oracle, the algorithm proceeds as follows. Given a learning rate parameter γ at round t we receive context x_t and do the following:

1. For each action a get predictions $\hat{y}_{t,a} := \hat{y}_t(x_t, a)$ from SqAlg.
2. Let $b_t := \mathrm{argmax}_{a\in\mathcal{A}}\,\hat{y}_{t,a}$
3. Set

$$p_t(a) := \frac{1}{A + \gamma(\hat{y}_{t,b_t} - \hat{y}_{t,a})}, \qquad p_t(b_t) := 1 - \sum_{a\neq b_t} p_t(a)$$

4. Sample $a_t \sim p_t$, observe $r_t \sim R(x_t, a_t)$ and pass (x_t, a_t) with target r_t to SqAlg.

This algorithm is called SquareCB. The action selection is based on a technique called "inverse gap weighting," or IGW. Here the gaps are the differences between our predictions, $\hat{y}_{t,b_t} - \hat{y}_{t,a}$. The intuition for IGW is that if action a has a large gap it should be played infrequently, since we expect it to be quite suboptimal. In particular our predicted best action b_t should be played quite frequently.

Alternatively, the probabilities are set to correspond to how much we'll learn if we are wrong. If we predict incorrectly that an action is very bad, then we will learn quite a lot when we play it. So we can afford to play it somewhat infrequently, since when we do play it, we'll immediately realize how wrong we were.

The guarantee for SquareCB is a reduction from contextual bandit regret to online square loss regret.

Theorem 1. *Setting $\gamma = \sqrt{AT/(\mathrm{Reg}_{\mathrm{Sq}}(T) + \log(2/\delta))}$, SquareCB guarantees (with probability $\geq 1-\delta$):*

$$\sum_{t=1}^{T} f^\star(x_t, \pi^\star(x_t)) - f^\star(x_t, a_t) \leq 4\sqrt{AT\cdot\mathrm{Reg}_{\mathrm{Sq}}(T)} + 8\sqrt{AT\log(2/\delta)}.$$

Proof. The first claim is that the following two inequalities hold with probability at least $1-\delta$

$$\sum_{t=1}^{T} f^\star(x_t, \pi^\star(x_t)) - f^\star(x_t, a_t) \leq \sum_t\sum_a p_t(a)\left[f^\star(x_t, \pi^\star(x_t)) - f^\star(x_t, a_t)\right] + \sqrt{2T\log(2/\delta)},$$

$$\sum_t\sum_a p_t(a)(\hat{y}_t(x_t, a) - f^\star(x_t, a))^2 \leq 2\mathrm{Reg}_{\mathrm{Sq}}(T) + 16\log(2/\delta). \tag{1}$$

The first of these is by a concentration argument (Azuma-Hoeffding). The second also uses concentration, but additionally that with realizability, for any t, taking expectation over just the random action a_t and any randomness in the reward, we have

$$\begin{aligned}
&\mathbb{E}_{a\sim p_t, r_t(x_t,a)}\left[(\hat{y}_t(x_t, a) - r_t(x_t, a)^2 - (f^\star(x_t, a) - r_t(x_t, a))^2\right] \\
&= \sum_a p_t(a)\mathbb{E}_{r_t(x_t,a)}\left[\hat{y}_t(x_t, a)^2 - f^\star(x_t, a)^2 - 2r_t(x_t, a)(\hat{y}_t(x_t, a) - f^\star(x_t, a))\right] \\
&= \sum_a p_t(a)(\hat{y}_t(x_t, a) - f^\star(x_t, a))^2.
\end{aligned}$$

So the left hand side of (1) is the (conditional) expectation of the square loss regret. Intuitively one should concentrate to the other, but the fact that we don't have a $\sqrt{T}$ on the deviation term is important. In Appendix A, we will see why this is true.

Now, by adding and subtracting we have

$$\sum_{t=1}^{T} f^\star(x_t, \pi^\star(x_t)) - f^\star(x_t, a_t) \leq \sum_{t=1}^{T} \sum_a p_t(a) \left[(f^\star(x_t, \pi^\star(x_t)) - f^\star(x_t, a)) - \frac{\gamma}{4} (\hat{y}_t(x_t, a) - f^\star(x_t, a))^2 \right] + \frac{\gamma}{2} \mathrm{Reg}_{\mathrm{Sq}}(T) + 4\gamma \log(2/\delta) + \sqrt{2T \log(2/\delta)}$$

The theorem follows by using the next lemma to bound the first term by $2AT/\gamma$ and then by our choice of γ. □

Lemma 2. *For any $y \in [0,1]^A$, the distribution p ensures that for any $f^\star$*

$$\mathbb{E}_{a \sim p}[\max_{a^\star} f^\star(a^\star) - f^\star(a)] - \frac{\gamma}{4} \mathbb{E}_{a \sim p}(y(a) - f^\star(a))^2 \leq \frac{2A}{\gamma}$$

Proof. The basic idea is to add and subtract many terms and then handle each part on its own. Consider some $f^\star$ and let $a^\star = \mathrm{argmax}_a f^\star(a)$. Recall that $b = \mathrm{argmax}_a y(a)$. Then, looking at the "CB regret" term

$$\sum_a p(a)(f^\star(a^\star) - f^\star(a)) = \sum_{a \neq a^\star} p(a) \left[\underbrace{(f^\star(a^\star) - y(a^\star))}_{=:T_1} + \underbrace{(y(a^\star) - y(b))}_{=:T_2} + \underbrace{(y(b) - y(a))}_{=:T_3} + \underbrace{(y(a) - f^\star(a))}_{=:T_4} \right]$$

We bound T_1 and T_4 using the AM-GM inequality, which will lead to some cancellation with the square loss term. We will bound T_3 using the definition of p. For T_4 we will combine it with a remainder term in T_1 and use the definition of p to get the bound.

Recall that AM-GM can be used in the following way: $x = 2 \cdot \sqrt{1/\eta} \cdot x\sqrt{\eta/4} \leq \frac{1}{\eta} + \eta x^2/4$.

Bound on T_4. Using AM-GM we get

$$\sum_{a \neq a^\star} p(a)(y(a) - f^\star(a)) \leq \frac{(1 - p(a^\star))}{\gamma} + \frac{\gamma}{4} \sum_{a \neq a^\star} p(a)(y(a) - f^\star(a))^2.$$

We will drop $p(a^\star)$ in the first term and the second term cancels with things in the square loss term on the LHS.

Bound on T_1. Using AM-GM here, we get

$$(1 - p(a^\star))(f^\star(a^\star) - y(a^\star)) \leq \frac{(1 - p(a^\star))^2}{p(a^\star)\gamma} + \frac{\gamma}{4} p(a^\star)(f^\star(a^\star) - y(a^\star))^2$$

Again we'll drop the numerator on the first term and the second cancels what is left of the square loss term on the LHS. The challenge is that we have $p(a^\star)$ in the denominator so we need to argue that $p(a^\star)$ is not too small.

Bound on T_2 using residual from T_1. Combining T_3 and the remaining term from the T_1 bound, we have

$$(1 - p(a^\star))(y(a^\star) - y(b)) + \frac{1}{p(a^\star)\gamma} \leq (y(a^\star) - y(b)) + \frac{1}{p(a^\star)\gamma}$$

For this consider two cases. First if $a^\star = b$ then we claim that $p(b) = p(a^\star) \geq 1/A$, since $\forall a \neq b : p(a) = \frac{1}{A + \gamma(y(b) - y(a))} \leq 1/A$. In this case this quantity is at most A/γ. In the second case, if $a^\star \neq b$ then

$$y(a^\star) - y(b) + \frac{1}{p(a^\star)\gamma} = y(a^\star) - y(b) + \frac{A + \gamma(y(b) - y(a^\star))}{\gamma} = \frac{A}{\gamma}$$

Bound on T_3 using the allocation rule. Using the definition of $p(a)$ we get

$$\sum_{a \neq a^\star} p(a)(y(b) - y(a)) = \sum_{a \neq a^\star} \frac{1}{A + \gamma(y(b) - y(a))} \cdot (y(b) - y(a)) \leq \frac{A - 1}{\gamma}.$$

Combining everything proves the result. □

3 Contextual bandits with classification oracles

There is also a line of work on developing contextual bandit algorithms using the policy optimization oracle. The simplest of these is the ϵ-greedy strategy, which you may have heard of before. The idea for this algorithm is that we choose an exploration parameter ϵ and at each round, with probability ϵ we choose an action uniformly at random and with the remaining probability we act according to the empirically best policy so far. This algorithm is quite simple and, even though it achieves a suboptimal regret bound, it works quite well in practice for contextual bandit settings.

A rough approximation to this, which has a similar guarantee but more straightforward proof, is the "explore-first" algorithm. Here we pick a budget $N \leq T$ of exploration rounds and we act randomly for the first N rounds. Then we call the policy optimization oracle once to get a policy $\hat{\pi}$ which we use for the remaining $T - N$ rounds.

Roughly the regret analysis is as follows. If we collect N exploration samples, we will have N/A samples for each action so we will be able to prove that

$$\max_{\pi \in \Pi} \mathbb{E}_x[R(x, \pi(x))] - \mathbb{E}_x[R(x, \hat{\pi}(x))] \lesssim \sqrt{\frac{A \log(|\Pi|/\delta)}{N}}.$$

Then,

$$\text{Regret}(T) \lesssim N + (T - N)\sqrt{\frac{A \log(|\Pi|/\delta)}{N}} \lesssim N + T\sqrt{\frac{A \log(|\Pi|/\delta)}{N}} \leq O(T^{2/3}(A \log(|\Pi|/\delta))^{1/3}),$$

if we choose $N \asymp T^{2/3}(A \log |\Pi|)^{1/3}$. ϵ-greedy, which does not front-load all of the exploration, can also be shown to achieve a regret bound scaling as $T^{2/3}$. Unfortunately this is suboptimal, since $\sqrt{AT \log |\Pi|}$ is the optimal rate (and SquareCB achieves this optimal rate, with realizability).

It is possible to get this optimal regret rate using classification oracle-based algorithms, but (a) the analysis is quite complicated, and (b) SquareCB tends to outperform these methods in practice. As such, we will not discuss these methods in the course.

A Fast rates for square loss

To understand why the deviation term in (1) scales as $\log(1/\delta)$ rather than $\sqrt{T \log(1/\delta)}$, let us take a step back and consider the *offline* regression setting. Here we have a function class $\mathcal{F} : \mathcal{Z} \to [0, 1]$, a distribution $\mathcal{D} \in \Delta(\mathcal{Z})$, and a reward function $R : \mathcal{Z} \to \Delta([0, 1])$ such that $f^\star : z \mapsto \mathbb{E}[r \mid z]$ is in our function class $\mathcal{F}$. Given dataset $\{(z_i, r_i)\}_{i=1}^n$ we solve the square loss empirical risk minimization problem:

$$\hat{f} \leftarrow \operatorname*{argmin}_{f \in \mathcal{F}} \underbrace{\frac{1}{n} \sum_{i=1}^{n} (f(z_i) - r_i)^2}_{=: R_n(f)}$$

The square loss is nice because it has a convexity or self-bounding property, which enables faster rates of convergence. Indeed, we can prove the following.

Proposition 3. *With probability at least $1 - \delta$ we have*

$$\mathbb{E}[(\hat{f}(z) - f^\star(z))^2] \leq \frac{12 \log(2|\mathcal{F}|/\delta)}{n}.$$

Proof. We will use Bernstein's inequality (which you saw in the homework) and a union bound. Recall that Bernstein's inequality states that, with probability $1 - \delta$

$$\bar{X}_n - \mu \leq \sqrt{\frac{2\sigma^2 \log(1/\delta)}{n}} + \frac{2M \log(1/\delta)}{n},$$

where $\bar{X}_n = \frac{1}{n}\sum_i X_i$ is an average of n iid random variables with mean μ, variance σ^2 and range M. In our case, we will apply this inequality to the random variable $(f(z_i) - r_i)^2 - (f^\star(z_i) - r_i)^2$. We need to calculate/bound the mean, variance, and range. One crucial observation is that our random variable is a difference of squares.

$$\begin{aligned}
\mu(f) &:= \mathbb{E}\left[(f(z_i) - r_i)^2 - (f^\star(z_i) - r_i)^2\right] \\
&= \mathbb{E}\left[f(z_i)^2 - f^\star(z_i)^2 - 2r_i(f(z_i) - f^\star(z_i))\right] \\
&= \mathbb{E}\left[f(z_i)^2 - f^\star(z_i)^2 - 2f^\star(z_i)(f(z_i) - f^\star(z_i))\right] \\
&= \mathbb{E}\left[(f(z) - f^\star(z))^2\right] \\
\sigma^2(f) &:= \mathrm{Var}\left[(f(z_i) - r_i)^2 - (f^\star(z_i) - r_i)^2\right] \\
&\le \mathbb{E}\left[\left((f(z_i) - r_i)^2 - (f^\star(z_i) - r_i)^2\right)^2\right] \\
&= \mathbb{E}\left[((f(z_i) - f^\star(z_i))(f(z_i) + f^\star(z_i) - 2r_i))^2\right] \\
&\le 4\mathbb{E}\left[(f(z) - f^\star(z))^2\right] = 4\mu(f).
\end{aligned}$$

It is easily seen that since rewards and predictions are in $[0, 1]$ the random variable is in $[-1, 1]$, so we can take $M = 1$. Here observe that we have $\sigma^2(f) \le 4\mu(f)$, which is referred to as *self-bounding*. This is one property that allows us to obtain $1/n$ rates.

Now, Bernstein's inequality and a union bound reveals that, with probability $1 - \delta$:

$$\forall f \in \mathcal{F} : |R_n(f) - R_n(f^\star) - \mu(f)| \le \sqrt{\frac{8\mu(f)\log(2|\mathcal{F}|/\delta)}{n}} + \frac{2\log(2|\mathcal{F}|/\delta)}{n} \le \frac{1}{2}\mu(f) + \frac{6\log(2|\mathcal{F}|/\delta)}{n},$$

where the second inequality is by AM-GM (just like we saw previously). From here, we can deduce two things by re-arranging.

$$\begin{aligned}
\forall f \in \mathcal{F} : \hat{R}_n(f) - R_n(f^\star) &\le \frac{3}{2}\mathbb{E}\left[(f(z) - f^\star(z))^2\right] + \frac{6\log(2|\mathcal{F}|/\delta)}{n}, \\
\mathbb{E}\left[(\hat{f}(z) - f^\star(z))^2\right] &\le 2\left(\hat{R}_n(\hat{f}) - R_n(f^\star)\right) + \frac{12\log(2|\mathcal{F}|/\delta)}{n} \le \frac{12\log(2|\mathcal{F}|/\delta)}{n}.
\end{aligned}$$

□

To prove (1), we use roughly the same argument, with two modifications: (1) we need a martingale version since the random variables are not iid, and (2) we don't need to use a union bound over $|\mathcal{F}|$. Here the random variable in consideration is

$$(\hat{y}_t(x_t, a_t) - r_t)^2 - (f^\star(x_t, a_t) - r_t)^2.$$

Conditioning on everything before round t the expectation of this r.v. is $\mu_t := \sum_a p_t(a)(\hat{y}_t(x_t, a) - f^\star(x_t, a))^2$ (we saw this calculation previously), we can similarly show that the conditional variance $\sigma_t^2 \le 4\mu_t$, and the range is 1. So the martingale version of Bernstein's inequality, which is called Freedman's inequality shows that with probability at least $1 - \delta$:

$$\sum_t \mu_t - (\hat{y}_t(x_t, a_t) - r_t)^2 - (f^\star(x_t, a_t) - r_t)^2 \le \sqrt{8\sum_t \mu_t \log(1/\delta)} + 2\log(1/\delta) \le \frac{1}{2}\sum_t \mu_t + 6\log(1/\delta)$$

Re-arranging this inequality gives

$$\begin{aligned}
\sum_t \sum_a p_{t,a}(\hat{y}_{t,a} - f^\star(x_t, a))^2 = \sum_t \mu_t &\le 2\sum_t (\hat{y}_t(x_t, a_t) - r_t)^2 - (f^\star(x_t, a_t) - r_t)^2 + 12\log(1/\delta) \\
&\le 2\mathrm{Reg}_{\mathrm{Sq}}(T) + 12\log(1/\delta).
\end{aligned}$$