Lecture 19

Instructor: Haipeng Luo

1 Contextual Bandit

For the rest of the lectures we will focus on a generalization of the multi-armed bandit problem, called *contextual bandit*, which is still a very active research area and has also shown great practical potential recently. Specifically, the setting is the following: on each round $t = 1, \dots, T$,

1. the environment first decides a context-loss pair $(x_t, \ell_t) \in \mathcal{X} \times [0,1]^K$ for some arbitrary context space $\mathcal{X}$;
2. the environment reveals x_t to the learner, who then picks an action $a_t \in [K]$;
3. the learner suffers and observes $\ell_t(a_t)$.

So far it is not clear yet what the role of the contexts x_t's is. This is actually reflected in the regret, which is now defined as

$$\mathcal{R}_T = \sum_{t=1}^{T} \ell_t(a_t) - \operatorname*{argmin}_{\pi \in \Pi} \sum_{t=1}^{T} \ell_t(\pi(x_t))$$

where $\pi : \mathcal{X} \to [K]$ is called a *policy* and Π is a set of policies that is fixed and known to the learner ahead of time. In other words, instead of competing with the best fixed action as in multi-armed bandit, the goal in contextual bandit is to compete with the best fixed policy from a class, which could potentially pick different actions at different rounds based on the given context.

Contextual bandit is especially suitable to model problems such as personalized recommendation. Here, each time corresponds to a visit of a user. The context x_t can be seen as a feature vector of the user, capturing all the available contextual information such as gender, IP address, purchase history, and so on. An action is then one of the products/articles/movies to recommend to the user and the loss of the recommendation can be constructed based on whether it's clicked by the user or not.

So far this is pretty similar to the example we talked about in Lecture 15 for stochastic linear bandit. However, the key difference is that now we do not make assumptions on how the losses are related to the contexts. Instead, they are connected through the concept of policies, which could be some linear predictors, decision trees, neural nets, or really any kind of predictors used in typical machine learning problems. This greatly improves the generality and practicality of the model.

For simplicity, we assume that Π is finite but with a huge cardinality N. For example, think about Π as a set of decision trees with a fixed depth and a fixed number of possible decision rules on each node. Then N is exponentially large and it is prohibitive to have regret (or running time) that is polynomial in N.

The multi-armed problem can be seen as a special case of contextual bandit, where there are only K policies in Π and each of them commits to a fixed (and different) action independent of the context input. It is clear that in this case the regret simply degenerates to the usual regret defined for the multi-armed bandit setting.

However, the connection goes even deeper. In general one can simply see this as an N-armed bandit problem where each policy is an arm. Picking an arm at time t naturally amounts to picking the action suggested by this policy under the current context x_t. At the end of the round we indeed observe the loss of the selected policy for this round. This suggests a trivial way of solving contextual bandit with a multi-armed bandit algorithm, but it clearly leads to a regret of order $\mathcal{O}(\sqrt{TN})$, independent of K but polynomial in N, which is not acceptable.

However, all is not lost. It turns out that simply using Exp3 algorithm with a natural loss estimator will solve the problem. In fact, we have seen similar phenomenon in the analysis of Exp2 already. Specifically, let $P_t \in \Delta(N)$ be such that for all $\pi \in \Pi$,[1]

$$P_t(\pi) \propto \exp\left(-\eta \sum_{\tau=1}^{t-1} \widehat{\ell}_\tau(\pi(x_\tau))\right),$$

for some estimated loss vector $\widehat{\ell}_\tau$. We use the notation $P_t(\cdot|x) \in \Delta(K)$ to denote the distribution over actions induced by P_t on a context x, such that for all $a \in [K]$,

$$P_t(a|x) = \sum_{\pi\in\Pi:\pi(x)=a} P_t(\pi),$$

which is exactly the probability of picking action a if we randomly select a policy according to P_t and then follow the suggestion of the selected policy. Finally the algorithm simply picks $a_t \sim P_t(\cdot|x_t)$ and construct the usual loss estimator $\widehat{\ell}_t(a) = \frac{\ell_t(a)}{P_t(a|x_t)}\mathbf{1}\{a = a_t\}$.

This algorithm is called Exp4 [Auer et al., 2002], which stands for "Exponential-weight algorithm for Exploration and Exploitation using Expert advice" (originally policies are called experts). It is straightforward to show the following regret bound for Exp4 (assuming oblivious environments again):

Theorem 1. *With $\eta = \sqrt{\frac{\ln N}{TK}}$, Exp4 ensures $\mathbb{E}[\mathcal{R}_T] = \mathcal{O}(\sqrt{TK \ln N})$.*

Proof. The proof is again based on the following adaptive regret bound of Hedge: for any $\pi^\star$,

$$\sum_{t=1}^T \sum_{\pi\in\Pi} P_t(\pi)\widehat{\ell}_t(\pi(x_t)) - \sum_{t=1}^T \widehat{\ell}_t(\pi^\star(x_t)) \le \frac{\ln N}{\eta} + \eta \sum_{t=1}^T \sum_{\pi\in\Pi} P_t(\pi)\widehat{\ell}_t(\pi(x_t))^2.$$

Note that as before we have $\mathbb{E}_{a_t}\left[\widehat{\ell}_t(\pi(x_t))\right] = \ell_t(\pi(x_t))$ and $\mathbb{E}_{a_t}\left[\widehat{\ell}_t(\pi(x_t))^2\right] \le \frac{1}{P_t(\pi(x_t)|x_t)}$. Therefore, the last term of the above regret bound can be bounded as:

$$\mathbb{E}_{a_t}\left[\sum_{\pi\in\Pi} P_t(\pi)\widehat{\ell}_t(\pi(x_t))^2\right] \le \sum_{\pi\in\Pi} \frac{P_t(\pi)}{P_t(\pi(x_t)|x_t)} = \sum_{a=1}^K \sum_{\pi:\pi(x_t)=a} \frac{P_t(\pi)}{P_t(a|x_t)} = K.$$

Finally realizing $\sum_{\pi\in\Pi} P_t(\pi)\widehat{\ell}_t(\pi(x_t)) = \sum_{\pi:\pi(x_t)=a_t} P_t(\pi)\frac{\ell_t(a_t)}{P_t(a_t|x_t)} = \ell_t(a_t)$, taking expectation on both sides, and using the (optimal) choice of η finish the proof. □

This regret bound has only logarithmic dependence on N and is in fact almost optimal [Seldin and Lugosi, 2016]. However, it is also clear that the algorithm is computationally inefficient since it needs to maintain weights for each policy and thus has time complexity $\mathcal{O}(N)$ per round.

2 Oracle-efficient Algorithms

One of the main research directions in contextual bandit is to get around the computational obstacle so that one can actually apply contextual bandit in practice. Without any additional structures or assumptions of the problem, this appears to be impossible. Starting from the work of [Langford and Zhang, 2008], many existing works study efficient contextual bandit algorithms under a specific computational model where an offline optimization oracle is given. Specifically, an optimization oracle, denoted by ERM (stands for Empirical Risk Minimization), takes a set $\mathcal{S}$ of context-loss pairs $(x, \ell) \in \mathcal{X} \times \mathbb{R}^K$ as inputs, and outputs

$$\text{ERM}(\mathcal{S}) = \operatorname*{argmin}_{\pi\in\Pi} \sum_{(x,\ell)\in\mathcal{S}} \ell(\pi(x)),$$

[1] We switch to notation P_t since p_t has been used for a distribution over actions previously.

which is the policy with the smallest loss on the input dataset. An algorithm is called *oracle-efficient* if its running time and number of oracle queries are both polynomial in T, K and $\ln N$ (excluding the running time of the oracle itself). Naively the oracle can be implemented in $O(N)$ time, but the point is exactly that we assume we are given a "smart" oracle that is somehow much more efficient.

The justification of this computational model is two-fold. From a theoretical viewpoint, since the oracle is simply computing the benchmark (that is, the second term) in the definition of regret, the question of whether oracle-efficient algorithm exists is essentially asking whether offline optimization and online optimization are computationally equivalent, which seems to be a very natural question.

Moreover, from a practical viewpoint, the optimization oracle is essentially solving a supervised learning problem (specifically a "cost-sensitive classification" problem), which has been heuristically solved in practice by various algorithms already. In other words, this computational model allows one to reduce the contextual bandit problem to a well-studied supervised learning problem, and to reuse any existing packages from an engineering perspective. As a result, any advances in solving the offline problem practically will also directly lead to advances for the contextual bandit problem.

Somewhat surprisingly, it has been shown that oracle-efficient algorithm does not exist in general when the environment is adversarial [Hazan and Koren, 2016] (even under full information setting), and therefore there is indeed a gap between offline and online optimization. However, with additional assumptions, oracle-efficiency becomes possible. We will focus on one of these assumptions for the rest of the lecture, which simply states that the pairs $(x_1, \ell_1), \ldots, (x_T, \ell_T)$ are i.i.d. samples of an arbitrary and unknown joint distribution $\mathcal{D}$.

In such an i.i.d. setting, we denote the expected loss of a policy by $\bar{\ell}(\pi) = \mathbb{E}_{(x,\ell)\sim\mathcal{D}}[\ell(\pi(x))]$ and the policy with the smallest expected loss by $\pi^\star = \operatorname{argmin}_{\pi\in\Pi} \bar{\ell}(\pi)$. Since $\pi^\star$ will have very similar performance compared to the empirically best policy $\pi' = \operatorname{argmin}_{\pi\in\Pi} \sum_{t=1}^{T} \ell_t(\pi)$ due to Hoeffding's inequality and union bound: with probability $1-\delta$,

$$\bar{\ell}(\pi^\star) \le \bar{\ell}(\pi') \le \frac{1}{T}\sum_{t=1}^{T} \ell_t(\pi') + \mathcal{O}\left(\sqrt{\frac{\ln(N/\delta)}{T}}\right),$$

we redefine the regret as $\mathcal{R}_T = \sum_{t=1}^{T} \left(\ell_t(a_t) - \bar{\ell}(\pi^\star)\right)$, which is away from the original definition by only a (non-dominant) term $\mathcal{O}(\sqrt{T\ln(N/\delta)})$.

2.1 Warm-up: Full Information

To get a sense on why oracle-efficiency is possible, we start with a full information setting, that is, the entire loss vector ℓ_t is revealed instead of just $\ell_t(a_t)$ at the end of round t. In this case, one can simply follow the leader: query the oracle to get $\pi_t = \text{ERM}(\{(x_1, \ell_1), \ldots, (x_{t-1}, \ell_{t-1})\})$ and then play $a_t = \pi_t(x_t)$. This is clearly an oracle-efficient algorithm (in fact, the number of oracle queries can even be substantially reduced).

Theorem 2. *FTL ensures $\mathcal{R}_T = \widetilde{\mathcal{O}}\left(\sqrt{T\ln(N/\delta)}\right)$ with probability $1-\delta$ in the full information i.i.d. setting.*[2]

Proof. By Azuma's inequality we have with probability $1-\delta/2$,

$$\sum_{t=1}^{T} \ell_t(a_t) \le \sum_{t=1}^{T} \mathbb{E}_{x_t,\ell_t}[\ell_t(a_t)] + \mathcal{O}\left(\sqrt{T\ln(1/\delta)}\right) = \sum_{t=1}^{T} \bar{\ell}(\pi_t) + \mathcal{O}\left(\sqrt{T\ln(1/\delta)}\right).$$

Define $\bar{\ell}_t(\pi) = \frac{1}{t}\sum_{\tau=1}^{t} \ell_\tau(\pi(x_\tau))$ to be the empirical average loss of π up to time t. By Hoeffding's inequality and union bound we have with probability $1-\delta/2$, for all $t \in [T]$ and all $\pi \in \Pi$,

$$|\bar{\ell}_t(\pi) - \bar{\ell}(\pi)| \le \mathcal{O}\left(\sqrt{\frac{\ln(TN/\delta)}{t}}\right).$$

[2] Notation $\widetilde{\mathcal{O}}(\cdot)$ hides dependence that is logarithmic in T, K, and $\ln N$.

Therefore by the optimality of π_t, we have for $t > 1$,

$$\bar{\ell}(\pi_t) \le \bar{\ell}_{t-1}(\pi_t) + \mathcal{O}\left(\sqrt{\frac{\ln(TN/\delta)}{t-1}}\right) \le \bar{\ell}_{t-1}(\pi^\star) + \mathcal{O}\left(\sqrt{\frac{\ln(TN/\delta)}{t-1}}\right) \le \bar{\ell}(\pi^\star) + \mathcal{O}\left(\sqrt{\frac{\ln(TN/\delta)}{t-1}}\right).$$

Combining everything, we have with probability $1-\delta$,

$$\mathcal{R}_T = \sum_{t=1}^{T} \left(\ell_t(a_t) - \bar{\ell}(\pi_t) + \bar{\ell}(\pi_t) - \bar{\ell}(\pi^\star)\right) = \mathcal{O}\left(\sqrt{T\ln(TN/\delta)}\right),$$

completing the proof. □

2.2 First Attempt for Bandit

Moving on to the bandit setting, we again need to deal with the exploration-exploitation dilemma. The simplest extension of FTL is to uniformly explore the K actions with certain probability. Specifically, let $\pi_t = \text{ERM}(\{(x_1, \widehat{\ell}_1), \ldots, (x_{t-1}, \widehat{\ell}_{t-1})\})$ and $p_t \in \Delta(K)$ be such that $p_t(a) = (1 - K\mu)\{a = \pi_t(x_t)\} + \mu$ for some $\mu \le 1/K$. Pick action $a_t \sim p_t$ and construct estimator $\widehat{\ell}_t(a) = \frac{\ell_t(a)}{p_t(a)}\mathbf{1}\{a = a_t\}$.

This algorithm is called Epsilon-Greedy, and is clearly also an oracle-efficient algorithm. However, it achieves a suboptimal regret as shown in the next theorem. In the next two lectures we will eventually improve the regret to almost optimal.

Theorem 3. *In the i.i.d. contextual bandit setting, with the optimal tuning of μ Epsilon-Greedy ensures $\mathcal{R}_T = \widetilde{\mathcal{O}}\left(T^{\frac{2}{3}}(K\ln(N/\delta))^{\frac{1}{3}} + \sqrt{TK\ln(N/\delta)} + K\ln(N/\delta)\right)$ with probability $1-\delta$.*

Proof. By Azuma's inequality we have with probability $1-\delta/2$,

$$\sum_{t=1}^{T} \ell_t(a_t) \le \sum_{t=1}^{T} \mathbb{E}_{x_t,\ell_t,a_t}[\ell_t(a_t)] + \mathcal{O}\left(\sqrt{T\ln(1/\delta)}\right) = \sum_{t=1}^{T} \bar{\ell}(\pi_t) + TK\mu + \mathcal{O}\left(\sqrt{T\ln(1/\delta)}\right).$$

Redefine $\bar{\ell}_t(\pi) = \frac{1}{t}\sum_{\tau=1}^{t} \widehat{\ell}_\tau(\pi(x_\tau))$ to be the empirical average estimated loss of π up to time t. While one can apply Azuma's inequality (and union bound) to show that with probability $1-\delta/2$, for all $t \in [T]$ and all $\pi \in \Pi$,

$$|\bar{\ell}_t(\pi) - \bar{\ell}(\pi)| \le \mathcal{O}\left(\frac{1}{\mu}\sqrt{\frac{\ln(TN/\delta)}{t}}\right),$$

this will in fact only lead to a regret of order $\mathcal{O}(T^{\frac{3}{4}})$. Instead we will apply a tighter inequality called Freedman's inequality (see Lemma 1). Note that

$$\mathbb{E}_{x_t,\ell_t,a_t}\left[\left(\widehat{\ell}_t(\pi(x_t)) - \bar{\ell}(\pi)\right)^2\right] \le \mathbb{E}_{x_t,\ell_t,a_t}\left[\widehat{\ell}_t(\pi(x_t))^2\right] \le \mathbb{E}_{x_t}\left[\frac{1}{p_t(\pi(x_t))}\right] \le \frac{1}{\mu}.$$

Applying Freedman's inequality we thus have with probability $1-\delta/2$, for all $t \in [T]$ and all $\pi \in \Pi$,

$$|\bar{\ell}_t(\pi) - \bar{\ell}(\pi)| \le \mathcal{O}\left(\sqrt{\frac{\ln(TN/\delta)}{\mu t}} + \frac{\ln(TN/\delta)}{\mu t}\right),$$

Therefore similarly by the optimality of π_t, we have for $t > 1$,

$$\bar{\ell}(\pi_t) \le \bar{\ell}(\pi^\star) + \mathcal{O}\left(\sqrt{\frac{\ln(TN/\delta)}{\mu(t-1)}} + \frac{\ln(TN/\delta)}{\mu(t-1)}\right),$$

Combining everything, we have with probability $1-\delta$,

$$\mathcal{R}_T = \sum_{t=1}^{T} \left(\ell_t(a_t) - \bar{\ell}(\pi_t) + \bar{\ell}(\pi_t) - \bar{\ell}(\pi^\star)\right) = \mathcal{O}\left(TK\mu + \sqrt{\frac{T\ln(TN/\delta)}{\mu}} + \frac{\ln(TN/\delta)\ln T}{\mu}\right).$$

Picking the optimal μ completes the proof. □

Lemma 1 (Freedman's inequality)**.** *Let* $X_1, \ldots, X_T \in [-B, B]$ *for some* $B > 0$ *be a martingale difference sequence and with* $\sum_{t=1}^{T} \mathbb{E}_t[X_t^2] \leq V$ *for some fixed quantity* V*. We have for all* $\delta \in (0, 1)$*, with probability* $1 - \delta$*,*

$$\sum_{t=1}^{T} X_t \leq \min_{\lambda \in [0,1/B]} \left(\lambda V + \frac{\ln(1/\delta)}{\lambda} \right) \leq 2\sqrt{V \ln(1/\delta)} + B \ln(1/\delta).$$

References

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

Elad Hazan and Tomer Koren. The computational power of optimization in online learning. In *48th Annual ACM Symposium on the Theory of Computing*, 2016.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems 21*, 2008.

Yevgeny Seldin and Gábor Lugosi. A lower bound for multi-armed bandits with expert advice. In *13th European Workshop on Reinforcement Learning (EWRL)*, 2016.