Lecture 1

Instructor: Haipeng Luo

1 Online Learning: Examples and Models

Below is a list of examples of online learning:

- spam detection (online classification/regression): At each time $t = 1, 2, \ldots$
  - receive an email $x_t \in \mathbb{R}^d$;
  - predict whether it is a spam $\hat{y}_t \in \{-1, +1\}$;
  - see its true label $y_t \in \{-1, +1\}$.
- sequential investment (universal portfolio): Start with capital W_1. At each day $t = 1, 2, \ldots$
  - decide $p_t \in \Delta(N) \stackrel{\text{def}}{=} \{p \in \mathbb{R}_+^N : \sum_{i=1}^N p(i) = 1\}$ and invest $W_t p_t(i)$ on asset i;
  - at the end of the day observe relative prices $r_t \in \mathbb{R}^N$ and arrive at total capital $W_{t+1} = W_t \langle p_t, r_t \rangle$.
- aggregating weather prediction (the expert problem): At each day $t = 1, 2, \ldots$
  - obtain temperature predictions from N experts/models;
  - make the final prediction by randomly following an expert according to $p_t \in \Delta(N)$;
  - on the next day observe the loss of each model $\ell_t \in [0, 1]^N$.
- product recommendation (multi-armed bandits): At each time $t = 1, 2, \ldots$
  - randomly recommend one of the K products a to a customer visiting the website;
  - observe the loss of this product $\ell_t(a)$ (e.g. 0 if clicked, 1 otherwise), but not the losses for the other products.
- multiple-product recommendation (combinatorial bandits): At each time $t = 1, 2, \ldots$
  - randomly recommend k of the K products to a customer visiting the website;
  - observe the losses of the k recommended products but not the other ones.
- personalized product recommendation (contextual bandits): Given N policies $\pi^1, \ldots, \pi^N$, each of which is a mapping from $\mathcal{X}$ to $[K]$. At each time $t = 1, 2, \ldots$
  - observe the contextual information $x_t \in \mathcal{X}$ of a customer (e.g. gender, IP address, purchase history, etc);
  - randomly select one of the N policies π_t and recommend product $\pi_t(x_t)$;
  - observe the loss of this product $\ell_t(\pi_t(x_t))$ but not the other ones.

All of these problems can be (essentially) captured by a learning model called *Online Convex Optimization (OCO)*, first proposed by Zinkevich [2003]. OCO can be viewed as a game between a learner/player and an environment/adversary. Before the game starts, a fixed compact convex set Ω is given to the player as the action space. The game then proceeds for T rounds for some integer T. At each round $t = 1, \ldots, T$,

1. the player first picks a point $w_t \in \Omega$;
2. the environment then picks a convex loss function $f_t : \Omega \to [0, 1]$;
3. the player suffers loss $f_t(w_t)$, and observes some information about f_t.

Depending on the power of the environment, there are several possible settings:

- stochastic setting: $f_1, \ldots, f_T$ are i.i.d samples of a fixed distribution;
- oblivious adversary: $f_1, \ldots, f_T$ are arbitrary, but decided before the game starts (i.e. independent of the player's actions);
- non-oblivious/adaptive adversary: for each t, f_t depends on $w_1, \ldots, w_t$.

Depending on the feedback model, there are also several possible settings:

- full information setting: player observes f_t (or sometimes just (sub)gradient $\nabla f_t(w_t)$);
- bandit setting: player observes only $f_t(w_t)$;
- other partial information settings.

The table below summarizes how OCO captures different kinds of online learning problems.

Problems	Ω	f_t
linear classification linear regression	e.g. $\{w : \|w\|_2 \le 1\}$	$f_t(w) = \ell(\langle w, x_t \rangle, y_t)$, e.g. logistic loss: $\ell(\hat{y}, y) = \ln(1 + e^{-\hat{y}y})$ or square loss: $\ell(\hat{y}, y) = (\hat{y} - y)^2$
universal portfolio	$\Delta(N)$	$f_t(p) = -\ln(\langle p, r_t \rangle)$ (note the unboundedness)
the expert problem	$\Delta(N)$	$f_t(p) = \langle p, \ell_t \rangle$
multi-armed bandits	$\Delta(K)$	$f_t(p) = \langle p, \ell_t \rangle$ (note the feedback model)
combinatorial bandits	$\left\{w = \sum_{j=1}^{M} p(j) v_j \mid p \in \Delta(M)\right\}$ for some $v_1, \ldots, v_M \in \{0,1\}^K$. e.g. $\{w \in [0,1]^K : \sum_{i=1}^{K} w(i) = k\}$	$f_t(w) = \langle w, \ell_t \rangle$ (note the feedback model)
contextual bandits	$\Delta(N)$	$f_t(w) = \sum_{j=1}^{N} w(j) \ell_t(\pi^j(x_t))$ (note the feedback model and the loss structure among policies)

The classic goal of OCO is to minimize the player's regret against the best fixed action in hindsight:

$$\mathcal{R}_T = \sum_{t=1}^{T} f_t(w_t) - \min_{w \in \Omega} \sum_{t=1}^{T} f_t(w).$$

If $\mathcal{R}_T = o(T)$, then it implies that $\lim_{T \to \infty} \mathcal{R}(T)/T = 0$ and thus on average the player is doing almost as well as the best fixed action in hindsight, which is a pretty strong guarantee. Beside minimizing this regret measurement, there are many more harder objectives in OCO that we will cover later.

2 Connection to Statistical Learning

Statistical learning is a classic learning model. Here we explore the connections and differences between statistical learning and online learning.

In statistical learning, a set of training examples $z_1, \ldots, z_T \in \mathcal{Z}$ is given to the learner where each example z_t is an i.i.d. sample of some unknown distribution $\mathcal{D}$. Based on these training examples, the learner outputs an action $w(z_1, \ldots, z_T) \in \Omega$ for some compact convex set Ω. For some loss function $\ell : \Omega \times \mathcal{Z} \to [0,1]$, the training error of the learner is defined as $\frac{1}{T} \sum_{t=1}^{T} \ell(w(z_1, \ldots, z_T), z_t)$ while the generalization error is defined as $\mathbb{E}_{z \sim \mathcal{D}} \ell(w(z_1, \ldots, z_T), z)$. Note that the generalization error is a random variable with respect to the randomness of the training set, and the goal of the learner is to have small generalization error with high probability.

As one can see, distributional assumptions are built in the definition of statistical learning. On the other hand, online learning does not necessarily assume that data is from some fixed distribution, which makes it much more suitable for dealing with time-varying environments. In fact, even if the data is entirely adversarial, which is indeed the case for applications such as spam detection, meaningful and strong guarantees can still be derived for online learning as we will see soon.

Another key advantage is that online learning algorithms are usually more memory-efficient, in the sense that they usually do not need to store data from the past. That is, at each round, the new data is used to update the current states of the algorithm and then discarded. On the other hand, most statistical learning algorithms require storing the training set and touching each example multiple times.

Moreover, it can in fact be shown that online learning is strictly harder than statistical learning in the sense that a full information online learning algorithm can be used to solve statistical learning. The reduction is as follows [Cesa-Bianchi et al., 2004]:

Algorithm 1: Online-to-batch reduction

Input: training set $\{z_1, \dots, z_T\}$, an online learning algorithm $\mathcal{A}$ with action space Ω
for $t = 1, \dots, T$ **do**
  let w_t be the output of $\mathcal{A}$ for this round
  feed $\mathcal{A}$ with loss function $f_t(w) = \ell(w, z_t)$
Output: $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$.

One can show the following:

Theorem 1. *If $w \to \mathbb{E}_{z\sim\mathcal{D}}\,\ell(w, z)$ is convex, then with probability at least $1-\delta$, the generalization error of the output of Algorithm 1 satisfies*

$$\mathbb{E}_{z\sim\mathcal{D}}\,\ell(\bar{w}, z) \le \mathbb{E}_{z\sim\mathcal{D}}\,\ell(w^\star, z) + \frac{\mathcal{R}_T}{T} + 2\sqrt{\frac{2\ln(2/\delta)}{T}}$$

where $w^\star \in \operatorname{argmin}_{w\in\Omega} \mathbb{E}_{z\sim\mathcal{D}}\,\ell(w, z)$ and $\mathcal{R}_T$ is the regret of the $\mathcal{A}$ after T rounds.

To prove the theorem, we first state the following two important concentration results in probability theory which will be used extensively in the rest of the course.

Lemma 1 (Hoeffding's inequality)**.** *Let $X_1, \dots, X_T \in [-B, B]$ for some $B > 0$ be independent random variables such that $\mathbb{E}[X_t] = 0$ for all $t \in [T]$, then we have for all $\delta \in (0, 1)$,*

$$\Pr\left(\sum_{t=1}^{T} X_t \ge B\sqrt{2T\ln\tfrac{1}{\delta}}\right) \le \delta.$$

Lemma 2 (Azuma's inequality)**.** *Let $X_1, \dots, X_T \in [-B, B]$ for some $B > 0$ be a martingale difference sequence (i.e. $\mathbb{E}[X_t | X_{t-1}, \dots, X_1] = 0$ for all $t \in [T]$), then we have for all $\epsilon > 0$,*

$$\Pr\left(\sum_{t=1}^{T} X_t \ge B\sqrt{2T\ln\tfrac{1}{\delta}}\right) \le \delta.$$

Proof of Theorem 1. With probability at least $1-\delta$, we have

$$\begin{aligned}
\mathbb{E}_{z\sim\mathcal{D}}\,\ell(\bar{w}, z) &= \mathbb{E}_{z\sim\mathcal{D}}\,\ell\left(\frac{1}{T}\sum_{t=1}^{T} w_t, z\right) \\
&\le \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{z\sim\mathcal{D}}\,\ell(w_t, z) && \text{(Jensen's inequality)} \\
&= \frac{1}{T}\sum_{t=1}^{T} \ell(w_t, z_t) + \sqrt{\frac{2\ln(2/\delta)}{T}} \\
& && \text{(Azuma's inequality with } X_t = \mathbb{E}_{z\sim\mathcal{D}}\,\ell(w_t, z) - \ell(w_t, z_t))
\end{aligned}$$

$$
\begin{aligned}
&= \min_{w \in \Omega} \frac{1}{T} \sum_{t=1}^{T} \ell(w, z_t) + \frac{\mathcal{R}_T}{T} + \sqrt{\frac{2\ln(2/\delta)}{T}} && \text{(by definition of regret)} \\
&\leq \frac{1}{T} \sum_{t=1}^{T} \ell(w^\star, z_t) + \frac{\mathcal{R}_T}{T} + \sqrt{\frac{2\ln(2/\delta)}{T}} \\
&\leq \mathbb{E}_{z \sim \mathcal{D}}\, \ell(w^\star, z) + \frac{\mathcal{R}_T}{T} + 2\sqrt{\frac{2\ln(2/\delta)}{T}}, \\
& && \text{(Hoeffding's inequality with } X_t = \ell(w^\star, z_t) - \mathbb{E}_{z \sim \mathcal{D}}\, \ell(w^\star, z))
\end{aligned}
$$

which completes the proof. □

For many problems we will show that $\mathcal{R}_T = \mathcal{O}(\sqrt{T})$ and therefore the online-to-batch approach provides a convergence rate of $1/\sqrt{T}$ for the generalization error, which is known to be optimal for many cases.

References

Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, 2003.

Lecture 2

Instructor: Haipeng Luo

1 Hedge

The expert problem [Freund and Schapire, 1997] mentioned in last lecture turns out to play a fundamental role in online learning, and we will focus on this problem for a couple of lectures. The first native algorithm that comes to one's mind is probably the *follow the leader* (FTL) approach, which puts all the weights to the current best expert $i_t^\star = \operatorname{argmax}_{i\in[N]} - \sum_{\tau=1}^{t-1} \ell_s(i)$. It is not hard to see that such an approach does not work well in general, at least not in the adversarial setting.

It turns out, however, that simply replacing the "max" by some "softmax" would change to situation greatly. In fact, this leads to the the classic algorithm Hedge [Freund and Schapire, 1997] (generalizing [Littlestone and Warmuth, 1994]), also known as multiplicative weights update or simply exponential weights. Below is the pseudocode.

Algorithm 1: Hedge

Input: learning rate $\eta > 0$
Initialization: let $L_0 \in R^N$ be the all-zero vector
for $t = 1, \ldots, T$ **do**
  compute $p_t \in \Delta(N)$ such that $p_t(i) \propto \exp(-\eta L_{t-1}(i))$
  play p_t and observe loss vector $\ell_t \in [0,1]^N$
  update $L_t = L_{t-1} + \ell_t$

Recall the definition of regret for this setting:

$$\mathcal{R}_T = \sum_{t=1}^T \langle p_t, \ell_t \rangle - \min_{p\in\Delta(N)} \sum_{t=1}^T \langle p, \ell_t \rangle = \sum_{t=1}^T \langle p_t, \ell_t \rangle - \sum_{t=1}^T \ell_t(i^\star)$$

where $i^\star \in \operatorname{argmin}_i \sum_{t=1}^T \ell_t(i)$ is the best expert in hindsight. Hedge guarantees the following regret bound:

Theorem 1. *Hedge with learning rate η guarantees*

$$\mathcal{R}_T \le \frac{\ln N}{\eta} + \eta \sum_{t=1}^T \sum_{i=1}^N p_t(i)\ell_t^2(i) \tag{1}$$

$$\le \frac{\ln N}{\eta} + T\eta, \tag{2}$$

which is of order $\mathcal{O}(\sqrt{T \ln N})$ if η is optimally set to $\sqrt{(\ln N)/T}$.

There are many different proofs that lead to bound (2). Here we present a "potential-based" proof that obtains bound (1) as an intermediate step, which will turn out to be very useful later on.

Proof. Let $\Phi_t = \frac{1}{\eta} \ln\left(\sum_{i=1}^N \exp(-\eta L_t(i))\right)$. First note that

$$\Phi_t - \Phi_{t-1} = \frac{1}{\eta} \ln\left(\frac{\sum_{i=1}^N \exp(-\eta L_t(i))}{\sum_{i=1}^N \exp(-\eta L_{t-1}(i))}\right)$$

$$
\begin{aligned}
&= \frac{1}{\eta} \ln \left(\sum_{i=1}^{N} p_t(i) \exp(-\eta \ell_t(i)) \right) \\
&\leq \frac{1}{\eta} \ln \left(\sum_{i=1}^{N} p_t(i) \left(1 - \eta \ell_t(i) + \eta^2 \ell_t^2(i)\right) \right) && (e^{-y} \leq 1 - y + y^2 \text{ for all } y \geq 0) \\
&= \frac{1}{\eta} \ln \left(1 - \eta \left\langle p_t, \ell_t \right\rangle + \eta^2 \sum_{i=1}^{N} p_t(i) \ell_t^2(i) \right) \\
&\leq - \left\langle p_t, \ell_t \right\rangle + \eta \sum_{i=1}^{N} p_t(i) \ell_t^2(i). && (\ln(1+y) \leq y)
\end{aligned}
$$

Summing over t, telescoping and rearranging give

$$
\begin{aligned}
\sum_{t=1}^{T} \left\langle p_t, \ell_t \right\rangle &\leq \Phi_0 - \Phi_T + \eta \sum_{t=1}^{T} \sum_{i=1}^{N} p_t(i) \ell_t^2(i) \\
&\leq \frac{\ln N}{\eta} - \frac{1}{\eta} \ln \left(\exp(-\eta L_T(i^\star)) \right) + \eta \sum_{t=1}^{T} \sum_{i=1}^{N} p_t(i) \ell_t^2(i) \\
&\leq \frac{\ln N}{\eta} + L_T(i^\star) + \eta \sum_{t=1}^{T} \sum_{i=1}^{N} p_t(i) \ell_t^2(i),
\end{aligned}
$$

which proves Eq. (1). Eq. (2) follows immediately by the boundedness of losses. □

Note that the regret of Hedge has only logarithmic dependence on N, which as we will see is very useful in solving many problems with huge number of experts.

2 Lower bound for the Expert Problem

Is the regret bound of Hedge good or bad? In general, how can we tell whether a regret upper bound is satisfactory or not? The notion of minimax regret can be used to answer these questions exactly. Intuitively, minimax regret is the smallest possible worst-case regret of any algorithm. For example, the minimax regret of the expert problem can be defined as

$$
\min_{\mathcal{A}} \max_{\ell_1, \ldots, \ell_T} \mathcal{R}_T
$$

where $\mathcal{A}$ is any legitimate expert algorithm. Note that $\mathcal{R}_T$ depends on both $\mathcal{A}$ and all the losses even if the dependence is not explicitly spelled out. Also note that in general $\mathcal{R}_T$ should be viewed as the expected regret if the algorithm is randomized. The existence of the Hedge algorithm already shows that

$$
\min_{\mathcal{A}} \max_{\ell_1, \ldots, \ell_T} \mathcal{R}_T \leq 2\sqrt{T \ln N}.
$$

The following theorem proves that this bound is minimax optimal (up to a constant of $2\sqrt{2}$). In the proof we use an implicit and probabilistic construction of the environment, which is a very useful technique in proving lower bounds.

Theorem 2. *For any algorithm, we have*

$$
\sup_{T, N} \max_{\ell_1, \ldots, \ell_T} \frac{\mathcal{R}_T}{\sqrt{T \ln N}} \geq \frac{1}{\sqrt{2}}.
$$

Proof. Let $\mathcal{D}$ be the uniform distribution over $\{0, 1\}$. We have

$$\begin{aligned}
\max_{\ell_1,\ldots,\ell_T} \mathcal{R}_T &\geq \mathbb{E}_{\ell_1,\ldots,\ell_T \overset{iid}{\sim} \mathcal{D}^N} [\mathcal{R}_T] \\
&= \sum_{t=1}^{T} \mathbb{E}_{\ell_1,\ldots,\ell_{t-1}} \mathbb{E}_{\ell_t}[\langle p_t, \ell_t \rangle \,|\ell_{t-1}, \ldots, \ell_1] - \mathbb{E}_{\ell_1,\ldots,\ell_T} \left[\min_{i\in[N]} \sum_{t=1}^{T} \ell_t(i) \right] \\
&= \sum_{t=1}^{T} \mathbb{E}_{\ell_1,\ldots,\ell_{t-1}} \langle p_t, \mathbb{E}_{\ell_t}[\ell_t|\ell_{t-1}, \ldots, \ell_1] \rangle - \mathbb{E}_{\ell_1,\ldots,\ell_T} \left[\min_{i\in[N]} \sum_{t=1}^{T} \ell_t(i) \right] \\
&= T/2 - \mathbb{E}_{\ell_1,\ldots,\ell_T} \left[\min_{i\in[N]} \sum_{t=1}^{T} \ell_t(i) \right] \\
&= \mathbb{E}_{\ell_1,\ldots,\ell_T} \left[\max_{i\in[N]} \sum_{t=1}^{T} (\tfrac{1}{2} - \ell_t(i)) \right] \\
&= \frac{1}{2} \mathbb{E}_{\sigma_1,\ldots,\sigma_T} \left[\max_{i\in[N]} \sum_{t=1}^{T} \sigma_t(i) \right],
\end{aligned}$$

where $\sigma_t(i)$ for $i \in [N], t \in [T]$ are i.i.d. Rademacher random variables (i.e. -1 with probability 0.5 and 1 with probability 0.5). Using the following result from probability theory (see for example [Cesa-Bianchi and Lugosi, 2006, Chapter 3.7]) completes the proof.

$$\lim_{T\to\infty} \lim_{N\to\infty} \frac{\mathbb{E}_{\sigma_1,\ldots,\sigma_T} \left[\max_{i\in[N]} \sum_{t=1}^{T} \sigma_t(i) \right]}{\sqrt{T \ln N}} = \sqrt{2}.$$

□

3 Follow the Regularized Leader

Hedge is just one classic example of online learning. For a general OCO problem, how do we design low-regret algorithms? There are in fact several general frameworks to do this. Here we explore one of them, called *Follow the Regularized Leader* (FTRL).

To introduce FTRL, first recall the FTL algorithm for OCO: $w_t = \operatorname{argmin}_{w\in\Omega} \sum_{\tau=1}^{t-1} f_\tau(w)$. As mentioned, this is not a good algorithm generally. However, if we could cheat and play w_{t+1} at time t (which requires the knowledge of f_t), how small would the regret be? This invaild algorithm is often called *Be the Leader* (BTL) and the following lemma shows that it in fact has negative regret.

Lemma 1 (BTL lemma)**.** *If $w_t = \operatorname{argmin}_{w\in\Omega} \sum_{\tau=1}^{t-1} f_\tau(w)$, then*

$$\sum_{t=1}^{T} f_t(w_{t+1}) - \min_{w\in\Omega} \sum_{t=1}^{T} f_t(w) \leq 0.$$

Proof. By definition and optimality of w_t, we have

$$\begin{aligned}
\sum_{t=1}^{T} f_t(w_{t+1}) - \min_{w\in\Omega} \sum_{t=1}^{T} f_t(w) &= \sum_{t=1}^{T} f_t(w_{t+1}) - \sum_{t=1}^{T} f_t(w_{T+1}) \\
&= \sum_{t=1}^{T-1} f_t(w_{t+1}) - \sum_{t=1}^{T-1} f_t(w_{T+1}) \\
&\leq \sum_{t=1}^{T-1} f_t(w_{t+1}) - \sum_{t=1}^{T-1} f_t(w_T) \\
&= \sum_{t=1}^{T-2} f_t(w_{t+1}) - \sum_{t=1}^{T-2} f_t(w_T) \\
&\leq \ldots \leq f_1(w_2) - f_1(w_3) \leq 0.
\end{aligned}$$

□

Therefore, the regret of FTL can be bounded by:

$$\sum_{t=1}^{T} f_t(w_t) - \min_{w\in\Omega} \sum_{t=1}^{T} f_t(w) \leq \sum_{t=1}^{T} \left(f_t(w_t) - f_t(w_{t+1})\right),$$

which means the regret is controlled by how close w_t and w_{t+1} are, or in other words, how stable the algorithm is. One way to see that FTL is not a low-regret algorithm is exactly by arguing that it is not stable. Therefore, to fix this issue, we should think about how to improve the stability of the algorithm.

Regularization, a widely-used technique in machine learning, turns out to be also extremely useful here in terms of stabilizing the algorithms. Specifically, FTRL plays at round t:

$$w_t = \operatorname*{argmin}_{w\in\Omega} \sum_{\tau=1}^{t-1} f_\tau(w) + \frac{1}{\eta}\psi(w) \tag{3}$$

where $\eta > 0$ is some learning rate parameter to be specified and $\psi : \Omega \to \mathbb{R}$ is the regularizer. To ensure stability, the regularizer ψ needs to be *strongly convex*, which means for any $w, u \in \Omega$, the following holds[1]

$$\psi(w) - \psi(u) \leq \langle \nabla\psi(w), w - u\rangle - \tfrac{1}{2}\|w - u\|^2$$

for some norm $\|\cdot\|$. The next lemma shows that FTRL is stable and the level of stability is controlled by the parameter η.

Lemma 2 (Stability of FTRL)**.** *The FTRL strategy* (3) *ensures*

$$f_t(w_t) - f_t(w_{t+1}) \leq \eta \|\nabla f_t(w_t)\|_\star^2,$$

where $\|\cdot\|_\star$ is the dual norm of $\|\cdot\|$.[2]

Proof. Let $F_t(w) = \sum_{\tau=1}^{t} f_\tau(w) + \frac{1}{\eta}\psi(w)$. By strong convexity of ψ, one can verify

$$F_{t-1}(w_t) - F_{t-1}(w_{t+1}) \leq \langle \nabla F_{t-1}(w_t), w_t - w_{t+1}\rangle - \frac{1}{2\eta}\|w_t - w_{t+1}\|^2.$$

Since $w_t = \operatorname{argmin}_w F_{t-1}(w)$, first order optimality condition implies $\langle \nabla F_{t-1}(w_t), w_t - w_{t+1}\rangle \leq 0$ and thus

$$F_{t-1}(w_t) - F_{t-1}(w_{t+1}) \leq -\frac{1}{2\eta}\|w_t - w_{t+1}\|^2.$$

By the same argument, we have

$$F_t(w_{t+1}) - F_t(w_t) \leq \langle \nabla F_t(w_{t+1}), w_{t+1} - w_t\rangle - \frac{1}{2\eta}\|w_t - w_{t+1}\|^2 \leq -\frac{1}{2\eta}\|w_t - w_{t+1}\|^2.$$

Summing up the above two inequalities and rearranging give

$$\|w_t - w_{t+1}\|^2 \leq \eta(f_t(w_t) - f_t(w_{t+1})).$$

Finally by convexity and Hölder's inequality we have

$$\begin{aligned} f_t(w_t) - f_t(w_{t+1}) &\leq \langle \nabla f_t(w_t), w_t - w_{t+1}\rangle \leq \|\nabla f_t(w_t)\|_\star \|w_t - w_{t+1}\| \\ &\leq \|\nabla f_t(w_t)\|_\star \sqrt{\eta(f_t(w_t) - f_t(w_{t+1}))}, \end{aligned}$$

and solving for $f_t(w_t) - f_t(w_{t+1})$ finishes the proof. □

With this stability lemma, we can show the following regret bound for FTRL.

[1] More precisely, this is the definition of ψ being 1-strongly convex.

[2] The definition of dual norm is $\|u\|_\star = \max_{\|w\|\leq 1} \langle u, w\rangle$.

Theorem 3. *With parameter η FTRL ensures,*

$$\mathcal{R}_T \leq \frac{D}{\eta} + \eta \sum_{t=1}^{T} \|\nabla f_t(w_t)\|_\star^2 ,$$

where $D = \max_{w\in\Omega} \psi(w) - \min_{w\in\Omega} \psi(w)$. If we further have $\|\nabla f_t(w)\|_\star \leq G$ for all $w \in \Omega$, then setting $\eta = \sqrt{\frac{D}{TG^2}}$ leads to $\mathcal{R}_T = \mathcal{O}(G\sqrt{TD})$.

Proof. Define $f_0(w) = \frac{\psi(w)}{\eta}$ so that $w_t = \operatorname{argmin}_{w\in\Omega} \sum_{\tau=0}^{t-1} f_\tau(w)$. By the BTL lemma, we have for $w^\star = \operatorname{argmin}_{w\in\Omega} \sum_{t=1}^{T} f_t(w)$,

$$\sum_{t=0}^{T} f_t(w_{t+1}) - \sum_{t=0}^{T} f_t(w^\star) \leq 0.$$

Therefore, the regret of FTRL is

$$\begin{aligned}
\mathcal{R}_T &= \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(w^\star) \\
&\leq \sum_{t=1}^{T} f_t(w_t) - \sum_{t=0}^{T} f_t(w_{t+1}) + f_0(w^\star) \\
&= f_0(w^\star) - f_0(w_1) + \sum_{t=1}^{T} (f_t(w_t) - f_t(w_{t+1})) \\
&\leq \frac{D}{\eta} + \eta \sum_{t=1}^{T} \|\nabla f_t(w_t)\|_\star^2 ,
\end{aligned}$$

where the last step if by the definition of D and the stability lemma. □

References

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.

Lecture 3

Instructor: Haipeng Luo

1 Instances of FTRL

In the last lecture we study FTRL, a general online learning algorithm, and prove the following regret bound

$$\mathcal{R}_T \leq \frac{D}{\eta} + \eta \sum_{t=1}^{T} \|\nabla f_t(w_t)\|_\star^2,$$

where $D = \max_{w\in\Omega} \psi(w) - \min_{w\in\Omega} \psi(w)$. Here we present two concrete instances of FTRL.

1.1 Recovering Hedge

Recall that in the expert problem $\Omega = \Delta(N)$ and $f_t(p) = \langle p, \ell_t \rangle$. If we pick the (negative) entropy function as the regularizer, that is, $\psi(p) = \sum_{i=1}^{N} p(i) \ln p(i)$, one can verify that the solution of

$$p_t = \operatorname*{argmin}_{p\in\Delta(N)} \left\langle p, \sum_{\tau=1}^{t-1} \ell_t \right\rangle + \frac{1}{\eta} \sum_{i=1}^{N} p(i) \ln p(i)$$

is exactly the Hedge algorithm, that is, $p_t(i) \propto \exp(-\eta \sum_{\tau=1}^{t-1} \ell_t(i))$. In other words, Hedge is just one special case of FTRL.

To apply the FTRL regret bound, we use the fact that the entropy function is strongly convex with respect to the L_1 norm. To see this note that in this case the definition of strong convexity

$$\psi(p) - \psi(q) \leq \langle \nabla\psi(p), p - q \rangle - \tfrac{1}{2} \|p - q\|_1^2$$

is equivalent to

$$\frac{1}{2} \|p - q\|_1^2 \leq \sum_{i=1}^{N} q(i) \ln \frac{q(i)}{p(i)} \stackrel{\text{def}}{=} \mathrm{KL}(q, p).$$

The latter turns out to be exactly the Pinsker's inequality, which states that the Kullback-Leibler divergence of two distributions is lower bounded by half of their L_1 distance square.

Also notice that the dual norm of the L_1 norm is the L_∞ norm and by boundedness of losses we have $\|\nabla f_t(p_t)\|_\infty = \|\ell_t\|_\infty \leq 1$. Moreover, the (negative) entropy function has maximal value 0 (when the distribution concentrates on one coordinate) and minimum value $-\ln N$ (when the distribution is uniform), and thus $D = \ln N$. Therefore, applying the FTRL regret bound we again arrive at

$$\mathcal{R}_T \leq \frac{\ln N}{\eta} + T\eta,$$

the same bound we proved last time using a different potential-based argument.

1.2 Online Gradient Descent

In the next example we consider an arbitrary OCO problem and pick $\psi(w) = \frac{1}{2} \|w\|_2^2$. The FTRL algorithm becomes

$$w_t = \operatorname*{argmin}_{w\in\Omega} \sum_{\tau=1}^{t-1} f_\tau(w) + \frac{1}{2\eta} \|w\|_2^2.$$

One can (approximately) solve this convex optimization problem using standard methods. However, it turns out that it is without loss of generality to assume that f_t is a linear function. To see this, note that by convexity the regret can be bounded as

$$\mathcal{R}_T = \max_{w\in\Omega} \sum_{t=1}^{T} \left(f_t(w_t) - f_t(w)\right) \le \max_{w\in\Omega} \sum_{t=1}^{T} \langle \nabla f_t(w_t), w_t - w\rangle .$$

Therefore, we can imagine that the loss function is actually a linear function $f'_t(w) = \langle f_t(w_t), w\rangle$, and a regret bound for this linear problem is clearly also a regret bound for the original problem. With this reduction, we rewrite the above FTRL as

$$w_t = \operatorname*{argmin}_{w\in\Omega} \left\langle w, \sum_{\tau=1}^{t-1} \nabla f_\tau(w_\tau) \right\rangle + \frac{1}{2\eta} \|w\|_2^2 . = \operatorname*{argmin}_{w\in\Omega} \left\| w + \eta \sum_{\tau=1}^{t-1} \nabla f_\tau(w_\tau) \right\|_2^2 ,$$

which means w_t is the L_2 projection of $u_t = -\eta \sum_{\tau=1}^{t-1} \nabla f_\tau(w_\tau)$ onto Ω. This algorithm is called *Online Gradient Descent* (OGD) [Zinkevich, 2003]. To see the connection to the regular gradient descent, note that OGD can be equivalently written as

$$u_{t+1} = u_t - \eta \nabla f_t(w_t); \quad w_{t+1} = \operatorname*{argmin}_{w\in\Omega} \|w - u_{t+1}\| ,$$

while regular gradient descent would instead do

$$u_{t+1} = w_t - \eta \nabla f_t(w_t); \quad w_{t+1} = \operatorname*{argmin}_{w\in\Omega} \|w - u_{t+1}\| .$$

In fact, there is little real difference between these two algorithms and one can prove the same guarantee for both of them. Below we apply the general FTRL guarantee to prove a regret bound.

Indeed, one can easily verify that $\psi(w) = \frac{1}{2}\|w\|_2^2$ is strongly convex with respect to the L_2 norm. Note that the dual norm of the L_2 norm is itself. So if we let G be an upper bound on all the gradients, that is, $\|\nabla f_t(w_t)\|_2 \le G$, then the regret of OGD is bounded by

$$\mathcal{R}_T \le \frac{\max_{w\in\Omega} \|w\|_2^2}{2\eta} + \eta T G^2 = \mathcal{O}\left(\max_{w\in\Omega} \|w\|_2 G\sqrt{T} \right),$$

where the last step is by picking the optimal η.

Examples Consider the online regression problem where $\Omega = \{w \in \mathbb{R}^d : \|w\|_2 \le 1\}$ is a set of linear predictors with bounded norm, and $f_t(w) = \frac{1}{2}(\langle w, x_t\rangle - y_t)^2$ is the square loss for an example $x_t \in \{x \in \mathbb{R}^d : \|x\|_2 \le 1\}$ and its label $y_t \in [-1, 1]$. Then because $\nabla f_t(w) = (\langle w, x_t\rangle - y_t)x_t$, we have $G = 2$ and $\max_{w\in\Omega} \|w\|_2 = 1$, and therefore OGD has regret $\mathcal{O}(\sqrt{T})$, independent of the dimension of the problem d.

Next consider using OGD for the expert problem. Note that for the simplex, $\max_{p\in\Delta(N)} \|p\|_2 \le \max_{p\in\Delta(N)} \|p\|_1 = 1$, but $\|\ell_t\|_2 \le \sqrt{N}$. Thus OGD's regret is $\mathcal{O}(\sqrt{TN})$ in this case, which has an exponentially worse dependence on N compared to Hedge.

2 Follow the Perturbed Leader and Combinatorial Problems

As we have seen, FTRL uses regularization to stabilize the algorithm. Here, we introduce another very different approach, *perturbation*. To motivate this approach, we consider the following online combinatorial problems.

Let $S = \{v_1, \ldots, v_M\}$ be a set of combinatorial actions such that $v_j \in \{0, 1\}^N$ and $\|v_j\|_1 \le m$ for some integer $m \le N$ and all $j \in [M]$. The decision space for the learner is the convex hull of S, that is, $\Omega = \left\{ \sum_{j=1}^{M} p(j) v_j : p \in \Delta(M) \right\}$. Thus, each point in Ω specifies a distribution over these combinatorial actions or in other words a randomized strategy. We consider linear loss functions so that $f_t(w) = \langle w, \ell_t\rangle$ for some $\ell_t \in [0, 1]^N$. Finally, for simplicity we restrict our attention to oblivious environments so that $\ell_1, \ldots, \ell_T$ are decided before the game starts.

The expert problem is clearly a special case where S consists of all the standard basis vectors in $\mathbb{R}^N$ and $m = 1$. Another example is when $S = \{v \in \{0,1\}^N : \|v\|_1 = m\}$ so that $\Omega = \{w \in [0,1]^N : \|w\|_1 = m\}$ (recall the multiple-product recommendation example in Lecture 1).

Yet another important example is the online shortest path problem. In this problem, a direct acyclic graph with N edges, a source vertex, and a destination vertex is given. Each round the player first randomly picks a path, then the loss (e.g. delay) for each edge is revealed and the player suffers the total loss of all the edges on the picked path. This can be formulated as a special case of the above combinatorial problem by setting S to be the set of all paths starting from the source and ending at the destination (that is, a path is represented by a vector in $\{0,1\}^N$ so that each coordinate indicates whether the corresponding edge is on the path or not). Note that m is the length of the longest path in S.

One can again use the FTRL approach to tackle this problem, but it is not often clear how to solve the optimization problem in FTRL. Instead, we consider a different approach called *Follow the Perturbed Leader* (FTPL) [Kalai and Vempala, 2005], which only requires a linear optimization step over Ω. Specifically, let ℓ_0 be a uniformly random draw from $[0, 1/\eta]^N$ for some $\eta > 0$. Then at round t FTPL plays

$$w_t = \operatorname*{argmin}_{w \in \Omega} \left\langle w, \sum_{\tau=0}^{t-1} \ell_\tau \right\rangle.$$

In other words, w_t is the leader according the cumulative losses plus some perturbation ℓ_0. Note that this leader can always be some point in S due to linearity. Moreover, for many problems this linear optimization admits efficient algorithm. For example, for the expert problem this is trivially solved by picking the best coordinate. For the online shortest path problem, this can be solved by a shortest path algorithm such as Dijkstra's algorithm.

It remains to prove the regret bound of FTPL. To this end, we first show that perturbation provides stability in expectation.

Lemma 1 (Stability of FTPL)**.** *FTPL with parameter η ensures that*

$$\mathbb{E}[f_t(w_t) - f_t(w_{t+1})] \le mN\eta$$

where the expectation is with respect to the random draw of ℓ_0.

Proof. To make the dependence explicit, define $h_t(\ell_0) = \left\langle \operatorname{argmin}_{w \in \Omega} \sum_{\tau=0}^{t-1} \ell_\tau, \ell_t \right\rangle$. We then have

$$\begin{aligned}
\mathbb{E}[f_t(w_t) - f_t(w_{t+1})] &= \mathbb{E}[h_t(\ell_0) - h_t(\ell_0 + \ell_t)] \\
&= \eta^N \int_{\ell_0 \in [0,1/\eta]^N} h_t(\ell_0) - h_t(\ell_0 + \ell_t) d\ell_0 \\
&= \eta^N \left(\int_{\ell_0 \in [0,1/\eta]^N} h_t(\ell_0) d\ell_0 - \int_{\ell_0 \in \ell_t + [0,1/\eta]^N} h_t(\ell_0) d\ell_0 \right) \\
&\qquad \text{(change of variable)} \\
&\le \eta^N \int_{\ell_0 \in [0,1/\eta]^N \setminus \ell_t + [0,1/\eta]^N} h_t(\ell_0) d\ell_0 \\
&\le m\eta^N \int_{\ell_0 \in [0,1/\eta]^N \setminus \ell_t + [0,1/\eta]^N} d\ell_0 \qquad (h_t(\ell_0) \le m) \\
&= m \Pr\left(\exists i : \ell_0(i) \le \ell_t(i)\right) \\
&\le m \sum_{i=1}^N \Pr\left(\ell_0(i) \le 1\right) \qquad \text{(union bound and } \ell_t(i) \le 1) \\
&= mN\eta.
\end{aligned}$$

□

With the stability lemma, we can prove the following bound using similar argument as in FTRL.

Theorem 1. *FTPL with parameter η ensures that*

$$\mathbb{E}[\mathcal{R}_T] \leq \frac{m}{2\eta} + mNT\eta$$

where the expectation is with respect to the random draw of ℓ_0. With η optimally set to $\sqrt{1/(2NT)}$ we thus have $\mathbb{E}[\mathcal{R}_T] = \mathcal{O}(m\sqrt{TN})$.

Proof. Note that by the BTL lemma, with $w^\star = \operatorname{argmin}_{w\in\Omega} \sum_{t=1}^T f_t(w)$, we again have

$$\begin{aligned}
\mathcal{R}_T &= \sum_{t=1}^T f_t(w_t) - \sum_{t=1}^T f_t(w^\star) \\
&\leq \sum_{t=1}^T f_t(w_t) - \sum_{t=0}^T f_t(w_{t+1}) + f_0(w^\star) \\
&= f_0(w^\star) - f_0(w_1) + \sum_{t=1}^T (f_t(w_t) - f_t(w_{t+1})).
\end{aligned}$$

Applying the stability lemma and realizing $\mathbb{E}[f_0(w^\star) - f_0(w_1)] \leq \mathbb{E}[\langle \ell_0, w^\star \rangle] = \langle \mathbb{E}[\ell_0], w^\star \rangle \leq \frac{m}{2\eta}$ finish the proof. □

Note that the bound is suboptimal in general. For example, in the expert problem it has polynomial dependence on N instead of logarithmic dependence. However, with a more sophisticated noise distribution (rather than uniform), the regret bounds can often be improved to be optimal. In fact, it is well-known that in the expert problem, FTPL with Gumbel noise is *equivalent* to Hedge!

A final remark is that to deal with non-oblivious environments, it turns out that one only needs to draw a fresh sample of ℓ_0 at the beginning of each round. The intuition is that this will prevent a non-oblivious environment, whose strategy can depend on the player's actions, from figuring out ℓ_0 gradually. A formal proof can be found in [Hutter and Poland, 2005].

References

Marcus Hutter and Jan Poland. Adaptive online prediction by following the perturbed leader. *Journal of Machine Learning Research*, 6(Apr):639–660, 2005.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, 2003.

Lecture 4

Instructor: Haipeng Luo

In the following lectures, we focus on the expert problem and study more *adaptive* algorithms. Although Hedge is proven to be worst-case optimal, one may wonder how well it would actually perform when dealing with a practical problem that is probably not the worst case or even relatively easy. Indeed, the regret bound we proved for Hedge only says that for all problem instances, Hedge's regret is uniformly bounded by $\mathcal{O}(\sqrt{T \ln N})$. However, ideally we want to have an algorithm that enjoys a much smaller regret in many easy situations, but in the worst case still guarantees the minimax regret $\mathcal{O}(\sqrt{T \ln N})$. Deriving adaptive algorithms and adaptive regret bounds is exactly one way to achieve this goal.

1 "Small-loss" Bounds

We start with the arguably simplest adaptive bound, sometimes called "small-loss" bound or first order bound. Recall that we proved the following intermediate bound for Hedge:

$$\mathcal{R}_T = \widetilde{L}_T - L_T(i^\star) \leq \frac{\ln N}{\eta} + \eta \sum_{t=1}^{T} \sum_{i=1}^{N} p_t(i) \ell_t^2(i),$$

where L_T is the cumulative loss vector, $i^\star$ is the best expert and we define $\widetilde{L}_T = \sum_{t=1}^{T} \langle p_t, \ell_t \rangle$ to be the cumulative loss of the algorithm. By boundedness of losses the last term above can be bounded by $\eta \widetilde{L}_T$. If $\eta < 1$, then rearranging gives

$$\mathcal{R}_T \leq \frac{1}{1-\eta} \left(\frac{\ln N}{\eta} + \eta L_T(i^\star) \right).$$

Therefore, if for a moment we assume we knew the quantity $L_T(i^\star)$ ahead of time and was able to set $\eta = \min \left\{ \frac{1}{2}, \sqrt{\frac{\ln N}{L_T(i^\star)}} \right\}$, then we arrive at

$$\mathcal{R}_T \leq 2 \left(\max \left\{ 2 \ln N, \frac{\ln N}{\sqrt{(\ln N)/L_T(i^\star)}} \right\} + \sqrt{\frac{\ln N}{L_T(i^\star)}} L_T(i^\star) \right) = \mathcal{O}\left(\sqrt{L_T(i^\star) \ln N} + \ln N \right).$$

The final bound above is the so-called "small-loss" bound, which essentially replaces the dependence on T in the minimax bound $\sqrt{T \ln N}$ by the loss of the best expert $L_T(i^\star)$. Note that $L_T(i^\star)$ is bounded by T, therefore the "small-loss" bound is not worse than the minimax bound. More importantly, it can be much smaller than T when the best expert indeed suffers very small loss. In particular, if the best expert makes no mistakes at all and have $L_T(i^\star) = 0$, then the "small-loss" bound is only $\mathcal{O}(\ln N)$, independent of T. This is one typical example of adaptive bounds that we are aiming for.

Of course, one obvious issue in the above derivation is that the learning rate has to be set in terms of the unknown quantity $L_T(i^\star)$. In fact, this becomes an even more severe problem in a non-oblivious environment since $L_T(i^\star)$ can depend on the algorithm's actions and thus η, making the definition of η circular.

Fortunately, there are many different ways to address this issue, and we explore one of them here. The idea is to use a more adaptive and time-varying learning rate schedule. Specifically, the algo-

rithm predicts $p_t(i) \propto \exp(-\eta_t L_{t-1}(i))$ where

$$\eta_t = \sqrt{\frac{\ln N}{\widetilde{L}_{t-1} + 1}}. \tag{1}$$

Note that $\widetilde{L}_{t-1} = \sum_{\tau=1}^{t-1} \langle p_\tau, \ell_\tau \rangle$ is the cumulative loss of the algorithm up to round $t-1$ and is thus available at the beginning of round t. This is sometimes called a "self-confident" learning rate since the algorithm is confident that its loss is close to the loss of the best expert and thus uses it as a proxy for the loss of the best expert to tune the learning rate. We next prove that this algorithm indeed provides a "small-loss" bound.

Theorem 1. *Hedge with adaptive learning rate schedule* (1) *ensures*

$$\mathcal{R}_T \leq 3\sqrt{(L_T(i^\star) + 1) \ln N} + 9 \ln N.$$

Proof. Let $\Phi_t(\eta) = \frac{1}{\eta} \ln \left(\frac{1}{N} \sum_{i=1}^N \exp(-\eta L_t(i)) \right)$. In Lecture 2 we already proved

$$\Phi_t(\eta_t) - \Phi_{t-1}(\eta_t) \leq - \langle p_t, \ell_t \rangle + \eta_t \sum_{i=1}^N p_t(i) \ell_t^2(i).$$

Summing over t and rearranging give

$$\begin{aligned}
\widetilde{L}_T &\leq \Phi_0(\eta_1) - \Phi_T(\eta_T) + \sum_{t=1}^T \eta_t \sum_{i=1}^N p_t(i) \ell_t^2(i) + \sum_{t=1}^{T-1} (\Phi_t(\eta_{t+1}) - \Phi_t(\eta_t)) \\
&\leq \frac{\ln N}{\eta_T} - \frac{1}{\eta_T} \ln \left(\exp(-\eta_T L_T(i^\star)) \right) + \sum_{t=1}^T \eta_t \sum_{i=1}^N p_t(i) \ell_t(i) + \sum_{t=1}^{T-1} (\Phi_t(\eta_{t+1}) - \Phi_t(\eta_t)) \\
&= \sqrt{(\widetilde{L}_{T-1} + 1) \ln N} + L_T(i^\star) + \sum_{t=1}^T \eta_t \langle p_t, \ell_t \rangle + \sum_{t=1}^{T-1} (\Phi_t(\eta_{t+1}) - \Phi_t(\eta_t)).
\end{aligned}$$

To bound the term $\sum_{t=1}^T \eta_t \langle p_t, \ell_t \rangle$, note that

$$\begin{aligned}
\sum_{t=1}^T \frac{\langle p_t, \ell_t \rangle}{\sqrt{\widetilde{L}_{t-1} + 1}} &= \sum_{t=1}^T \frac{\widetilde{L}_t - \widetilde{L}_{t-1}}{\sqrt{\widetilde{L}_{t-1} + 1}} \\
&= \sum_{t=1}^T \frac{\widetilde{L}_t - \widetilde{L}_{t-1}}{\sqrt{\widetilde{L}_t + 1}} + \sum_{t=1}^T (\widetilde{L}_t - \widetilde{L}_{t-1}) \left(\frac{1}{\sqrt{\widetilde{L}_{t-1} + 1}} - \frac{1}{\sqrt{\widetilde{L}_t + 1}} \right) \\
&\leq \sum_{t=1}^T \frac{\widetilde{L}_t - \widetilde{L}_{t-1}}{\sqrt{\widetilde{L}_t + 1}} + \sum_{t=1}^T \left(\frac{1}{\sqrt{\widetilde{L}_{t-1} + 1}} - \frac{1}{\sqrt{\widetilde{L}_t + 1}} \right) \qquad (\widetilde{L}_t - \widetilde{L}_{t-1} \leq 1) \\
&\leq 1 + \sum_{t=1}^T \frac{\widetilde{L}_t - \widetilde{L}_{t-1}}{\sqrt{\widetilde{L}_t + 1}} \\
&\leq 1 + \int_{\widetilde{L}_0}^{\widetilde{L}_T} \frac{dx}{\sqrt{x+1}} \\
&\leq 2\sqrt{\widetilde{L}_T + 1},
\end{aligned}$$

and thus $\sum_{t=1}^T \eta_t \langle p_t, \ell_t \rangle \leq 2\sqrt{(\widetilde{L}_T + 1) \ln N}$.

To bound $\Phi_t(\eta_{t+1}) - \Phi_t(\eta_t)$, we prove that $\Phi_t(\eta)$ in increasing in η and thus $\Phi_t(\eta_{t+1}) \leq \Phi_t(\eta_t)$. It suffices to prove that the derivative is non-negative. Indeed, direct calculation shows that with

$p^\eta_{t+1}(i) \propto \exp(-\eta L_t(i))$,

$$\begin{aligned}
\eta^2 \Phi_t'(\eta) &= \eta^2 \left(-\frac{1}{\eta^2} \ln \left(\frac{1}{N} \sum_{i=1}^N \exp(-\eta L_t(i)) \right) - \frac{1}{\eta} \frac{\sum_{i=1}^N L_t(i) \exp(-\eta L_t(i))}{\sum_{i=1}^N \exp(-\eta L_t(i))} \right) \\
&= \ln N - \sum_{i=1}^N p^\eta_{t+1}(i) \left(\ln \left(\sum_{j=1}^N \exp(-\eta L_t(j)) \right) + \eta L_t(i) \right) \\
&= \ln N - \sum_{i=1}^N p^\eta_{t+1}(i) \ln \left(\frac{\sum_{j=1}^N \exp(-\eta L_t(j))}{\exp(-\eta L_t(i))} \right) \\
&= \ln N - \sum_{i=1}^N p^\eta_{t+1}(i) \ln \frac{1}{p^\eta_{t+1}(i)} \geq 0,
\end{aligned}$$

where the last step is by the fact that entropy is maximized by the uniform distribution. To sum up, we have proven that

$$\mathcal{R}_T = \widetilde{L}_T - L_T(i^\star) \leq 3\sqrt{(\widetilde{L}_T + 1) \ln N}.$$

Solving for $\sqrt{\widetilde{L}_T + 1}$ leads to

$$\sqrt{\widetilde{L}_T + 1} \leq \frac{3}{2}\sqrt{\ln N} + \sqrt{L_T(i^\star) + 1 + \frac{9}{4} \ln N}.$$

Finally squaring both sides and using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ give

$$\widetilde{L}_T \leq 9 \ln N + L_T(i^\star) + 3\sqrt{(L_T(i^\star) + 1) \ln N},$$

which completes the proof. □

Besides enjoying a better theoretical regret bound, this algorithm is also intuitively more reasonable since it tunes the learning rate adaptively based on observed data. In general, learning rate tuning is an important topic in machine learning and could be of great practical importance.

2 Quantile Bounds

"Small-loss" bounds improve the dependence on T in the minimax regret bound to $L_T(i^\star)$. Is it possible to improve the other term $\ln N$ in the minimax bound to something better? To answer this question, consider again Hedge with a fixed learning rate for simplicity, and note that we proved in Lecture 2,

$$\widetilde{L}_T \leq \frac{\ln N}{\eta} - \frac{1}{\eta} \ln \left(\sum_{i=1}^N \exp(-\eta L_T(i)) \right) + T\eta.$$

Without loss of generality, assume $L_T(1) \leq \cdots \leq L_T(N)$ so that expert i is the i-th best expert. Previously we obtained the final regret bound by lower bounding $\sum_{i=1}^N \exp(-\eta L_T(i))$ by $\max_i \exp(-\eta L_T(i)) = \exp(-\eta L_T(1))$. In general, however, for each i we have

$$\sum_{j=1}^N \exp(-\eta L_T(j)) \geq \sum_{j=1}^i \exp(-\eta L_T(j)) \geq i \exp(-\eta L_T(i)),$$

and we therefore have the following regret bound against the i-th best expert:

$$\widetilde{L}_T - L_T(i) \leq \frac{\ln \left(\frac{N}{i}\right)}{\eta} + T\eta. \tag{2}$$

With η optimally tuned to $\sqrt{\ln \left(\frac{N}{i}\right) / T}$, the bound becomes $2\sqrt{T \ln \left(\frac{N}{i}\right)}$. This is called the quantile bound and it states that the algorithm suffers at most this amount of regret for all but i/N fraction of

the experts. Of course, at the end of the day what we care about is actually the loss of the algorithm. So assuming we had the knowledge of L_T for a moment, then we could pick the optimal η to achieve

$$\widetilde{L}_T \leq \min_{i\in[N]} \left(L_T(i) + 2\sqrt{T \ln\left(\tfrac{N}{i}\right)} \right), \tag{3}$$

which is a strictly better bound compared to $L_T(1) + 2\sqrt{T \ln N}$. To understand the improvement, consider the case when N is huge but there are many similar experts so that for example the top 1% of them all have the same cumulative loss. Then bound (3) is at most

$$L_T(1\% \times N) + 2\sqrt{T \ln\left(\tfrac{N}{1\%\times N}\right)} = L_T(1) + 2\sqrt{T \ln(100)},$$

which is independent of N.

Just as in the previous discussion, one obvious issue in the derivation of bound (3) above is again that the learning rate η needs to be tuned based on unknown knowledge. To address the issue, here we explore a quite different approach. The idea is to have different instances of Hedge running with different learning rates, and have a "master" Hedge to combine the predictions of these "meta-experts". To this end, we use Hedge(η) to denote an instance of Hedge running with learning rate η. The algorithm is shown below.

Algorithm 1: Hedge with Quantile Bounds

Input: master learning rate $\eta > 0$, base learning rates $\eta_1, \ldots, \eta_M$
Initialize: M Hedge algorithms Hedge(η_1), …, Hedge(η_M), $C_0(j) = 0$ for all $j \in [M]$
for $t = 1, \ldots, T$ **do**
- let p_t^j be the prediction of Hedge(η_j) on round t
- compute $p_t = \sum_{j=1}^M q_t(j) p_t^j$ where $q_t(j) \propto \exp(-\eta C_{t-1}(j))$
- play p_t and observe loss vector $\ell_t \in [0,1]^N$
- update $C_t(j) = C_{t-1}(j) + \left\langle p_t^j, \ell_t \right\rangle$ for all $j \in [M]$
- pass ℓ_t to Hedge(η_1), …, Hedge(η_M).

By Eq. (2), we have for each Hedge(η_j) and each expert i

$$\sum_{t=1}^T \left\langle p_t^j, \ell_t \right\rangle - L_T(i) \leq \frac{\ln\left(\frac{N}{i}\right)}{\eta_j} + T\eta_j.$$

On the other hand, for the master Hedge, we have for each meta-expert j,

$$\sum_{t=1}^T \sum_{j=1}^M q_t(j) \left\langle p_t^j, \ell_t \right\rangle - C_T(j) \leq \frac{\ln M}{\eta} + T\eta.$$

Note that by construction, we have $\sum_{j=1}^M q_t(j) \left\langle p_t^j, \ell_t \right\rangle = \langle p_t, \ell_t \rangle$ and $C_T(j) = \sum_{t=1}^T \left\langle p_t^j, \ell_t \right\rangle$. Therefore summing up the above two inequalities lead to

$$\sum_{t=1}^T \langle p_t, \ell_t \rangle - L_T(i) \leq \frac{\ln\left(\frac{N}{i}\right)}{\eta_j} + T\eta_j + \frac{\ln M}{\eta} + T\eta = \frac{\ln\left(\frac{N}{i}\right)}{\eta_j} + T\eta_j + 2\sqrt{T \ln M},$$

where the last step is by picking the optimal $\eta = \sqrt{\ln M / T}$. Note that the above holds for all j and all i. Therefore, suppose we have (a) for each i, there is an η_j such that $\frac{1}{\eta_j} \ln\left(\frac{N}{i}\right) + T\eta_j = \mathcal{O}\left(\sqrt{T \ln\left(\frac{N}{i}\right)}\right)$, and (b) M is much smaller than N, then we obtain bound (3).

Setting $M = N$ and $\eta_j = \sqrt{\ln(\frac{N}{j})/T}$ would clearly satisfy (a), but not (b). Fortunately, it turns out that one only needs to create $M \approx \ln N$ meta-experts and still satisfy (a). Specifically, let

$$\eta_j = \frac{1}{2^{j-1}} \sqrt{\frac{\ln N}{T}} \quad \text{and} \quad M = \left\lceil \log_2 \sqrt{\frac{\ln N}{\ln(\frac{N}{N-1})}} \right\rceil + 1.$$

Now clearly for each $i \neq N$, there exist a j such that $\eta_j \leq \sqrt{\ln\left(\frac{N}{i}\right)/T} \leq 2\eta_j$ and therefore

$$\begin{aligned}\sum_{t=1}^{T} \langle p_t, \ell_t \rangle - L_T(i) &\leq \frac{\ln\left(\frac{N}{i}\right)}{\eta_j} + T\eta_j + 2\sqrt{T \ln M} \\ &\leq \frac{\ln\left(\frac{N}{i}\right)}{\frac{1}{2}\sqrt{\ln\left(\frac{N}{i}\right)/T}} + T\sqrt{\ln\left(\frac{N}{i}\right)/T} + 2\sqrt{T \ln M} \\ &= 3\sqrt{T \ln\left(\tfrac{N}{i}\right)} + 2\sqrt{T \ln M}.\end{aligned}$$

It remains to show that M is small enough. Indeed, since $\ln(1+x) \geq x/2, \ \forall x \leq 1$, we have

$$\ln\left(\frac{N}{N-1}\right) = \ln\left(1 + \frac{1}{N-1}\right) \geq \frac{1}{2(N-1)},$$

and therefore $M = \mathcal{O}(\ln(N \ln N))$. So as least for the case when N/i is larger than $\mathcal{O}(\ln(N \ln N))$, the term $\sqrt{T \ln M}$ is dominated by $\sqrt{T \ln\left(\frac{N}{i}\right)}$ in the regret bound. We summarize the result in the following theorem.

Theorem 2. *Algorithm 1 with* $\eta = \sqrt{\frac{\ln N}{T}}$, $\eta_j = \frac{1}{2^{j-1}}\sqrt{\frac{\ln N}{T}}$ *and* $M = \left\lceil \log_2 \sqrt{\frac{\ln N}{\ln(\frac{N}{N-1})}} \right\rceil + 1$ *ensures*

$$\widetilde{L}_T \leq \min_{i \neq N} \left(L_T(i) + 3\sqrt{T \ln\left(\tfrac{N}{i}\right)} \right) + \mathcal{O}(\sqrt{T \ln(\ln(N \ln N))}).$$

This idea of combining algorithms using Hedge is useful for many other problems. It is usually a quick and easy way to verify whether some regret bound is possible or not in theory. However, the resulting algorithm might not be so elegant and practical. In the next lecture, we will study a different algorithm that not only guarantees a quantile bound (in fact even better than the one proven here), but also enjoys several more useful properties.

Lecture 5

Instructor: Haipeng Luo

1 Second Order Bounds and Squint

In this lecture we study one of the state-of-the-art algorithms for the expert problem, called Squint [Koolen and Van Erven, 2015], which enjoys many nice properties simultaneously.

To introduce the algorithm, recall that Hedge predicts $p_t(i) \propto \exp(-\eta L_{t-1}(i))$. Denote $r_t(i) = \langle p_t, \ell_t \rangle - \ell_t(i)$ and $R_t(i) = \sum_{\tau=1}^{t} r_\tau(i)$ to be the instantaneous regret and cumulative regret to expert i respectively. Hedge can then be equivalently written as $p_t(i) \propto \exp(\eta R_{t-1}(i))$. Now the first idea of Squint is to introduce a second order "correction term" in the exponent:

$$p_t(i) \propto \exp\left(\eta R_{t-1}(i) - \eta^2 V_{t-1}(i)\right)$$

where $V_t(i) = \sum_{\tau=1}^{t} r_\tau(i)^2$ is the cumulative square of the regret. In other words, the strategy is putting more weights on experts whose loss is closer to the algorithm's loss. This can also be seen as some kind of "self-confidence" since the algorithm is using its loss as a benchmark to evaluate the experts.

In fact, one can make the algorithm even more general by allowing some prior knowledge of the problem. This can be simply done by letting p_1 be the user's prior distribution over the experts (instead of a uniform distribution), and for $t > 1$ predicts

$$p_t(i) \propto p_1(i) \exp\left(\eta R_{t-1}(i) - \eta^2 V_{t-1}(i)\right). \tag{1}$$

The analysis of this algorithm is also straightforward. Let $\Phi_t = \mathbb{E}_{i \sim p_1}\left[\exp\left(\eta R_t(i) - \eta^2 V_t(i)\right)\right]$ be the potential. If $\eta \leq 1/2$, we have

$$\begin{aligned}
&\Phi_t - \Phi_{t-1} \\
&= \mathbb{E}_{i \sim p_1}\left[\exp\left(\eta R_t(i) - \eta^2 V_t(i)\right) - \exp\left(\eta R_{t-1}(i) - \eta^2 V_{t-1}(i)\right)\right] \\
&= \mathbb{E}_{i \sim p_1}\left[\exp\left(\eta R_{t-1}(i) - \eta^2 V_{t-1}(i)\right)\left(\exp\left(\eta r_t(i) - \eta^2 r_t^2(i)\right) - 1\right)\right] \\
&\leq \eta \mathbb{E}_{i \sim p_1}\left[\exp\left(\eta R_{t-1}(i) - \eta^2 V_{t-1}(i)\right) r_t(i)\right] \qquad (e^{x-x^2} \leq 1 + x,\ \forall x \geq -\tfrac{1}{2}) \\
&= \eta \sum_{i=1}^{N} \left(p_1(i) \exp\left(\eta R_{t-1}(i) - \eta^2 V_{t-1}(i)\right)\right) r_t(i) \\
&= 0
\end{aligned}$$

where the last equality is by the fact that for any $a \in \mathbb{R}_+^N$, if $p_t(i) \propto a(i)$ then for any loss vector ℓ_t,

$$\sum_{i=1}^{N} a(i) r_t(i) = \sum_{i=1}^{N} a(i)(\langle p_t, \ell_t \rangle - \ell_t(i)) = \left(\sum_{i=1}^{N} a(i)\right) \sum_{i=1}^{N} \frac{a(i)}{\sum_{j=1}^{N} a(j)} \ell_t(i) - \sum_{i=1}^{N} a(i) \ell_t(i) = 0.$$

Therefore, the potential is non-increasing and we have

$$\Phi_T \leq \Phi_{T-1} \leq \cdots \Phi_0 = 1.$$

On the other hand, by the definition of Φ_T, we have for any i,

$$\Phi_T \geq p_1(i) \exp\left(\eta R_T(i) - \eta^2 V_T(i)\right).$$

Solving for $R_T(i)$ then leads to

$$R_T(i) \leq \frac{\ln(1/p_1(i))}{\eta} + \eta V_T(i),$$

which is again the bound we have seen for Hedge if one sets p_1 to be uniform and upper bounds $V_T(i)$ by T. However, $V_T(i)$ can be much smaller than T and we will discuss more in the next section. For now, let's see how one can in fact obtain an even more general bound that competes with not just a single expert, but an arbitrary distribution over the experts. Indeed, for any distribution $q \in \Delta(N)$ that we want to compete with, we have

$$\begin{aligned}
1 \geq \Phi_T &\geq \mathbb{E}_{i\sim q}\left[\frac{p_1(i)}{q(i)}\exp\left(\eta R_T(i) - \eta^2 V_T(i)\right)\right] \\
&= \mathbb{E}_{i\sim q}\left[\exp\left(\ln\left(\frac{p_1(i)}{q(i)}\right) + \eta R_T(i) - \eta^2 V_T(i)\right)\right] \\
&\geq \exp\left(\mathbb{E}_{i\sim q}\left[\ln\left(\frac{p_1(i)}{q(i)}\right) + \eta R_T(i) - \eta^2 V_T(i)\right]\right) \qquad \text{(Jensen's inequality)} \\
&= \exp\left(-\mathrm{KL}(q, p_1) + \eta \mathbb{E}_{i\sim q}[R_T(i)] - \eta^2 \mathbb{E}_{i\sim q}[V_T(i)]\right) \\
&= \exp\left(-\mathrm{KL}(q, p_1) + \frac{\left(\mathbb{E}_{i\sim q}[R_T(i)]\right)^2}{4\mathbb{E}_{i\sim q}[V_T(i)]}\right)
\end{aligned}$$

where the last step is by using the optimal $\eta = \frac{\mathbb{E}_{i\sim q}[R_T(i)]}{2\mathbb{E}_{i\sim q}[V_T(i)]}$ (for simplicity assume it is in $[0, 1/2]$). Solving for $\mathbb{E}_{i\sim q}[R_T(i)]$ gives

$$\mathbb{E}_{i\sim q}[R_T(i)] \leq 2\sqrt{\mathbb{E}_{i\sim q}[V_T(i)]\mathrm{KL}(q, p_1)}.$$

Note that $\mathbb{E}_{i\sim q}[R_T(i)] = \widetilde{L}_T - \langle q, L_T\rangle$ is exactly the difference between the algorithm's total loss and the total loss of a fixed strategy q. The KL divergence term implies that the closer the prior knowledge p_1 is from the competitor q, the smaller the regret becomes. Before exploring the many implications of this bound, let's first discuss how to address the learning rate tuning issue again.

Squint uses a quite different technique to deal with this issue compared to what we have seen in the last lecture. The idea is to put a prior on the learning rate $\eta \in [0, 1/2]$, which resembles a Bayesian approach. The hope is that for every possible optimal tuning of η, there is a sufficient mass around it in the prior. To state the algorithm, it is in fact easier to first look at the analysis. Specifically, let γ be a prior distribution on η to be specified later and re-define $\Phi_t = \mathbb{E}_{i\sim p_1, \eta\sim\gamma}\left[\exp\left(\eta R_t(i) - \eta^2 V_t(i)\right)\right]$. By the exact same argument as before, we have

$$\begin{aligned}
\Phi_t - \Phi_{t-1} &\leq \mathbb{E}_{i\sim p_1, \eta\sim\gamma}\left[\eta \exp\left(\eta R_{t-1}(i) - \eta^2 V_{t-1}(i)\right) r_t(i)\right] \\
&= \sum_{i=1}^N \left(p_1(i)\mathbb{E}_{\eta\sim\gamma}\left[\eta \exp\left(\eta R_{t-1}(i) - \eta^2 V_{t-1}(i)\right)\right]\right) r_t(i)
\end{aligned}$$

and if we again want the last term to be zero, we need to set

$$p_t(i) \propto p_1(i)\mathbb{E}_{\eta\sim\gamma}\left[\eta \exp\left(\eta R_{t-1}(i) - \eta^2 V_{t-1}(i)\right)\right], \tag{2}$$

which defines the algorithm. Notice the extra η in the formula compared to Eq. (1) where η was a constant and could be dropped. Now for any distribution q that we want to compete with, let $\eta_\star = \frac{\mathbb{E}_{i\sim q}[R_T(i)]}{2\mathbb{E}_{i\sim q}[V_T(i)]}$ be the optimal tuning. For simplicity, assume again $\eta_\star$ is in $[0, 1/2]$ and has $\gamma(\eta_\star)$ mass in the prior. Then we have by a similar argument as before

$$\begin{aligned}
1 \geq \Phi_T &\geq \gamma(\eta_\star)\mathbb{E}_{i\sim p_1}\left[\exp\left(\eta_\star R_T(i) - \eta_\star^2 V_T(i)\right)\right] \\
&\geq \gamma(\eta_\star)\exp\left(-\mathrm{KL}(q, p_1) + \eta_\star \mathbb{E}_{i\sim q}[R_T(i)] - \eta_\star^2 \mathbb{E}_{i\sim q}[V_T(i)]\right) \\
&= \gamma(\eta_\star)\exp\left(-\mathrm{KL}(q, p_1) + \frac{\left(\mathbb{E}_{i\sim q}[R_T(i)]\right)^2}{4\mathbb{E}_{i\sim q}[V_T(i)]}\right)
\end{aligned}$$

Solving for $\mathbb{E}_{i\sim q}[R_T(i)]$ gives

$$\mathbb{E}_{i\sim q}[R_T(i)] \leq 2\sqrt{\mathbb{E}_{i\sim q}[V_T(i)]\left(\mathrm{KL}(q,p_1) + \ln\left(\frac{1}{\gamma(\eta_\star)}\right)\right)}.$$

Therefore, if $\gamma(\eta_\star)$ is large enough, we essentially obtain the bound that we aim for with a parameter-free algorithm. The only technical difficulty now is that it is impossible to ensure sufficient mass for every η in $[0, 1/2]$. Instead, we should pick a prior so that for each q, the set of *approximately* optimal η has a sufficient mass, or in other words, every η has a sufficient mass around it.

It turns out that picking $\gamma(\eta) \approx 1/\eta$ will do the job and leads to a bound that essentially replaces $\ln(1/\gamma(\eta_\star))$ by $\ln\ln T$, which is a very small overhead. We refer the interested reader to [Koolen and Van Erven, 2015] for details on the exact prior that Squint uses. Importantly, this prior leads to a closed form for the update rule (2) and the algorithm can be efficiently implemented!

2 Implications

In this section we discuss why the Squint bound is interesting and useful. For simplicity we ignore constants and negligible terms and assume we have an algorithm that achieves for any $q \in \Delta(N)$,

$$\mathbb{E}_{i\sim q}[R_T(i)] \leq \sqrt{\mathbb{E}_{i\sim q}[V_T(i)]\mathrm{KL}(q,p_1)}. \tag{3}$$

Below we prove that this bound *simultaneously* implies three adaptive regret bounds.

2.1 Bound (3) implies "small-loss" bounds

Theorem 1. *Bound* (3) *implies*

$$\mathbb{E}_{i\sim q}[R_T(i)] \leq \sqrt{2\mathbb{E}_{i\sim q}\left[\sum_{t=1}^T \max\{\ell_t(i) - \langle p_t, \ell_t\rangle, 0\}\right]\mathrm{KL}(q,p_1)} + \mathrm{KL}(q,p_1) \tag{4}$$

$$\leq \sqrt{2\langle q, L_T\rangle \mathrm{KL}(q,p_1)} + \mathrm{KL}(q,p_1) \tag{5}$$

Proof. Note that since $r_t(i) = \langle p_t, \ell_t\rangle - \ell_t(i) \in [-1, 1]$, we have

$$\mathbb{E}_{i\sim q}[V_T(i)] \leq \mathbb{E}_{i\sim q}\left[\sum_{t=1}^T |r_t(i)|\right] = \mathbb{E}_{i\sim q}\left[\sum_{t=1}^T (r_t(i) + 2\max\{-r_t(i), 0\})\right]$$

$$= \mathbb{E}_{i\sim q}[R_T(i)] + 2\mathbb{E}_{i\sim q}\left[\sum_{t=1}^T \max\{\ell_t(i) - \langle p_t, \ell_t\rangle, 0\}\right].$$

Plugging the above inequality into Eq. (3) and solving for $\mathbb{E}_{i\sim q}[R_T(i)]$ prove Eq. (4). Eq. (5) is simply by the fact $\max\{\ell_t(i) - \langle p_t, \ell_t\rangle, 0\} \leq \ell_t(i)$. □

In particular, if one simply sets p_1 to be uniform and q to be the distribution that concentrates on the best expert $i^\star$, then Eq. (5) becomes the "small-loss" bound we saw last time:

$$R_T(i^\star) \leq \sqrt{2L_T(i^\star)\ln N} + \ln N.$$

In fact, one can obtain an even tighter bound using Eq. (4) instead.

2.2 Bound (3) implies quantile bounds

Theorem 2. *Assume $L_T(1) \leq \cdots \leq L_T(N)$ without loss of generality. With p_1 being the uniform distribution Bound* (3) *implies*

$$\widetilde{L}_T \leq \min_{i\in[N]}\left(\ell_T(i) + \sqrt{2\ell_T(i)\ln\left(\frac{N}{i}\right)} + \ln\left(\frac{N}{i}\right)\right).$$

Proof. For any $i \in [N]$, setting $q(j) = 1/i$ for all $j \leq i$ and $q(j) = 0$ for all $j > i$ and using Eq. (5) give

$$\widetilde{L}_T \leq \frac{1}{i}\sum_{j=1}^{i} \ell_T(j) + \sqrt{2\left(\frac{1}{i}\sum_{j=1}^{i} \ell_T(j)\right)\ln\left(\frac{N}{i}\right)} + \ln\left(\frac{N}{i}\right).$$

Noting that $\frac{1}{i}\sum_{j=1}^{i} \ell_T(j) \leq \ell_T(i)$ and the above holds for all i simultaneously finishes the proof. □

Note that this is an improved version of the quantile bound that we proved last time, and it combines both the quantile bound and the "small-loss" bound.

2.3 Bound (3) implies constant regret in a stochastic setting

Finally, we consider a specific stochastic setting where there is a good expert that is distinguishable from others in expectation.

Theorem 3. *Suppose there exists a good expert $i^\star$ and a constant gap $\Delta \in (0, 1]$ such that $\mathbb{E}_t[\ell_t(i) - \ell_t(i^\star)] \geq \Delta$ for all t and $i \neq i^\star$, where $\mathbb{E}_t$ denotes the conditional expectation given all the randomness up to round t. Then Bound (3) implies*

$$\mathbb{E}[R_T(i^\star)] \leq \frac{\ln\left(\frac{1}{p_1(i^\star)}\right)}{\Delta}.$$

Proof. By the condition we have

$$\mathbb{E}_t[r_t(i^\star)] = \mathbb{E}_t\left[\sum_{i \neq i^\star} p_t(i)(\ell_t(i) - \ell_t(i^\star))\right] \geq \Delta(1 - p_t(i^\star)),$$

and therefore $\mathbb{E}[R_T(i^\star)] \geq \Delta B$ where we define $B = \mathbb{E}\left[\sum_{t=1}^{T}(1 - p_t(i^\star))\right]$. On the other hand,

$$\begin{aligned}
\mathbb{E}[V_T(i^\star)] &\leq \mathbb{E}\left[\sum_{t=1}^{T} |\langle p_t, \ell_t\rangle - \ell_t(i^\star)|\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_t(i)|\ell_t(i) - \ell_t(i^\star)|\right] && \text{(Jensen's inequality)} \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i \neq i^\star} p_t(i)\right] = B.
\end{aligned}$$

Therefore, by setting q to be the distribution that concentrates on $i^\star$, Eq. (3) implies

$$\begin{aligned}
\Delta B \leq \mathbb{E}[R_T(i^\star)] &\leq \mathbb{E}\sqrt{V_T(i^\star)\ln\left(\frac{1}{p_1(i^\star)}\right)} \\
&\leq \sqrt{\mathbb{E}[V_T(i^\star)]\ln\left(\frac{1}{p_1(i^\star)}\right)} && \text{(Jensen's inequality)} \\
&\leq \sqrt{B\ln\left(\frac{1}{p_1(i^\star)}\right)}. && (6)
\end{aligned}$$

Solving for B gives $B \leq \ln\left(\frac{1}{p_1(i^\star)}\right)/\Delta^2$. Plugging this back to Eq. (6) finishes the proof. □

For a uniform prior p_1, the bound simply becomes $(\ln N)/\Delta$, which is independent of T!

References

Wouter M Koolen and Tim Van Erven. Second-order quantile methods for experts and combinatorial games. In *28th Annual Conference on Learning Theory*, 2015.

Lecture 6

Instructor: Haipeng Luo

1 Variation Bounds

Although Squint enjoys several nice properties simultaneously, there are still other "easy" cases that Squint does not cover. In this lecture we will talk about one of those where the "variation" of the experts' losses is small in some senses.

Recall that the Squint's regret bound is in terms of the quantity

$$V_T(i) = \sum_{t=1}^{T} (\langle p_t, \ell_t \rangle - \ell_t(i))^2,$$

where we use the loss of the algorithm $\langle p_t, \ell_t \rangle$ as a benchmark to measure the performance of each expert. In general, for an arbitrary benchmark $m_t \in [0,1]^N$, is it possible to obtain a bound that is in terms of the variation quantity

$$D_T(i) = \sum_{t=1}^{T} (m_t(i) - \ell_t(i))^2 \quad ?$$

The answer is obviously no if there are no restrictions on the benchmark m_t at all, because otherwise setting $m_t = \ell_t$ would lead to zero regret. To see when this is possible, let's first revisit the analysis of Squint (with a fixed η) and see what needs to be changed accordingly to obtain such bounds. Naturally, we redefine the potential as $\Phi_t = \mathbb{E}_{i \sim p_1} \left[\exp(\eta R_t(i) - \eta^2 D_t(i)) \right]$. We then have with $d_t(i) = m_t(i) - \ell_t(i)$ and $\eta < 1/2$,

$$\begin{aligned}
&\Phi_t - \Phi_{t-1} \\
&= \mathbb{E}_{i \sim p_1} \left[\exp\left(\eta R_t(i) - \eta^2 D_t(i)\right) - \exp\left(\eta R_{t-1}(i) - \eta^2 D_{t-1}(i)\right) \right] \\
&= \mathbb{E}_{i \sim p_1} \left[\exp\left(\eta R_{t-1}(i) - \eta^2 D_{t-1}(i)\right) \left(\exp\left(\eta r_t(i) - \eta^2 d_t^2(i)\right) - 1 \right) \right] \\
&= \mathbb{E}_{i \sim p_1} \left[\exp\left(\eta R_{t-1}(i) - \eta^2 D_{t-1}(i)\right) e^{\eta r_t(i) - \eta d_t(i)} \left(e^{\eta d_t(i) - \eta^2 d_t^2(i)} - e^{\eta d_t(i) - \eta r_t(i)} \right) \right] \\
&= \mathbb{E}_{i \sim p_1} \left[\exp\left(\eta R_{t-1}(i) - \eta^2 D_{t-1}(i) + \eta \langle p_t, \ell_t \rangle - \eta m_t(i)\right) \left(e^{\eta d_t(i) - \eta^2 d_t^2(i)} - e^{\eta d_t(i) - \eta r_t(i)} \right) \right] \\
&= \mathbb{E}_{i \sim p_1} \left[\exp\left(\eta R_{t-1}(i) - \eta^2 D_{t-1}(i) + \eta \langle p_t, \ell_t \rangle - \eta m_t(i)\right) \left(1 + \eta d_t(i) - e^{\eta d_t(i) - \eta r_t(i)} \right) \right] \\
&\qquad\qquad (e^{x - x^2} \le 1 + x, \ \forall x \ge -\tfrac{1}{2}) \\
&= \mathbb{E}_{i \sim p_1} \left[\exp\left(\eta R_{t-1}(i) - \eta^2 D_{t-1}(i) + \eta \langle p_t, \ell_t \rangle - \eta m_t(i)\right) \eta r_t(i) \right]. \\
&\qquad\qquad (e^x \ge 1 + x, \ \forall x)
\end{aligned}$$

As discussed before, if we want the last term to be zero, the algorithm should play

$$p_t(i) \propto p_1(i) \exp\left(\eta R_{t-1}(i) - \eta^2 D_{t-1}(i) + \eta \langle p_t, \ell_t \rangle - \eta m_t(i)\right).$$

Noting that $\eta \langle p_t, \ell_t \rangle$ is a constant for all experts, the above is equivalent to

$$p_t(i) \propto p_1(i) \exp\left(\eta R_{t-1}(i) - \eta^2 D_{t-1}(i) - \eta m_t(i)\right). \tag{1}$$

Now by the exact same argument as Squint, with an optimal tuned η, the fact $\Phi_T \leq 1$ implies that for any $q \in \Delta(N)$,

$$\mathbb{E}_{i\sim q}[R_T(i)] \leq 2\sqrt{\mathbb{E}_{i\sim q}[D_T(i)]\mathrm{KL}(q, p_1)}.$$

Therefore, ignoring the learning rate tuning issue for a moment, we indeed obtain a bound that replaces V_T by D_T for any benchmark m_t, *as long as* m_t is available at the beginning of round t (that is, before seeing ℓ_t) so that Eq. (1) is a valid algorithm.

This excludes the choice of $m_t = \ell_t$ (which makes sense), but there are still several interesting valid choices. For example, setting $m_t(i) = \mu_{t-1}(i)$ where $\mu_t(i)$ is the empirical average loss of expert i at round t (that is $\mu_t(i) = \frac{1}{t}\sum_{\tau=1}^{t} \ell_\tau(i)$ and $\mu_0(i) = 0$), we have

$$\begin{aligned}
D_T(i) &= \sum_{t=1}^{T} (\ell_t(i) - \mu_{t-1}(i))^2 \\
&= \sum_{t=1}^{T} (\ell_t(i) - \mu_t(i))^2 + (2\ell_t(i) - \mu_t(i) - \mu_{t-1}(i))(\mu_t(i) - \mu_{t-1}(i)) \\
&= \sum_{t=1}^{T} (\ell_t(i) - \mu_t(i))^2 + (2\ell_t(i) - \mu_t(i) - \mu_{t-1}(i))\frac{\ell_t(i) - \mu_{t-1}(i)}{t} \\
&\leq \sum_{t=1}^{T} (\ell_t(i) - \mu_t(i))^2 + \frac{2}{t} \\
&\leq \sum_{t=1}^{T} \left(\ell_t(i) - \mu_T(i)\right)^2 + \mathcal{O}(\ln T),
\end{aligned}$$

where the last step is actually due to the BTL lemma (hint: $\mu_t(i) = \mathrm{argmin}_\mu \sum_{\tau=1}^{t} (\ell_\tau(i) - \mu)^2$). Therefore, in this case $D_T(i)$ is essentially the empirical variance of expert i's losses (up to $\mathcal{O}(\ln T)$), and the bound indicates that small variance implies small regret.

Another example is to let $m_t(i) = \ell_{t-1}(i)$ for $t \neq 1$ and $m_1(i) = 0$, so that $D_T(i) = \sum_{t=1}^{T} (\ell_t(i) - \ell_{t-1}(i))^2$ measures the variation of expert i's losses over time, which is sometimes called "path-length" bound. In fact, path-length is also bounded by the variance up to constants:

$$\begin{aligned}
\sum_{t=1}^{T} (\ell_t(i) - \ell_{t-1}(i))^2 &\leq 3\sum_{t=1}^{T} \left((\ell_t(i) - \mu_t(i))^2 + (\ell_{t-1}(i) - \mu_{t-1}(i))^2 + (\mu_t(i) - \mu_{t-1}(i))^2\right) \\
&\leq 6\sum_{t=1}^{T} \left(\ell_t(i) - \mu_T(i)\right)^2 + 3\sum_{t=1}^{T} \frac{1}{t^2} \\
&\leq 6\sum_{t=1}^{T} \left(\ell_t(i) - \mu_T(i)\right)^2 + 6.
\end{aligned}$$

However, one can construct an example where the path-length is a constant while the variance (or even $\sum_{t=1}^{T} (\ell_t(i) - \mu_{t-1}(i))^2$) is of order $\Omega(T)$. (Simply think about the case where $\ell_t(i) = 0$ for all $t \leq T/2$ and $\ell_t(i) = 1$ otherwise.)

Coming back to the learning rate tuning issue, while one would naturally imagine that using the similar idea of putting a prior on η as Squint could possibly solve the problem, it actually does not work here in general. The reason is that according to discussions from last lecture the strategy should become

$$p_t(i) \propto p_1(i)\mathbb{E}_\eta \left[\eta \exp\left(\eta R_{t-1}(i) - \eta^2 D_{t-1}(i) + \eta \left\langle p_t, \ell_t\right\rangle - \eta m_t(i)\right)\right],$$

and critically the term $\eta \left\langle p_t, \ell_t\right\rangle$ cannot be factored out and removed anymore. This makes the algorithm invalid because it depends on ℓ_t which is of course not available before playing p_t. The only exception is to have $m_t(i) = \langle p_t, \ell_t\rangle$ so that the last two terms in the exponent cancel with each other, which simply just recovers Squint. It is in fact still not clear whether there is a parameter-free

algorithm that achieves the bound $\sqrt{\mathbb{E}_{i\sim q}[D_T(i)]\mathrm{KL}(q,p_1)}$ simultaneously for all q, even for the two special choices of m_t discussed previously.

Nevertheless, we remark that obtaining a weaker bound such as

$$R_T(i^\star) \le \mathcal{O}\left(\sqrt{\left(\max_{i\in[N]} D_T(i)\right)\ln N}\right)$$

with a parameter-free algorithm is possible via simple techniques such as the doubling trick (details omitted). While the bound is now in terms of the variation of all experts instead of the best expert, it could still be much smaller than the worse case $\mathcal{O}(\sqrt{T\ln N})$.

2 Optimistic FTRL

In this section we explore a different technique that allows one to obtain similar bounds but with a very important difference. The new algorithm is simply to remove the term $\eta^2 D_{t-1}(i)$ in Eq. (1). However, it is easier to interpret and analyze the algorithm if we write it in an FTRL format:

$$p_t = \operatorname*{argmin}_{p\in\Delta(N)} \langle p, L_{t-1} + m_t\rangle + \frac{1}{\eta}\psi(p). \tag{2}$$

where $\psi(p)$ is the negative entropy. One can verify that this is exactly equivalent to $p_t(i) \propto \exp(-\eta(L_{t-1}(i) + m_t(i)))$. Compared to the standard FTRL, the only difference here is the term m_t. One way to interpret the algorithm is that we are optimistic that the loss for round t will be close to m_t, and therefore we incorporate m_t into the cumulative loss and treat $L_{t-1} + m_t$ as a proxy for L_t. This is called optimistic FTRL [Rakhlin and Sridharan, 2013, Syrgkanis et al., 2015].

We prove the following theorem for this algorithm. Note that while we restrict to the expert setting with entropy regularizer, similar results hold for the general OCO setting with any decision space and any strongly convex regularizer.

Theorem 1. *Optimistic FTRL* (2) *ensures for any* $q \in \Delta(N)$,

$$\sum_{t=1}^T \langle p_t - q, \ell_t\rangle \le \frac{2+\ln N}{\eta} + \eta\sum_{t=1}^T \|\ell_t - m_t\|_\infty^2 - \frac{1}{4\eta}\sum_{t=1}^T \|p_{t+1} - p_t\|_1^2 .$$

Note that the term $\sum_{t=1}^T \|\ell_t - m_t\|_\infty^2 = \sum_{t=1}^T \max_i(\ell_t(i) - m_t(i))^2$ is worse than $\mathbb{E}_{i\sim q}[D_T(i)]$ or even $\max_i D_T(i)$ discussed in the last section. However, the negative term is crucial and turns out to be very useful. At first glance the negative term is a bit counter-intuitive since it is suggesting that less stability leads to smaller regret. We defer the discussion to the next lecture where we will see why this makes sense in the context of game theory.

Proof. Let $p'_t = \operatorname{argmin}_p \langle p, L_{t-1}\rangle + \frac{1}{\eta}\psi(p)$ be the regular FTRL strategy. The regret can be decomposed as

$$\sum_{t=1}^T \langle p_t - q, \ell_t\rangle = \sum_{t=1}^T \left\langle p_t - p'_{t+1}, \ell_t - m_t\right\rangle + \sum_{t=1}^T \left(\left\langle p_t - p'_{t+1}, m_t\right\rangle + \left\langle p'_{t+1} - q, \ell_t\right\rangle\right).$$

The first summation can be bounded in a similar way as in the stability lemma of FTRL, which we summarize again in Lemma 1. Using the lemma with $L = L_{t-1} + m_t$ and $L' = L_t$, we have

$$\left\|p_t - p'_{t+1}\right\| \le \eta \left\|\ell_t - m_t\right\|_\infty ,$$

and therefore $\left\langle p_t - p'_{t+1}, \ell_t - m_t\right\rangle \le \left\|p_t - p'_{t+1}\right\| \left\|\ell_t - m_t\right\|_\infty \le \eta \left\|\ell_t - m_t\right\|_\infty^2$.

It remains to bound the second summation. Intuitively, since the optimization in calculating p_t takes m_t into account while the one for p'_{t+1} does not, the term $\left\langle p_t - p'_{t+1}, m_t\right\rangle$ should be small. On the other hand, the other term $\left\langle p'_{t+1} - q, \ell_t\right\rangle$ is basically the regret for BTL, which also should be small. In fact, we will prove the following stronger statement

$$\sum_{t=1}^T \left(\left\langle p_t - p'_{t+1}, m_t\right\rangle + \left\langle p'_{t+1} - q, \ell_t\right\rangle\right) \le \frac{\ln N + \psi(q) - A_T}{\eta}, \tag{3}$$

where $A_T = \frac{1}{2}\sum_{t=1}^{T}\left(\left\|p_t - p'_{t+1}\right\|^2 + \left\|p_t - p'_t\right\|^2\right)$. This will finish the proof since $\psi(q) \le 0$ and

$$\begin{aligned}
A_T &= \frac{1}{2}\sum_{t=1}^{T}\left(\left\|p_t - p'_{t+1}\right\|^2 + \left\|p_t - p'_t\right\|^2\right) \\
&\ge \frac{1}{2}\sum_{t=1}^{T}\left(\left\|p_t - p'_{t+1}\right\|^2 + \left\|p_{t+1} - p'_{t+1}\right\|^2\right) - \frac{1}{2}\left\|p_{T+1} - p'_{T+1}\right\|^2 \\
&\ge \frac{1}{4}\sum_{t=1}^{T}\left(\left(\left\|p_t - p'_{t+1}\right\| + \left\|p_{t+1} - p'_{t+1}\right\|\right)^2\right) - 2 && ((a+b)^2 \le 2a^2 + 2b^2) \\
&\ge \frac{1}{4}\sum_{t=1}^{T}\left\|p_t - p_{t+1}\right\|^2 - 2 && \text{(by triangle inequality)}
\end{aligned}$$

We use induction to prove Eq. (3). The base case when $T = 0$ holds trivially since $\psi(q) \ge -\ln N$. Now assume we have for any q,

$$\sum_{t=1}^{T-1}\left(\left\langle p_t - p'_{t+1}, m_t\right\rangle + \left\langle p'_{t+1} - q, \ell_t\right\rangle\right) \le \frac{\ln N + \psi(q) - A_{T-1}}{\eta}. \tag{4}$$

We then have

$$\begin{aligned}
&\sum_{t=1}^{T}\left(\left\langle p_t - p'_{t+1}, m_t\right\rangle + \left\langle p'_{t+1}, \ell_t\right\rangle\right) \\
&\le \left\langle p_T - p'_{T+1}, m_T\right\rangle + \left\langle p'_{T+1}, \ell_T\right\rangle + \frac{\ln N + \psi(p'_T) - A_{T-1}}{\eta} + \left\langle p'_T, L_{T-1}\right\rangle \\
&\qquad \text{(by setting } q = p'_T \text{ in Eq. (4))} \\
&\le \left\langle p_T - p'_{T+1}, m_T\right\rangle + \left\langle p'_{T+1}, \ell_T\right\rangle + \frac{\ln N + \psi(p_T) - A_{T-1} - \frac{1}{2}\left\|p_T - p'_T\right\|^2}{\eta} + \left\langle p_T, L_{T-1}\right\rangle \\
&\qquad \text{(by Eq. (5))} \\
&= \left\langle p'_{T+1}, \ell_T - m_T\right\rangle + \frac{\ln N + \psi(p_T) - A_{T-1} - \frac{1}{2}\left\|p_T - p'_T\right\|^2}{\eta} + \left\langle p_T, L_{T-1} + m_T\right\rangle \\
&\le \left\langle p'_{T+1}, \ell_T - m_T\right\rangle + \frac{\ln N + \psi(p'_{T+1}) - A_T}{\eta} + \left\langle p'_{T+1}, L_{T-1} + m_T\right\rangle \qquad \text{(by Eq. (5))} \\
&= \frac{\ln N + \psi(p'_{T+1}) - A_T}{\eta} + \left\langle p'_{T+1}, L_T\right\rangle \\
&\le \frac{\ln N + \psi(q) - A_T}{\eta} + \left\langle q, L_T\right\rangle. \qquad \text{(by optimality of } p'_{T+1}\text{)}
\end{aligned}$$

Rearranging finishes the induction and thus proves the theorem. □

Lemma 1 (Stability)**.** *If $p_\star = \operatorname{argmin}_p \langle p, L\rangle + \frac{1}{\eta}\psi(p)$ and $p'_\star = \operatorname{argmin}_p \langle p, L'\rangle + \frac{1}{\eta}\psi(p)$ for a 1-strongly convex regularizer ψ (with respect to a norm $\|\cdot\|$) and some L and L'. Then*

$$\|p_\star - p'_\star\| \le \eta \|L - L'\|_\star.$$

Proof. Let $F(p; L) = \langle p, L\rangle + \frac{1}{\eta}\psi(p)$. By strong convexity and first order optimality we have for any q,

$$\begin{aligned}
F(p_\star; L) &\le F(q; L) + \langle \nabla F(p_\star, L); p_\star - q\rangle - \tfrac{1}{2\eta}\|p_\star - q\|^2 \\
&\le F(q; L) - \tfrac{1}{2\eta}\|p_\star - q\|^2.
\end{aligned} \tag{5}$$

Similar statement holds for $p'_\star$. Therefore we have

$$\langle p_\star - p'_\star, L' - L \rangle = F(p'_\star; L) - F(p_\star; L) + F(p_\star; L') - F(p'_\star; L') \geq \tfrac{1}{\eta} \left\| p_\star - p'_\star \right\|^2,$$

and on the other hand by Hölder's inequality

$$\langle p_\star - p'_\star, L' - L \rangle \leq \left\| p_\star - p'_\star \right\| \left\| L - L' \right\|_\star.$$

Combining the two inequalities proves the lemma. □

References

Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems 26*, 2013.

Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. In *Advances in Neural Information Processing Systems 28*, 2015.

Lecture 7

Instructor: Haipeng Luo

1 Two-player Zero-sum Games

In this lecture we explore the connection between game theory and online learning. We focus on simple two-player zero-sum games that could be represented using a matrix $G \in [0,1]^{N \times M}$, where one player (called the row player) has N possible actions and another player (called the column player) has M possible actions, and entry $G(i,j)$ represents the loss of the row player if he/she picks action i while the opponent picks action j, which is also the reward for the column player (hence zero-sum).

A classic example is the Rock-Paper-Scissors game. If we assign loss 1 for losing the game, 0 for winning and $1/2$ for a tie, then G is

$$\begin{array}{c} \\ \text{Rock} \\ \text{Paper} \\ \text{Scissors} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Rock} & \text{Paper} & \text{Scissors} \end{array} \\ \begin{pmatrix} 1/2 & 1 & 0 \\ 0 & 1/2 & 1 \\ 1 & 0 & 1/2 \end{pmatrix} \end{array}.$$

Instead of playing a fixed action (also called "pure strategy"), it often makes more sense to play a action randomly according to a distribution (called "mixed strategy"). For some mixed strategy $p \in \Delta(N)$ for the row player and some mixed strategy $q \in \Delta(M)$ for the column player, the expected loss for the row player, which is also the expected reward of the column player, is denoted by $G(p,q) = \sum_{i,j} p(i)q(j)G(i,j)$. We will also use the notation $G(i,q)$ and $G(p,j)$ to denote $\sum_j q(j)G(i,j)$ and $\sum_i p(i)G(i,j)$ respectively.

Perhaps the most important notion in game theory is the *Nash equilibrium*. A pair of mixed strategy (p,q) is called a Nash equilibrium if neither player has a incentive to change his/her strategy given that the opponent is keeping his/hers. In other words, everyone is happy about the current situation. Formally, this means that

$$G(p,q') \le G(p,q) \le G(p',q), \quad \forall p' \in \Delta(N), q' \in \Delta(M).$$

One can easily verify that for the Rock-Paper-Scissors game, playing uniformly at random for both players is a Nash Equilibrium (in fact the only one).

On the other hand, minimax solution is also a natural concept for a two-player zero-sum game. Specifically, in the worst case, playing p leads to a loss of at most $\max_q G(p,q)$ for the row player if the column player sees p before making decisions, and therefore in this sense the worst-case optimal strategy for the row player is $p^\star \in \operatorname{argmin}_p \max_q G(p,q)$, which is called the minimax strategy. Similarly, the maximin strategy for the column player is $q^\star \in \operatorname{argmax}_q \min_p G(p,q)$. Together, we call $(p^\star, q^\star)$ a minimax solution of the game.

Therefore, $\min_p \max_q G(p,q)$ and $\max_q \min_p G(p,q)$ are respectively the smallest loss and the largest reward the respective player can hope for when against an optimal opponent who plays second. How are these two values related? Intuitively, both players are playing optimally in the two expressions, but there should be no disadvantage in playing second. Therefore we should have $\min_p \max_q G(p,q) \ge \max_q \min_p G(p,q)$ (row player playing first on the left and second on the right). Indeed, this is true by a simple argument:

$$\min_p \max_q G(p,q) = \max_q G(p^\star, q) \ge G(p^\star, q^\star) \ge \min_p G(p, q^\star) = \max_q \min_p G(p,q).$$

While one may imagine that this inequality should be strict at least for some cases, the surprising fact is that the reverse inequality is also true and therefore the two values are exactly the same! In other words, if both players are playing optimally, there is no difference in playing first or second. This is the celebrated von Neumann's minimax theorem.

Theorem 1 (von Neumann's minimax theorem)**.** *For any two-player zero-sum game $G \in [0,1]^{N\times M}$, we have*

$$\min_p \max_q G(p,q) = \max_q \min_p G(p,q).$$

This single value is called the value of the game, denoted by $v(G)$. The original proof relies on a fixed-point theorem, but we will prove it in a different way *by running online learning algorithms* in the next section. For now, we discuss the connection between these different notions we have talked about so far: Nash equilibrium, minimax solution, and the value of the game.

Theorem 2. *A pair of mixed strategy (p,q) is a Nash equilibrium if and only if it is also a minimax solution. Moreover, $G(p,q)$ is the value of the game.*

Proof. Suppose (p,q) is a Nash equilibrium. By definition and optimality, we have

$$\min_{p'} \max_{q'} G(p',q') \le \max_{q'} G(p,q') = G(p,q) = \min_{p'} G(p',q) \le \max_{q'} \min_{p'} G(p',q').$$

Now by the minimax theorem, the above inequalities are actually equalities, which implies that $G(p,q) = v(G)$ and also (p,q) is a minimax solution.

Next for the other direction, if (p,q) is a minimax solution, then again by optimality and definition

$$\min_{p'} \max_{q'} G(p',q') = \max_{q'} G(p,q') \ge G(p,q) \ge \min_{p'} G(p',q) = \max_{q'} \min_{p'} G(p',q').$$

By the minimax theorem, the above is again an equality, which implies $G(p,q) = v(G)$ and (p,q) is a Nash equilibrium. □

By this theorem and the fact that minimax solutions always exist (due to compactness of the simplex), Nash equilibria also always exist.

2 Repeated Play

How should we play a game? If we know the matrix G, then playing with the minimax solutions seems to be a good strategy. However, what if G is unknown? Moreover, minimax solutions might also be too pessimistic. For example, if we play Rock-Paper-Scissors with a friend who we know prefers to play Paper for instance, then should we still play uniformly at random? In general, how do we exploit the fact that the opponent might not be optimal?

If the game is only played once, then there is little we can do. However, it is often the case that a game is repeatedly played for many times. In this case, there is hope to apply learning algorithms to learn to play well against a specific opponent. We take the row player as an example and formulated the learning model as follows: at round $t = 1, \dots, T$,

- the row player chooses mixed strategy p_t;
- the column player chooses mixed strategy q_t (which may or may not depend on p_t);
- the row player observes $G(i, q_t)$ for all $i \in [N]$.

The feedback model can be potentially extended to the more realistic case where only $G(i,j)$ is observed for some i and j drawn from p_t and q_t respectively, but for now we will stick with this easier full information feedback.

A very natural idea for the player is to make use of an expert algorithm such as Hedge, treating each available action i as an expert. Specifically, given an expert algorithm as a blackbox, p_t will be the output of this algorithm at round t, and the loss vector to pass back to the algorithm would be ℓ_t

such that $\ell_t(i) = G(i, q_t),\ \forall i$. Suppose the expert algorithm has regret bound $\mathcal{R}_T$, then it implies

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T} G(p_t, q_t) &\le \min_p \frac{1}{T}\sum_{t=1}^{T} G(p, q_t) + \frac{\mathcal{R}_T}{T} \\ &= \min_p G(p, \bar{q}) + \frac{\mathcal{R}_T}{T} && (\bar{q} = \tfrac{1}{T}\textstyle\sum_{t=1}^{T} q_t) \\ &\le \max_q \min_p G(p, q) + \frac{\mathcal{R}_T}{T}.\end{aligned}$$

Therefore, if the regret is sublinear and T is large, then the average loss of the row player is very close to the value of the game, which is the smallest possible loss if against an optimal opponent. However, by using a learning algorithm instead of a minimax solution directly (if it is available), the average loss can also be much smaller in the case when the opponent is not exactly optimal (that is, when $\bar{q}$ is not close to the maximin strategy and the last inequality is loose).

However, even more interesting thing happens if the column player also uses an expert algorithm (by using the negative rewards as losses). To see this, suppose the regret bound for the column player is $\mathcal{R}'_T$:

$$\sum_{t=1}^{T} -G(p_t, q_t) - \min_q \sum_{t=1}^{T} -G(p_t, q) = \max_q \sum_{t=1}^{T} G(p_t, q) - \sum_{t=1}^{T} G(p_t, q_t) \le \mathcal{R}'_T.$$

Then we have

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T} G(p_t, q_t) &\ge \max_q \frac{1}{T}\sum_{t=1}^{T} G(p_t, q) - \frac{\mathcal{R}'_T}{T} \\ &= \max_q G(\bar{p}, q) - \frac{\mathcal{R}'_T}{T} && (\bar{p} = \tfrac{1}{T}\textstyle\sum_{t=1}^{T} p_t) \\ &\ge \min_p \max_q G(p, q) - \frac{\mathcal{R}'_T}{T}.\end{aligned}$$

Combining the two derivations, we have

$$\min_p \max_q G(p, q) \le \frac{1}{T}\sum_{t=1}^{T} G(p_t, q_t) + \frac{\mathcal{R}'_T}{T} \le \max_q \min_p G(p, q) + \frac{\mathcal{R}_T}{T} + \frac{\mathcal{R}'_T}{T}.$$

If $\mathcal{R}_T$ and $\mathcal{R}'_T$ are sublinear, which we can indeed ensure by using for example Hedge, then the term $\frac{\mathcal{R}_T}{T} + \frac{\mathcal{R}'_T}{T}$ can be arbitrarily close to 0 as T goes to infinity. Therefore, we must have

$$\min_p \max_q G(p, q) \le \max_q \min_p G(p, q),$$

which means that we just proved the minimax theorem (recall the other direction is trivial)! This is one of the few proofs that prove a mathematical statement by running algorithms, and is taken from [Freund and Schapire, 1999].

In fact, the derivations above also tell us

$$\max_q G(\bar{p}, q) \le \min_p \max_q G(p, q) + \frac{\mathcal{R}_T}{T} + \frac{\mathcal{R}'_T}{T} \quad \text{and} \quad \max_q \min_p G(p, q) \le \min_p G(p, \bar{q}) + \frac{\mathcal{R}_T}{T} + \frac{\mathcal{R}'_T}{T},$$

which means $\bar{p}$ and $\bar{q}$ are approximately minimax solutions with error $\frac{\mathcal{R}_T}{T} + \frac{\mathcal{R}'_T}{T}$. In other words, this also provides a concrete way to calculate a minimax solution/Nash equilibrium.

3 Faster Convergence via Adaptivity

We know that the worst-case optimal regret for the expert problem is of order $\mathcal{O}(\sqrt{T})$, which means the convergence rate of the above approach is of order $\mathcal{O}(1/\sqrt{T})$. Is this the optimal rate in this specific context? The answer is no – one can in fact converge much faster using an expert algorithm

with some special adaptive property. Specifically, recall the bound we prove for Optimistic FTRL in Lecture 6 (with $m_t = \ell_{t-1}$):

$$\mathcal{R}_T \leq \frac{2+\ln N}{\eta} + \eta \sum_{t=1}^{T} \|\ell_t - \ell_{t-1}\|_\infty^2 - \frac{1}{4\eta} \sum_{t=1}^{T} \|p_{t+1} - p_t\|_1^2 . \tag{1}$$

In the context of game playing, for the row player we have by Hölder's inequality (for $t \neq 1$)

$$\|\ell_t - \ell_{t-1}\|_\infty^2 = \max_i |G(i, q_t) - G(i, q_{t-1})|^2 = \max_i |\langle G(i,\cdot), q_t - q_{t-1}\rangle|^2 \leq \|q_t - q_{t-1}\|_1^2$$

where we use $G(i,\cdot)$ to denote the i-th row of G. In other words, from the row player's perspective, the stability of the environment is controlled by the stability of the column player's strategy. The exact same argument holds for the column player and therefore if both players use Optimistic FTRL with the same learning rate η, then we have

$$\mathcal{R}_T \leq \frac{2+\ln N}{\eta} + \eta + \eta \sum_{t=2}^{T} \|q_t - q_{t-1}\|_1^2 - \frac{1}{4\eta} \sum_{t=2}^{T} \|p_t - p_{t-1}\|_1^2$$

$$\mathcal{R}'_T \leq \frac{2+\ln M}{\eta} + \eta + \eta \sum_{t=2}^{T} \|p_t - p_{t-1}\|_1^2 - \frac{1}{4\eta} \sum_{t=2}^{T} \|q_t - q_{t-1}\|_1^2 .$$

Summing up the two bounds gives

$$\mathcal{R}_T + \mathcal{R}'_T \leq \frac{4+\ln(NM)}{\eta} + 2\eta + \left(\eta - \frac{1}{4\eta}\right) \sum_{t=2}^{T} \left(\|p_t - p_{t-1}\|_1^2 + \|q_t - q_{t-1}\|_1^2\right),$$

and simply setting $\eta = 1/2$ leads to

$$\mathcal{R}_T + \mathcal{R}'_T \leq 9 + 2\ln(NM),$$

which is independent of T! In other words, the average strategy $(\bar{p}, \bar{q})$ converges to the Nash equilibrium at a rate $\mathcal{O}(1/T)$ instead of $\mathcal{O}(1/\sqrt{T})$. In fact, similar results hold even if the two players do not use the exact same optimistic FTRL algorithm. The key is clearly only the special adaptive bound in the form of Eq. (1) and there are several other algorithms that enjoy similar bounds. See [Syrgkanis et al., 2015] for details.

We finally point out that even if we only look at each player's individual regret, it could still be smaller than the worst-case $\mathcal{O}(\sqrt{T})$. Take the row player as an example, suppose the column player's algorithm is stable in the sense that $\|q_t - q_{t-1}\|_1 \leq c\eta$ for some constant $c > 0$. Then the regret for the row player who uses Optimistic FTRL is

$$\mathcal{R}_T \leq \frac{2+\ln N}{\eta} + \eta + \eta \sum_{t=2}^{T} \|q_t - q_{t-1}\|_1^2 \leq \frac{2+\ln N}{\eta} + \eta + c^2 T \eta^3.$$

Setting $\eta = \left(\frac{\ln N}{T}\right)^{\frac{1}{4}}$ we have $\mathcal{R}_T = \mathcal{O}(T^{\frac{1}{4}}(\ln N)^{\frac{3}{4}})$, which is again better than $\mathcal{O}(\sqrt{T})$. The stability condition on q_t is not unfamiliar to us by now – we know that if the column player uses Hedge (with learning rate η), then stability holds with $c = 1$; if the column player uses Optimistic FTRL (with learning rate η), then stability holds with $c = 2$ (use the stability lemma of Lecture 6 to verify why this is true).

References

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. In *Advances in Neural Information Processing Systems 28*, 2015.

Lecture 8

Instructor: Haipeng Luo

1 Boosting and AdaBoost

In this lecture we discuss the connection between boosting and online learning. Boosting is not only one of the most fundamental theories in machine learning, but also one of the most widely used machine learning algorithms in practice. It was originally proposed in a statistical learning setting for binary classification, which will also be the focus here. Informally, the key question that boosting tries to answer is that if we have a learning algorithm with some nontrivial (but also not great) prediction accuracy (say 51%), is it possible to boost the accuracy to something arbitrarily high (say 99.9%), in a blackbox manner?

Formally we consider the following binary classification task. Given a set of training examples $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ where each $(x_i, y_i) \in \mathcal{X} \times \{-1, +1\}$ (for some feature space $\mathcal{X}$) is an i.i.d. sample of an unknown distribution $\mathcal{D}$, our goal is to output a binary classifier $H : \mathcal{X} \to \{-1, +1\}$ with small training error $\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{H(x_i) \neq y_i\}$ and more importantly small generalization error $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathbf{1}\{H(x) \neq y\}]$.

Now suppose we are given a *weak learning oracle* $\mathcal{A}$ which takes a training set S and a distribution over the training examples $p \in \Delta(N)$ as input (in other words, a weighted training set), and outputs a classifier $h = \mathcal{A}(S, p) \in \mathcal{H}$ for some hypothesis space $\mathcal{H}$ such that the weighted average error according to p is always bounded as

$$\mathbb{E}_{i\sim p}[\mathbf{1}\{h(x_i) \neq y_i\}] = \sum_{i:h(x_i)\neq y_i} p(i) \leq \frac{1}{2} - \gamma \tag{1}$$

for some constant $\gamma > 0$. This is called the *weak learning assumption* and h is also called a weak classifer. One should think about γ as something very small such as 1%, so the weak learning assumption is very mild and is just saying that the oracle $\mathcal{A}$ is doing something slightly nontrivial since the trivial random guessing has expected error rate $1/2$. We called γ the *edge* of the oracle.

In practice, such oracle can be any existing off-the-shelf learning algorithms, such as SVM, decision tree algorithms (e.g. C4.5), neural nets algorithms, or even something much simpler such as decision stump algorithms (a stump is a tree with depth 1). Many of these algorithms support taking weighted training set as input. However, even if weighted training set is not supported, one can simply generate a new (and large enough) training set by sampling from S according to p with replacement and use it as the input for the oracle instead, so that the (unweighted) training error on this new training set is close to the weighted error rate on S.

A boosting algorithm is then a master algorithm that uses the oracle as a subroutine and outputs a strong classifier with very high accuracy. The fact that this is even possible is highly nontrivial and was not clear until the seminal work of [Schapire, 1990]. The idea is to repeatedly apply the oracle with a distribution that focuses on "hard" examples and to combine all the weak classifiers outputted by the oracle in some smart way.

The most successful boosting algorithm is AdaBoost [Freund and Schapire, 1997] (short for Adaptive Boosting), outlined in Algorithm 1. At each iteration AdaBoost maintains a distribution p_t and invoke the oracle with a training set weighted by p_t to get a weak classifier h_t. Afterwards, based on the weighted error rate of h_t, p_t is updated in a multiplicative way so that misclassified examples get more weights in the next iteration. The final output of AdaBoost is a weighted majority vote of

Algorithm 1: AdaBoost

Input: training set $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, weak learning oracle $\mathcal{A}$
Initialize: $p_1(i) = 1/N$ for $i = 1, \ldots, N$
for $t = 1, \ldots, T$ **do**
  invoke the oracle to get $h_t = \mathcal{A}(S, p_t)$
  calculate weighted error $\epsilon_t = \mathbb{E}_{i\sim p}[\mathbf{1}\{h(x_i) \neq y_i\}]$, and $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
  update distribution: $\forall i \in [N]$

$$p_{t+1}(i) \propto p_t(i) \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{else} \end{cases}$$

output the final classifier: $H(x) = \textsc{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$.

all the weak classifiers ($\textsc{sgn}(z)$ is 1 if $z > 0$ and -1 otherwise), where the weight (α_t) for each h_t is calculated based on the error rate of h_t so that more accurate classifier has more weights in the final vote. (Note that the algorithm makes sense even when $\epsilon_t \geq 1/2$. Think about why.)

The first result about AdaBoost is that it drives the training error to zero exponentially fast.

Theorem 1. *After T rounds, the error rate of the final output of AdaBoost is bounded as*

$$\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{H(x_i) \neq y_i\} \leq \exp\left(-2\sum_{t=1}^{T}\gamma_t^2\right) \tag{2}$$

where $\gamma_t = \frac{1}{2} - \epsilon_t$ is the edge of classifier h_t. Under the weak learning assumption, the error rate is then bounded by $\exp\left(-2T\gamma^2\right)$ and is 0 as long as $T > \frac{\ln N}{2\gamma^2}$.

Proof. Let $F(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$ so that $H(x) = \textsc{sgn}(F(x))$. The first step is to realize 0-1 loss is bounded by the exponential loss

$$\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{H(x_i) \neq y_i\} = \sum_{i=1}^{N} p_1(i)\mathbf{1}\left\{y_i F(x_i) \leq 0\right\} \leq \sum_{i=1}^{N} p_1(i)\exp\left(-y_i F(x_i)\right).$$

Now notice that the update rule of p_t can be written as $p_{t+1}(i) = p_t(i)\exp(-y_i\alpha_t h_t(x_i))/Z_t$ where Z_t is the normalization factor. We then have

$$p_{T+1}(i) = p_1(i)\prod_{t=1}^{T}\frac{\exp(-y_i\alpha_t h_t(x_i))}{Z_t} = \frac{p_1(i)\exp\left(-y_i F(x_i)\right)}{\prod_{t=1}^{T} Z_t}$$

and therefore the error rate is bounded by $\sum_{i=1}^{N}\left(p_{T+1}(i)\prod_{t=1}^{T} Z_t\right) = \prod_{t=1}^{T} Z_t$. It remains to bound each Z_t:

$$\begin{aligned} Z_t &= \sum_{i=1}^{N} p_t(i)\exp(-y_i\alpha_t h_t(x_i)) \\ &= \sum_{i:h_t(x_i)=y_i} p_t(i)\exp(-\alpha_t) + \sum_{i:h_t(x_i)\neq y_i} p_t(i)\exp(\alpha_t) \\ &= (1-\epsilon_t)\exp(-\alpha_t) + \epsilon_t\exp(\alpha_t) \\ &= \sqrt{1-4\gamma_t^2} \leq \exp(-2\gamma_t^2). \end{aligned}$$

where the last equality is by the definition of α_t (which is in fact chosen to minimize the expression), and the last step is by the fact $1 + z \leq e^z$. This finishes the proof for Eq. (2). Under weak learning assumption, we further have $\gamma_t \geq \gamma$ and thus the stated bound $\exp\left(-2T\gamma^2\right)$. Finally, note that as soon as the error rate drops below $1/N$, it must become zero. Therefore, as long as $\exp\left(-2T\gamma^2\right) < 1/N$, which means $T > \frac{\ln N}{2\gamma^2}$, AdaBoost ensures zero training error. □

Of course, at the end of the day one cares about the generalization error instead of the training error. Standard VC theory says that the difference between the generalization error and the training error is bounded by something like $\widetilde{\mathcal{O}}(\sqrt{C/N})$ where C is some complexity measure of the hypothesis space. For boosting, it is not so surprising that this complexity is growing as $O(T)$, since combining more weak classifiers leads to a more complicated final classifier H. Since we just proved that after $T > \frac{\ln N}{2\gamma^2}$ rounds the training error of AdaBoost is zero, it means that if we stop the algorithm at the right time, AdaBoost will have generalization error of order $\widetilde{\mathcal{O}}\left(\sqrt{\frac{\ln N}{\gamma^2 N}}\right)$, which can be arbitrarily small when N is large enough. In other words, under the weak learning assumption AdaBoost does ensure arbitrarily small generalization error given enough examples, implying that "boosting" is indeed possible, or in more technical terms, weak learnability is equivalent to strong learnability.

Preventing Overfitting. Like all other machine learning algorithms, AdaBoost could also overfit the training set. This is consistent with the VC theory: if we keep running the algorithm even after the training error has dropped to zero, then the generalization error is of order $\sqrt{T/N}$, which is increasing in the number of rounds T.

However, what usually happens in practice is that AdaBoost tends to prevent overfitting, in the sense that even if one keeps running the algorithm for many more rounds after the training error has dropped to zero, the generalization error still keeps decreasing. How should we explain this?

It turns out that the concept of *margin* is the key for understanding this phenomenon. The margin of an example (x, y) (with respect to the classifier H) is defined as $yf(x)$ where

$$f(x) = \frac{F(x)}{\sum_{t=1}^{T} \alpha_t} = \sum_{t=1}^{T} \left(\frac{\alpha_t}{\sum_{\tau=1}^{T} \alpha_\tau} \right) h_t(x).$$

It is clear that the margin is always in $[-1, 1]$, and the sign of the margin indicates whether the final classifier H makes a mistake on x or not. Specifically we want the margin to be positive in order to have low error rate. However, intuitively we also want the margin to be a large positive (that is, close to 1), since in this case there is a huge difference between the vote for $+1$ and -1 (recall H is just doing a weighted majority vote), and the decisive win of one label makes us feel more confident about the final prediction.

Indeed, margin theory says that for any θ, the generalization error of H and the fraction of training examples with margin at most θ are related as follows:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathbf{1}\{H(x) \neq y\}] \leq \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{y_i f(x_i) \leq \theta\} + \widetilde{\mathcal{O}}\left(\frac{1}{\theta} \sqrt{\frac{C_{\mathcal{H}}}{N}} \right)$$

where $C_{\mathcal{H}}$ is the complexity of the hypothesis space $\mathcal{H}$ used by the oracle (recall $h_t \in \mathcal{H}$), which is independent of the number of weak classifiers combined by H (that is, T)! Therefore, if keep running AdaBoost has the effect of increasing margin even after the training error drops to zero, then this explains why overfitting does not happen. This is indeed true: under the weak learning assumption AdaBoost guarantees

$$\frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{y_i f(x_i) \leq \theta\} \leq \left(\sqrt{(1-2\gamma)^{1-\theta}(1+2\gamma)^{1+\theta}} \right)^T.$$

One way to interpret this bound is to note that as long as θ is such that $(1-2\gamma)^{1-\theta}(1+2\gamma)^{1+\theta} < 1$, which translates to $\theta \leq \Gamma(\gamma) \stackrel{\text{def}}{=} \frac{-\ln(1-4\gamma^2)}{\ln\left(\frac{1+2\gamma}{1-2\gamma}\right)} \leq 2\gamma$, then the fraction of examples with margin at most θ is eventually zero when T is large enough. In other words, if we keep running AdaBoost, eventually the smallest margin is $\Gamma(\gamma)$ and the generalization error is thus $\widetilde{\mathcal{O}}\left(\frac{1}{\Gamma(\gamma)} \sqrt{\frac{C_{\mathcal{H}}}{N}} \right)$.

As a final remark, interestingly AdaBoost is not doing the best possible job in maximizing the smallest margin. The best possible smallest margin under the weak learning assumption can be shown to be exactly 2γ.

2 Boosting via Online Learning

In this section we show a simple reduction from boosting to an expert problem. In other words, given an expert algorithm as a blackbox, we can turn it into a boosting algorithm, as shown below.

Algorithm 2: Boosting to Expert Problem

Input: training set $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, weak learning oracle $\mathcal{A}$, expert algorithm $\mathcal{E}$
Initialize the expert algorithm $\mathcal{E}$ with N experts
for $t = 1, \ldots, T$ **do**
  let p_t be the prediction of $\mathcal{E}$ at round t
  invoke the oracle to get $h_t = \mathcal{A}(S, p_t)$
  feed the loss vector ℓ_t to $\mathcal{E}$ where $\ell_t(i) = \mathbf{1}\{h_t(x_i) = y_i\}$
output the final classifier: $H(x) = \textsc{sgn}\left(\sum_{t=1}^{T} h_t(x)\right)$.

While one might imagine that it is natural to treat each possible weak classifier $h \in \mathcal{H}$ as an expert, the reduction above actually treats each training example as an expert. Moreover, the reduction punishes an expert/example by assigning loss 1 if it is *correctly* classified, which seems counter-intuitive at first glance but is in fact consistent with the key idea of boosting, that is, more focus should be put on "hard" examples.

Now suppose the regret of $\mathcal{E}$ is $\mathcal{R}_T$, we have by construction

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N} p_t(i)\mathbf{1}\{h_t(x_i) = y_i\} \le \frac{1}{T}\sum_{t=1}^{T}\mathbf{1}\{h_t(x_j) = y_j\} + \frac{\mathcal{R}_T}{T}$$

for any $j \in [N]$. On the other hand, reusing the notation $\epsilon_t = \mathbb{E}_{i\sim p}[\mathbf{1}\{h(x_i) \neq y_i\}]$ and $\gamma_t = 1/2 - \epsilon_t$, we have under the weak learning assumption

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N} p_t(i)\mathbf{1}\{h_t(x_i) = y_i\} = \frac{1}{T}\sum_{t=1}^{T}(1-\epsilon_t) = \frac{1}{2} + \frac{1}{T}\sum_{t=1}^{T}\gamma_t \ge \frac{1}{2} + \gamma.$$

Combining leads to

$$\frac{1}{2} + \gamma \le \frac{1}{T}\sum_{t=1}^{T}\mathbf{1}\{h_t(x_j) = y_j\} + \frac{\mathcal{R}_T}{T}. \tag{3}$$

Therefore, as long as $\gamma > \frac{\mathcal{R}_T}{T}$, we have $\frac{1}{2} < \frac{1}{T}\sum_{t=1}^{T}\mathbf{1}\{h_t(x_j) = y_j\}$, implying that more than half of the weak classifiers predicts correctly. Since the final classifier H is a simple majority vote, this also implies that H has zero training error. Plugging the optimal regret $\mathcal{R}_T = \mathcal{O}(\sqrt{T\ln N})$ (for example by using Hedge), we thus only need $T > \mathcal{O}(\frac{\ln N}{\gamma^2})$ rounds to achieve zero training error, same as AdaBoost.

Moreover, since $\mathbf{1}\{h(x) = y\} = \frac{1}{2}(yh(x) + 1)$, plugging this fact into Eq. (3) and simplifying lead to (with $f(x) = \frac{1}{T}\sum_{t=1}^{T} h_t(x)$)

$$2\gamma - \frac{2\mathcal{R}_T}{T} \le y_j f(x_j).$$

Note that $y_j f(x_j)$ is exactly the margin for example x_j. The above is therefore saying that if $\mathcal{R}_T$ is sublinear and we keep running the algorithm, eventually every example will have margin above something arbitrarily close to 2γ, the optimal margin as mentioned in the last section. In other words, this expert-algorithm-turned boosting is doing an even better job in maximizing margin compared to AdaBoost.

Finally, we point out how adaptive quantile regret bound can say something more meaningful in this context. For any margin threshold θ, what does the above derivation tell us about the fraction of examples with margin at most θ, that is, $\frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\{y_i f(x_i) \le \theta\}$? In fact, it only says that if $\theta \le 2\gamma - \frac{2\mathcal{R}_T}{T}$ then the fraction is zero, otherwise it could be as large as 1, a trivial bound. However suppose the expert algorithm admits quantile regret bounds (such as Squint), then assuming

$y_1 f(x_1) \leq \cdots \leq y_N f(x_N)$ without loss of generality, we have

$$2\gamma - \mathcal{O}\left(\sqrt{\frac{\ln(N/j)}{T}}\right) \leq y_j f(x_j).$$

Therefore for any $\theta < 2\gamma$, we can just find the largest j_θ such that $2\gamma - \mathcal{O}\left(\sqrt{\frac{\ln(N/j_\theta)}{T}}\right) \leq \theta$, and assert that the fraction of examples with margin at most θ will be j_θ/N, which is of order $\exp(-\Omega(T(2\gamma-\theta)^2))$.

References

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

Lecture 9

Instructor: Haipeng Luo

1 Interval Regret

In the following lectures, we will discuss how to evaluate online learning algorithms using measures that are more challenging than the classic notion of regret and also make more sense when dealing with non-stationary environments. We first focus on one of these measures: *interval regret*, in the general OCO setting for this lecture.

Recall that the classic regret compares the loss of the algorithm to the loss of the best fixed point. One natural question that we have deferred discussing until now is: is the loss of the best fixed point necessarily a good benchmark to compare to? The answer is no, especially not in some non-stationary environments. To see this, consider a simple expert problem with two experts where the first one always suffers loss 0 for the first $T/2$ rounds and loss 1 for the other $T/2$ rounds, and the situation for the second expert is exactly reversed. Then overall the best fixed expert (which is any one of the two experts) has loss $T/2$, and thus even if the regret to the best fixed expert is zero, all we can say is that the loss of the algorithm is bounded by $T/2$, not a very impressive guarantee.

Moreover, this is not just due to lazy analysis that does not give tight enough bounds. One can show that some algorithms with sublinear regret guarantees indeed suffer linear loss $\Omega(T)$ is this case. Take Hedge as an example. Observe that by the algorithm the weight for the first expert is always not smaller than the second one (since the cumulative loss of the first expert is always not larger). This means that for the second $T/2$ rounds, the loss of the algorithm is at least $1/2$, and thus the cumulative loss is at least $T/4$.

Therefore, classic regret is not the always the right objective to minimize, especially in a non-stationary environments where there is no single fixed point that does well overall. To address this issue, we introduce the notion of interval regret. Specifically, we use the notation $\mathcal{I} = [s, e]$ to denote the rounds $\{s, s+1, \ldots, e-1, e\}$ and call it an interval. Then the interval regret with respect to an interval $\mathcal{I}$ is simply (and literally) the regret on this interval

$$\mathcal{R}_{\mathcal{I}} = \sum_{t \in \mathcal{I}} f_t(w_t) - \min_{w \in \Omega} \sum_{t \in \mathcal{I}} f_t(w).$$

In other words, $\mathcal{R}_{\mathcal{I}}$ is comparing the loss of the algorithm on interval $\mathcal{I}$ to the loss of the best fixed point in terms of the cumulative loss on interval $\mathcal{I}$. Imagine we know where the starting point s of the interval $\mathcal{I}$ is, then we would simply run an online learning algorithm with sublinear (regular) regret starting from round s and obtain $\mathcal{R}_{\mathcal{I}} = \mathcal{O}(\sqrt{|\mathcal{I}|})$ (omitting other terms) where we use $|\mathcal{I}|$ to denote the length of interval $\mathcal{I}$. Of course, the challenge is that we do not know what $\mathcal{I}$ is beforehand, or in other words, we want to design an algorithm with interval regret $\mathcal{R}_{\mathcal{I}} = \mathcal{O}(\sqrt{|\mathcal{I}|})$ *simultaneously* for all $\mathcal{I}$. In the literature, such an algorithm is also sometimes called a *strongly adaptive* algorithm.

Going back to the illustrating example discussed above. If we have a strongly adaptive algorithm, then we can conclude that the interval regret for the first $T/2$ rounds and the second $T/2$ rounds are both of order $\mathcal{O}(\sqrt{T})$. More importantly, since the best expert on these two intervals (expert 1 and 2 respectively) both have zero cumulative loss on their respective interval, it means that the cumulative loss of such strongly adaptive algorithm over T rounds is only $\mathcal{O}(\sqrt{T})$, much better than Hedge for example.

2 Sleeping Experts

How should we design strongly adaptive algorithms? It turns out that there is a general mechanism that allows one to turn any algorithm with low (regular) regret into a strongly adaptive algorithm. To introduce this approach, we will need to take a detour and discuss the *sleeping experts* [Freund et al., 1997] problem first.

The sleeping expert problem is a generalization of the expert problem where experts with different expertise can choose to abstain from providing advice for a given round. Formally, at round $t = 1, \dots, T$,

1. the environment first decides (possibly adversarially) which of the N experts are awake and which are asleep: $a_t(i) = 1$ means expert i is awake and $a_t(i) = 0$ means it is asleep;
2. a_t is then revealed to the learner who needs to decide a distribution $p_t \in \Delta(N)$ with the restriction that no weights are put on asleep experts, that is, $p_t(i) = 0$ if $a_t(i) = 0$;
3. the environment reveals the losses for the awake experts, that is, $\ell_t(i)$ for i s.t. $a_t(i) = 1$.

The regular expert problem is clearly a special case with $a_t(i) = 1$ for all t and i. Because an expert is now not necessarily involved in every round of the game, the regret against this expert is naturally defined only in terms of those rounds when the expert is awake:

$$R_T(i) = \sum_{t:a_t(i)=1} \left(\langle p_t, \ell_t \rangle - \ell_t(i)\right).$$

Note that while we use the notation $\ell_t \in [0,1]^N$, some of it coordinates might not be defined since the corresponding experts could be asleep on round t. However, this is not really an issue because p_t is required to put zero weight on those coordinates anyway and thus they make no difference to the inner product $\langle p_t, \ell_t \rangle$.

It is natural to ask for a sleeping experts algorithm with $R_T(i) = \mathcal{O}(\sqrt{|\{t : a_t(i) = 1\}| \ln N})$ for all i. Indeed, this is achievable by reducing the sleeping expert problem to the regular expert problem. Specifically, suppose we are given an regular expert algorithm $\mathcal{E}$ with prediction $\widehat{p}_t$ on round t. To come up with a prediction p_t for the sleeping expert problem so that $p_t(i) = 0$ for those asleep experts, a natural idea is to simply ignore the weights in $\widehat{p}_t$ for asleep experts and renormalize the others, that is, $p_t(i) \propto a_t(i)\widehat{p}_t(i)$.

Next we need to come up with a loss vector as the feedback to $\mathcal{E}$. For the awake experts, it is natural to just use the same loss we observe in the sleeping expert problem. What about the asleep experts? Suppose we assign the same value x to all these asleep experts, with the goal of forcing the loss of $\mathcal{E}$ for this round to be the same as the loss of the sleeping expert algorithm we arrive at an equation

$$\sum_{i:a_t(i)=1} \widehat{p}_t(i)\ell_t(i) + \left(\sum_{i:a_t(i)=0} \widehat{p}_t(i)\right) x = \sum_{i:a_t(i)=1} p_t(i)\ell_t(i).$$

Using the definition of p_t and solving for x give

$$\begin{aligned} x &= \frac{\sum_{i:a_t(i)=1} \left(p_t(i) - \widehat{p}_t(i)\right) \ell_t(i)}{\sum_{i:a_t(i)=0} \widehat{p}_t(i)} \\ &= \frac{\left(\frac{1}{\sum_{i:a_t(i)=1} \widehat{p}_t(i)} - 1\right) \sum_{i:a_t(i)=1} \widehat{p}_t(i)\ell_t(i)}{1 - \sum_{i:a_t(i)=1} \widehat{p}_t(i)} \\ &= \sum_{i:a_t(i)=1} p_t(i)\ell_t(i), \end{aligned}$$

which means the "fake" loss of asleep experts should be exactly the loss of the sleeping expert algorithm! As discussed earlier, the value of $\ell_t(i)$ for $a_t(i) = 0$ does not matter in the sleeping expert problem. We will therefore overload the notation ℓ_t to denote the loss vector for both the regular expert problem and the sleeping expert problem, where $\ell_t(i) = \sum_{i:a_t(i)=1} p_t(i)\ell_t(i) = \langle p_t, \ell_t \rangle$ if $a_t(i) = 0$ and otherwise is the loss revealed by the environment. The complete reduction is shown in Algorithm 1.

Algorithm 1: Reduction from Sleeping Expert to Regular Expert

Input: a regular expert algorithm $\mathcal{E}$
for $t = 1, \dots, T$ **do**
  let $\widehat{p}_t$ be the prediction of $\mathcal{E}$ on round t
  observe a_t from the environment
  play p_t such that $p_t(i) \propto a_t(i)\widehat{p}_t(i)$
  observe $\ell_t(i)$ for i such that $a_t(i) = 1$
  set $\ell_t(i) = \langle p_t, \ell_t \rangle$ for i such that $a_t(i) = 0$
  pass ℓ_t to $\mathcal{E}$

By the reduction, we have

$$R_T(i) = \sum_{t:a_t(i)=1} (\langle p_t, \ell_t \rangle - \ell_t(i)) = \sum_{t=1}^{T} (\langle p_t, \ell_t \rangle - \ell_t(i)) = \sum_{t=1}^{T} (\langle \widehat{p}_t, \ell_t \rangle - \ell_t(i))$$

which is exactly the regret of $\mathcal{E}$ against expert i in the regular expert problem (and therefore we overload the notation $R_T(i)$ too). However, if $\mathcal{E}$ only has regret bound $\mathcal{O}(\sqrt{T \ln N})$ (such as Hedge), then we only obtain $R_T(i) = \mathcal{O}(\sqrt{T \ln N})$ instead of the desired bound $\mathcal{O}(\sqrt{|\{t : a_t(i) = 1\}| \ln N})$. In fact, what we need here is an expert algorithm with adaptive regret bounds such as $R_T(i) = \mathcal{O}(\sqrt{(\sum_t r_t^2(i)) \ln N})$ where $r_t(i) = \langle \widehat{p}_t, \ell_t \rangle - \ell_t(i)$ is the instantaneous regret. Indeed, with such an adaptive regret bound we arrive at

$$R_T(i) = \mathcal{O}\left(\sqrt{\left(\sum_{t=1}^{T} (\langle \widehat{p}_t, \ell_t \rangle - \ell_t(i))^2\right) \ln N}\right) = \mathcal{O}\left(\sqrt{\left(\sum_{t:a_t(i)=1} (\langle p_t, \ell_t \rangle - \ell_t(i))^2\right) \ln N}\right)$$

which is of order $\mathcal{O}(\sqrt{|\{t : a_t(i) = 1\}| \ln N})$. Such an adaptive regret bound is not unfamiliar to us – we have shown that Squint enjoys exactly such bound (and have in fact discussed several interesting consequences of having such adaptive bounds). Plugging the general Squint regret bound, we have for any T and any competitor $q \in \Delta(N)$ (omitting small terms),

$$\mathbb{E}_{i\sim q}[R_T(i)] = \mathcal{O}\left(\sqrt{(\mathbb{E}_{i\sim q}[|\{t : a_t(i) = 1\}|]) \operatorname{KL}(q, \widehat{p}_1)}\right) \tag{1}$$

where $\widehat{p}_1$ is a prior distribution. Note that the term $|\{t : a_t(i) = 1\}|$ can be even improved to the loss of expert i: $\sum_{t:a_t(i)=1} \ell_t(i)$ by the same argument we have used to show "small-loss" bounds for Squint.

As another remark, recall that Squint is completely parameter-free – it does not even need to know the total number of experts N in advance. This sounds strange in the regular expert problem since the algorithm needs to compute a distribution in $\Delta(N)$ and thus of course needs to and will know N. However, it does make more sense in the sleeping expert setting if the total number of experts is not given ahead of time. We will see such an example immediately.

3 Strongly Adaptive Algorithms via Sleeping Expert

We are now ready to introduce a general mechanism to derive strongly adaptive algorithms in the OCO setting given any OCO algorithm $\mathcal{A}$ with regular regret $\mathcal{O}(\sqrt{T})$ for all T.[1] As mentioned earlier, if the interval $\mathcal{I} = [s, e]$ was known, one could simply run $\mathcal{A}$ starting at time s. Now since we want to consider all intervals $\mathcal{I}$, a natural idea is to start a new instance of $\mathcal{A}$ at the beginning of every round and to combine the predictions from different instances to come up with the final prediction.

This can be exactly captured by the sleeping expert problem: each instance of $\mathcal{A}$ is an expert, and the instance that starts at time t (denoted by $\mathcal{A}_t$) is asleep for the first $t-1$ rounds and awake for the

[1] If an algorithm requires knowing T, then a simple doubling trick can make it agnostic to T.

Algorithm 2: Strongly Adaptive Algorithm via Sleeping Expert

Input: a regular OCO algorithm $\mathcal{A}$, a sleeping expert algorithm $\mathcal{S}$
for $t = 1, \dots, T$ **do**
  start a new instance of $\mathcal{A}$, called $\mathcal{A}_t$
  obtain predictions from $\mathcal{A}_1, \dots, \mathcal{A}_t$, denoted by $w_t^1, \dots, w_t^t$
  pass a_t to $\mathcal{S}$ where $a_t(i) = 1$ for $i \le t$ and $a_t(i) = 0$ for $i > t$
  obtain distribution p_t from $\mathcal{S}$
  predict $w_t = \sum_{i=1}^t p_t(i) w_t^i$
  observe loss function f_t, suffer loss $f_t(w_t)$
  pass f_t to $\mathcal{A}_1, \dots, \mathcal{A}_t$
  pass ℓ_t to $\mathcal{S}$ where $\ell_t(i) = f_t(w_t^i)$ for $i \le t$.

rest of the game. The final prediction at round t will be the convex combination of predictions from $\mathcal{A}_1, \dots, \mathcal{A}_t$ according to the distribution decided by a sleeping expert algorithm. See Algorithm 2 for details. By the construction, we have for any $w \in \Omega$ and interval $\mathcal{I} = [s, e]$,

$$\begin{aligned}
\sum_{t \in \mathcal{I}} (f_t(w_t) - f_t(w)) &\le \sum_{t \in \mathcal{I}} \left(\sum_{i=1}^t p_t(i) f_t(w_t^i) - f_t(w) \right) && \text{(Jensen's inequality)} \\
&= \sum_{t \in \mathcal{I}} (\langle p_t, \ell_t \rangle - \ell_t(s)) + \sum_{t \in \mathcal{I}} (f_t(w_t^s) - f_t(w)) \\
&= R_e(s) + \mathcal{O}(\sqrt{|\mathcal{I}|}) && \text{(by the guarantee of } \mathcal{A}) \\
&= \mathcal{O}(\sqrt{|\mathcal{I}| \ln T}) + \mathcal{O}(\sqrt{|\mathcal{I}|}). && \text{(by the guarantee of } \mathcal{S})
\end{aligned}$$

In fact, by using Squint as the sleeping expert algorithm and a special prior $\widehat{p}_1(i) \propto 1/i^2$, we can improve the first term to $\mathcal{O}(\sqrt{|\mathcal{I}| \ln s})$ since

$$\mathrm{KL}(q, \widehat{p}_1) = \ln \left(\frac{1}{\widehat{p}_1(s)} \right) = \ln \left(s^2 \sum_{i=1}^\infty \frac{1}{i^2} \right) = \mathcal{O}(\ln s).$$

Note that this is a concrete example where the total number of the experts is unknown ahead of time and keeps increasing, but it is clear that Squint can be run without any trouble.

Finally we discuss computational efficiency of Algorithm 2. Since we need to maintain t instances of $\mathcal{A}$ and thus t experts at time t, the running time per round is $\mathcal{O}(t)$, which is not very efficient and keeps increasing. Fortunately, it turns out that one can significantly improve the running time to $\mathcal{O}(\ln t)$ without sacrificing any regret guarantee. The idea is to kill some instances when they have lived long enough in some sense so that at each time there are only $\mathcal{O}(\ln t)$ instances alive. Killing an instance can be easily incorporated in the sleeping expert model by just putting the corresponding expert to sleep forever. The key is to do this in a careful way so that the regret is almost not affected.

There are in fact many different ways to do this. One simple approach taken from [Hazan and Seshadhri, 2007] is to let $\mathcal{A}_t$ live for $2^{d(t)}$ rounds where $d(t)$ is the largest integer such that $t = b(t) \times 2^{d(t)}$ for some (odd) integer $b(t)$. In other words, $d(t)$ is the number of 2's in t's prime factorization. (Try drawing a picture to see what the awake intervals are like for the first few (say 20) instances.)

To see why this leads to a more efficient algorithm, first note that at any time t and any integer d, there is at most one expert with lifetime 2^d awake (think about why). On the other hand, at time t the longest lifetime of any awake experts is clearly bounded by $2^{\lfloor \log_2 t \rfloor}$. Therefore, at any time t the total number of awake experts is at most $\lfloor \log_2 t \rfloor + 1$, meaning that the running time of the algorithm is only $\mathcal{O}(\ln t)$ per iteration.

Next we argue that the regret is almost not affected. For an interval $\mathcal{I} = [s, e]$, since $\mathcal{A}_s$ does not necessarily live until the end of this interval and we thus cannot just compare to $\mathcal{A}_s$ as before, it is natural to divide the interval into several disjoint and consecutive subintervals and compare to different instances of $\mathcal{A}$ on these subintervals. Formally, let $\mathcal{I}_m = [s_m, e_m]$ for $m = 1, \dots, M$ be

these subintervals where $s_1 = s, s_m = e_{m-1} + 1$ for $1 < m \leq M$, $e_m = s_m + 2^{d(s_m)} - 1$ for $1 \leq m < M$ and $e_M = e$. Clearly for each $\mathcal{I}_m$ $(m < M)$, there is an instance $(\mathcal{A}_{s_m})$ that is run solely on this interval. More importantly, the length of these intervals is doubling in the sense that $2|\mathcal{I}_m| \leq |\mathcal{I}_{m+1}|$ for $1 \leq m < M - 2$. This is because

$$s_{m+1} = e_m + 1 = s_m + 2^{d(s_m)} = (b(s_m) + 1) \times 2^{d(s_m)} = \frac{b(s_m) + 1}{2} \times 2^{d(s_m)+1}$$

where $\frac{b(s_m)+1}{2}$ has to be an integer since $b(s_m)$ is odd. This implies that $d(s_{m+1}) \geq d(s_m) + 1$ and thus $2|\mathcal{I}_m| \leq |\mathcal{I}_{m+1}|$. As a result, we have

$$\begin{aligned}
\sum_{t\in\mathcal{I}}(f_t(w_t) - f_t(w)) &\leq \sum_{t\in\mathcal{I}}\left(\sum_{i=1}^{t} p_t(i) f_t(w_t^i) - f_t(w)\right) && \text{(Jensen's inequality)} \\
&= \sum_{m=1}^{M}\sum_{t\in\mathcal{I}_m}(\langle p_t, \ell_t\rangle - \ell_t(s_m)) + \sum_{m=1}^{M}\sum_{t\in\mathcal{I}_m}(f_t(w_t^{s_m}) - f_t(w)) \\
&= \sum_{m=1}^{M} R_{e_m}(s_m) + \sum_{i=1}^{M}\mathcal{O}(\sqrt{|\mathcal{I}_m|}) && \text{(by the guarantee of } \mathcal{A}) \\
&= \sum_{m=1}^{M}\mathcal{O}(\sqrt{|\mathcal{I}_m|\ln T}) && \text{(by the guarantee of } \mathcal{S}) \\
&\leq \sum_{i=0}^{\infty}\mathcal{O}(\sqrt{2^{-i}|\mathcal{I}|\ln T}) \\
&= \mathcal{O}(\sqrt{|\mathcal{I}|\ln T}),
\end{aligned}$$

giving the same result (up to constants) as before.

References

Yoav Freund, Robert E Schapire, Yoram Singer, and Manfred K Warmuth. Using and combining predictors that specialize. In *29th Annual ACM Symposium on the Theory of Computing*, pages 334–343. ACM, 1997.

Elad Hazan and C. Seshadhri. Adaptive algorithms for online decision problems. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 14, 2007.

Lecture 10

Instructor: Haipeng Luo

1 Dynamic Regret

In the previous lecture we discussed interval regret which only considers regret on a certain time interval, or in other words, it only considers the performance of the algorithm in a local region. In this lecture we will discuss how to measure the global performance of an algorithm over T rounds, still under some non-stationary environment where the best single fixed point in hindsight is not the right benchmark to compare to.

The most ambitious goal would be to compare with the best decision $w_t^\star = \operatorname{argmin}_{w\in\Omega} f_t(w)$ *for each round*. Equivalently, this is the same as asking for sublinear regret against all competitor sequence $u_1, \ldots, u_T \in \Omega$:

$$\mathcal{R}_T(u_1, \ldots, u_T) \overset{\text{def}}{=} \sum_{t=1}^{T}(f_t(w_t) - f_t(u_t)) = o(T).$$

This is the so-called *dynamic regret*. Perhaps not so surprisingly, sublinear dynamic regret is not achievable in general, especially in an adversarial setting. To see this, simply think about a 2-expert problem where the environment always assigns loss 0 to the expert that has the highest weight $p_t(i)$ from the algorithm and loss 1 to the other expert (recall that the environment sees p_t before assigning losses). Then on each round, clearly the best expert has loss 0, while the algorithm suffers loss at least $1/2$, leading to linear dynamic regret.

However, this does not exclude many interesting situations where sublinear regret is still possible. The first situation we consider is when the competitors stay the same most of time, that is, $\sum_{t=2}^{T} \mathbf{1}\{u_t \neq u_{t-1}\} \leq S - 1$ for some constant S. In other words, we allow the benchmark to divide the total T rounds into S disjoint intervals, and to select the best fixed point on each interval. This is sometimes called the switching regret or tracking regret. The regular regret is clearly a special case when $S = 1$.

Such benchmark can be significantly better than a single fixed point (think about the 2-expert example from last lecture). Such locally stationary environments also appear in practice naturally. For example, think about the problem of product recommendation. It is often the case that data from, say each month, stays stationary and thus comparing to a best fixed decision for each month is pretty reasonable.

How should we obtain small switching regret? In fact, we have already solved this problem implicitly. Indeed, if we have a strongly adaptive algorithm with regret $\mathcal{O}(\sqrt{|\mathcal{I}|\ln T})$ for any interval $\mathcal{I}$, then just by running the exact same algorithm we have by dividing the T rounds into S intervals $\mathcal{I}_1, \ldots, \mathcal{I}_S$ so that the competitor stays the same on each interval,

$$\begin{aligned}\mathcal{R}_T(u_1, \ldots, u_T) &= \sum_{m=1}^{S}\sum_{t\in\mathcal{I}_m}(f_t(w_t) - f_t(u_t)) = \mathcal{O}\left(\sum_{m=1}^{S}\sqrt{|\mathcal{I}_m|\ln T}\right) \\ &\leq \mathcal{O}\left(\sqrt{S\sum_{m=1}^{S}|\mathcal{I}_m|\ln T}\right) = \mathcal{O}(\sqrt{ST\ln T})\end{aligned}$$

where the inequality is by Cauchy-Schwarz inequality. Therefore, as long as S is sublinear in T, the switching regret is also sublinear.

Next we consider an even more general situation where the competitors do not need to stay the same locally. Instead, the dynamic regret will depend on the variation of the loss functions $\ell_1, \dots, \ell_T$, defined as

$$V_T = \sum_{t=2}^{T} \max_{w \in \Omega} |f_t(w) - f_{t-1}(w)|.$$

Small variation implies that the environment is slowly drifting over time and therefore learning is possible. This is very similar to the path length that we discussed before. Previously we derive adaptive regular regret in terms of path length with the hope of getting regret smaller than $\sqrt{T}$, and now we derive dynamic regret in terms of variation with the goal of avoiding linear regret.

Interestingly, this problem is again already solved by using a strongly adaptive algorithm.

Theorem 1 ([Zhang et al., 2017])**.** *A strongly adaptive algorithm with $\mathcal{R}_{\mathcal{I}} = \mathcal{O}(\sqrt{|\mathcal{I}| \ln T})$ for any interval $\mathcal{I}$ ensures*

$$\mathcal{R}_T(w_1^\star, \dots, w_T^\star) = \mathcal{O}\left(T^{\frac{2}{3}}(V_T \ln T)^{\frac{1}{3}}\right)$$

where $w_t^\star = \operatorname{argmin}_{w \in \Omega} f_t(w)$.

Proof. Let $\mathcal{I}_1 = [s_1, e_1], \dots, \mathcal{I}_M = [s_M, e_M]$ be any partition of the whole game $[1, T]$, and

$$V_{\mathcal{I}_m} = \sum_{t=s_m+1}^{e_m} \max_{w \in \Omega} |f_t(w) - f_{t-1}(w)|$$

be the variation of interval $\mathcal{I}_m$. For any $m \in [M]$, we have

$$\begin{aligned}\sum_{t \in \mathcal{I}_m} (f_t(w_t) - f_t(w_t^\star)) &= \sum_{t \in \mathcal{I}_m} (f_t(w_t) - f_t(w_{s_m}^\star)) + \sum_{t \in \mathcal{I}_m} (f_t(w_{s_m}^\star) - f_t(w_t^\star)) \\ &\le \mathcal{O}(\sqrt{|\mathcal{I}_m| \ln T}) + 2|\mathcal{I}_m| V_{\mathcal{I}_m},\end{aligned}$$

where the last step is by the guarantee of the strongly adaptive algorithm and the fact

$$\begin{aligned}f_t(w_{s_m}^\star) - f_t(w_t^\star) &\le f_t(w_{s_m}^\star) - f_{s_m}(w_{s_m}^\star) + f_{s_m}(w_t^\star) - f_t(w_t^\star) \qquad \text{(by optimality of } w_{s_m}^\star\text{)} \\ &= \sum_{\tau=s_m+1}^{t} \left(f_\tau(w_{s_m}^\star) - f_{\tau-1}(w_{s_m}^\star)\right) + \sum_{\tau=s_m+1}^{t} (f_{\tau-1}(w_t^\star) - f_\tau(w_t^\star)) \\ &\le 2 \sum_{\tau=s_m+1}^{t} \max_{w \in \Omega} |f_\tau(w) - f_{\tau-1}(w)| \le 2V_{\mathcal{I}_m}.\end{aligned}$$

Therefore, the dynamic regret is bounded by

$$\begin{aligned}\mathcal{R}_T(w_1^\star, \dots, w_T^\star) &\le \sum_{m=1}^{M} \mathcal{O}(\sqrt{|\mathcal{I}_m| \ln T}) + 2|\mathcal{I}_m| V_{\mathcal{I}_m} \\ &\le \mathcal{O}(\sqrt{MT \ln T}) + 2 \max_m |\mathcal{I}_m| \sum_{m=1}^{M} V_{\mathcal{I}_m} \\ &\le \mathcal{O}(\sqrt{MT \ln T}) + 2 \max_m |\mathcal{I}_m| V_T.\end{aligned}$$

Finally we just need to balance M and $\max_m |\mathcal{I}_m|$. For a fixed M, it is clear that if we want to minimize $\max_m |\mathcal{I}_m|$, we should divide the game (almost) evenly into M intervals so that $\max_m |\mathcal{I}_m| = \mathcal{O}(T/M)$ and $\mathcal{R}_T(w_1^\star, \dots, w_T^\star) = \mathcal{O}(\sqrt{MT \ln T} + TV_T/M)$. Setting M optimally to $\lfloor (T/\ln T)^{\frac{1}{3}} V_T^{\frac{2}{3}} \rfloor$ finishes the proof. □

According to the theorem, as long as V_T is sublinear, the dynamic regret is sublinear. Note that this does not contradict with the earlier impossibility result since V_T can be linear in T in the worst case.

The bound $\mathcal{O}(T^{\frac{2}{3}}(V\ln T)^{\frac{1}{3}})$ is worst-case optimal as shown in [Besbes et al., 2015], but it is not always tight. For example, suppose f_t stays the same most of the time except for $S-1$ rounds. Then assuming $f_t(w)\in[0,1]$ we have $V_T = S-1$ and thus $\mathcal{R}_T(w_1^\star,\dots,w_T^\star) = \mathcal{O}(T^{\frac{2}{3}}(S\ln T)^{\frac{1}{3}})$. However, in this case the dynamic regret is clearly also the switching regret and we have showed $\mathcal{O}(\sqrt{ST\ln T})$ is possible (and better), and more importantly achieved by using the exact same strongly adaptive algorithm. Therefore, we can instead write the bound as

$$\mathcal{R}_T(w_1^\star,\dots,w_T^\star) = \mathcal{O}\left(\min\left\{T^{\frac{2}{3}}(V_T\ln T)^{\frac{1}{3}},\ \sqrt{\left(\sum_{t=2}^T \mathbf{1}\{w_t^\star\neq w_{t-1}^\star\}\right)T\ln T}\right\}\right).$$

2 Dynamic Regret for the Expert Problem

All the results discussed above apply to the special case of the expert problem of course, but in this section we will introduce a different type of dynamic regret that is more specific to the expert problem. First note that in this case the dynamic regret against a competitor sequence $u_1,\dots,u_T\in\Delta(N)$ can be written as

$$\mathcal{R}_T(u_1,\dots,u_T) = \sum_{t=1}^T\sum_{i=1}^N u_t(i)r_t(i)$$

where $r_t(i) = \langle p_t,\ell_t\rangle - \ell_t(i)$ is the instantaneous regret. It turns out that we can bound this regret with respect to another measure of non-stationarity:

$$A_T = \sum_{t=1}^T\sum_{i=1}^N [u_t(i)-u_{t-1}(i)]_+,$$

where $[x]_+ = \max\{x,0\}$ and u_0 is defined as an all-zero vector for convenience. In other words, A_T is the sum of "one-sided ℓ_1 norms" between consecutive competitors, and one can see it as a generalized and soft version of the number of switches (indeed when there are $S-1$ switches in the sequence, we have $A_T\le S$). Compared to V_T, which is the variation of the loss functions, A_T is the variation of the competitors. These two variations are related but in general not comparable (think about examples where $A_T=\mathcal{O}(1)$ while $V_T=\Omega(T)$, and vice versa.)

Somewhat surprisingly, all we need here is yet again a strongly adaptive algorithm:

Theorem 2 ([Luo and Schapire, 2015]). *For the expert problem, a strongly adaptive algorithm with $\mathcal{R}_\mathcal{I} = \mathcal{O}(\sqrt{|\mathcal{I}|\ln(NT)})$ for any interval $\mathcal{I}$ ensures*

$$\mathcal{R}_T(u_1,\dots,u_T) = \mathcal{O}\left(\sqrt{TA_T\ln(NT)}\right)$$

for any competitor sequence $u_1,\dots,u_T\in\Delta(N)$.

So again, whenever A_T is sublinear, the regret is sublinear. Rewriting the regret bound also leads to the following bound on the loss of the algorithm:

$$\sum_{t=1}^T\langle p_t,\ell_t\rangle \le \min_{u_1,\dots,u_T}\left(\sum_{t=1}^T\langle u_t,\ell_t\rangle + \mathcal{O}\left(\sqrt{TA_T\ln(NT)}\right)\right),$$

where one can pick $u_1,\dots,u_T$ in different ways to balance the benchmark and the regret term.

Proof. The idea is to decompose the regret into weighted sum of several interval regrets. To this end, we fix an expert i and let $\alpha_m>0$ and $\mathcal{I}_m\subset[1,T]$ $(m=1,\dots,M)$ for some M be a set of weighted intervals such that $u_t(i) = \sum_{m=1}^M \mathbf{1}\{t\in\mathcal{I}_m\}\alpha_m$ for any t. In other words, each $u_t(i)$ is decomposed as the sum of weights of the intervals that cover t. There are many different ways to do this but later we will specify what the optimal way is. For now, note that the regret (against expert

i) can be decomposed as

$$\sum_{t=1}^{T} u_t(i) r_t(i) = \sum_{t=1}^{T} \sum_{m=1}^{M} \mathbf{1}\{t \in \mathcal{I}_m\} \alpha_m r_t(i) = \sum_{m=1}^{M} \alpha_m \sum_{t=1}^{T} \mathbf{1}\{t \in \mathcal{I}_m\} r_t(i) = \sum_{m=1}^{M} \alpha_m \mathcal{R}_{\mathcal{I}_m}(i)$$

$$= \mathcal{O}\left(\sum_{m=1}^{M} \alpha_m \sqrt{|\mathcal{I}_m| \ln(NT)} \right) \leq \mathcal{O}\left(\sqrt{\sum_{m=1}^{M} \alpha_m} \sqrt{\sum_{m=1}^{M} \alpha_m |\mathcal{I}_m| \ln(NT)} \right) \quad \text{(Cauchy-Schwarz)}$$

$$= \mathcal{O}\left(\sqrt{\sum_{m=1}^{M} \alpha_m} \sqrt{\sum_{m=1}^{M} \alpha_m \sum_{t=1}^{T} \mathbf{1}\{t \in \mathcal{I}_m\} \ln(NT)} \right)$$

$$= \mathcal{O}\left(\sqrt{\sum_{m=1}^{M} \alpha_m} \sqrt{\sum_{t=1}^{T} u_t(i) \ln(NT)} \right).$$

Therefore, suppose we can pick α_m and $\mathcal{I}_m$ such that $\sum_{m=1}^{M} \alpha_m = \sum_{t=1}^{T} [u_t(i) - u_{t-1}(i)]_+$, then we prove the theorem since by applying Cauchy-Schwarz again

$$\mathcal{R}_T(u_1, \ldots, u_T) = \mathcal{O}\left(\sum_{i=1}^{N} \sqrt{\sum_{t=1}^{T} [u_t(i) - u_{t-1}(i)]_+} \sqrt{\sum_{t=1}^{T} u_t(i) \ln(NT)} \right)$$

$$\leq \mathcal{O}\left(\sqrt{T A_T \ln(NT)} \right).$$

In the remainder of the proof, we show a recursive construction of α_m and $\mathcal{I}_m$ so that $\sum_{m=1}^{M} \alpha_m = \sum_{t=1}^{T} [u_t(i) - u_{t-1}(i)]_+$. For notation convenience we drop the index i. First we let $t^\star \in \operatorname{argmin}_t u_t$ and create an interval $\mathcal{I}_1 = [1, T]$ with weight $\alpha_1 = u_{t^\star}$. Then we recursively perform the same construction for the inputs $u_1 - u_{t^\star}, \ldots, u_{t^\star - 1} - u_{t^\star}$ and $u_{t^\star + 1} - u_{t^\star}, \ldots, u_T - u_{t^\star}$ respectively until there are no non-zero inputs left. Let $h(u_1, \ldots, u_T)$ denote the sum of the weights of the above construction. We use an induction (on the length of the input T) to prove $h(u_1, \ldots, u_T) = \sum_{t=1}^{T} [u_t - u_{t-1}]_+$. The base case $T = 1$ holds trivially. Suppose the statement holds for any input length smaller than T. Then we have

$$h(u_1, \ldots, u_T) = u_{t^\star} + h(u_1 - u_{t^\star}, \ldots, u_{t^\star - 1} - u_{t^\star}) + h(u_{t^\star + 1} - u_{t^\star}, \ldots, u_T - u_{t^\star})$$

$$= u_{t^\star} + (u_1 - u_{t^\star}) + \sum_{t=2}^{t^\star - 1} [u_t - u_{t-1}]_+ + (u_{t^\star + 1} - u_{t^\star}) + \sum_{t=t^\star + 2}^{T} [u_t - u_{t-1}]_+$$

$$= u_1 + \sum_{t=2}^{t^\star - 1} [u_t - u_{t-1}]_+ + [u_{t^\star + 1} - u_{t^\star}]_+ + \sum_{t=t^\star + 2}^{T} [u_t - u_{t-1}]_+$$

$$= \sum_{t=1}^{T} [u_t - u_{t-1}]_+.$$

where the last step is by $[u_{t^\star} - u_{t^\star - 1}]_+ = 0$. This finishes the proof. In fact, the above construction is *optimal* in minimizing $\sum_{m=1}^{M} \alpha_m$ and the proof can be found in [Luo and Schapire, 2015]. □

References

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations Research*, 63(5):1227–1244, 2015.

Haipeng Luo and Robert E. Schapire. Achieving All with No Parameters: AdaNormalHedge. In *28th Annual Conference on Learning Theory*, 2015.

Lijun Zhang, Tianbao Yang, Rong Jin, and Zhi-Hua Zhou. Strongly adaptive regret implies optimally dynamic regret. *arXiv preprint arXiv:1701.07570*, 2017.

Lecture 11

Instructor: Haipeng Luo

1 The Fixed-share Algorithm

In this lecture we come back to the expert problem and introduce a specific algorithm for non-stationary environments. We start with considering the switching regret against a sequence of experts

$$\mathcal{R}_T(i_1, \ldots, i_T) = \sum_{t=1}^{T} \langle p_t, \ell_t \rangle - \ell_t(i_t)$$

where $i_1, \ldots, i_T \in [N]$ is such that $\sum_{t=2}^{T} \mathbf{1}\{i_t \neq i_{t-1}\} = S - 1$. According to discussions from previous lectures, we know that a strongly adaptive algorithm can achieve regret of order $\mathcal{O}(\sqrt{TS\ln(NT)})$, and a strongly adaptive algorithm can be constructed through a sleeping expert algorithm.

Now we discuss a very different way to derive a simple algorithm with similar regret bounds. The idea is to cast the problem as another expert problem with a set of more complicated experts. In this new expert problem, each expert (called meta-expert) can be represented by $e \in [N]^T$, a sequence of the original experts. The set of all meta-experts is $\mathcal{M} = \{e \in [N]^T : \sum_{t=2}^{T} \mathbf{1}\{e(t) \neq e(t-1)\} = S - 1\}$, that is, sequences with $S - 1$ switches. The cardinality of this set $M = |\mathcal{M}|$ is bounded by $\binom{T-1}{S-1} N^S$. Finally the loss of meta-expert e at time t is simply $\widehat{\ell}_t(e) \stackrel{\text{def}}{=} \ell_t(e(t))$, the loss of the original expert $e(t)$ at time t.

Suppose we apply an expert algorithm with regular regret guarantee to this new expert problem, and let $\widehat{p}_t(e)$ be the weight on meta-expert e at time t. Note that for any $e^\star \in \mathcal{M}$, the regret against this meta-expert $e^\star$ is

$$\begin{aligned}
\sum_{t=1}^{T} \left\langle \widehat{p}_t, \widehat{\ell}_t \right\rangle - \widehat{\ell}_t(e^\star) &= \sum_{t=1}^{T} \sum_{e \in \mathcal{M}} \widehat{p}_t(e) \ell_t(e(t)) - \ell_t(e^\star(t)) \\
&= \sum_{t=1}^{T} \sum_{i=1}^{N} \left(\sum_{e \in \mathcal{M}, e(t)=i} \widehat{p}_t(e) \right) \ell_t(i) - \ell_t(e^\star(t)),
\end{aligned}$$

which implies that if we let $p_t(i) = \sum_{e \in \mathcal{M}, e(t)=i} \widehat{p}_t(e)$ and $e^\star$ be such that $e^\star(t) = i_t$, then the regular regret in the new expert problem is exactly the switching regret $\mathcal{R}_T(i_1, \ldots, i_T)$ in the original problem. Moreover, the former should be of order $\sqrt{T \ln M} = \sqrt{T \ln \left(\binom{T-1}{S-1} N^S \right)} = \sqrt{TS \ln \left(\frac{NT}{S} \right)}$, which is the same bound we have shown for switching regret (in fact even slightly better).

The methodology above is extremely useful in quickly establishing a regret upper bound for a complicated problem and to get a sense of what is information-theoretically possible, and often time such bound also turns out to be optimal. However, the resulting algorithm is often inefficient in terms of running time because the new expert problem has a huge set of experts. This is indeed the case in the example above where M is exponential in S. Since typical expert algorithms such as Hedge have linear (in M) running time, this reduction does not lead to an efficient algorithm.

The way to address this issue is at first glance counter-intuitive: we will actually switch to an *even larger* set of experts $\mathcal{M} = \{e \in [N]^T\}$ (that is, all the possible sequences of length T). Clearly, the cardinality $M = |\mathcal{M}|$ becomes N^T, which is exponential in T. Even ignoring efficiency issues, this seems like a terrible idea since now the regular regret is $\mathcal{O}(\sqrt{T \ln M}) = \mathcal{O}(T\sqrt{\ln N})$, which is linear in T. This is actually consistent with our lower bound for dynamic regret: without any assumptions on the competitor sequence, one should not be able to obtain sublinear regret.

However, it turns out that we can address both issues (inefficiency and linear regret) at the same time by using Hedge *with a proper prior distribution*. Recall that the Hedge prediction (using current notation) is $\widehat{p}_{t+1}(e) \propto \exp(-\eta \sum_{\tau=1}^{t} \widehat{\ell}_\tau(e))$. Although we did not discuss this, similar to Squint it is straightforward to allow a prior distribution $\widehat{p}_1$ for Hedge and rewrite the update rule as

$$\widehat{p}_{t+1}(e) \propto \widehat{p}_1(e) \exp\left(-\eta \sum_{\tau=1}^{t} \widehat{\ell}_\tau(e)\right),$$

and the regret bound against any distribution $q \in \Delta(M)$ is

$$\sum_{t=1}^{T} \left\langle \widehat{p}_t - q, \widehat{\ell}_t \right\rangle \leq \frac{\mathrm{KL}(q, \widehat{p}_1)}{\eta} + T\eta \tag{1}$$

(Try to verify why this is true. Hint: the mirror descent framework is the easiest way to show this). The key is now to pick $\widehat{p}_1$ so that the prior for meta-experts with a small number of switches is large, which then implies small $\mathrm{KL}(q, \widehat{p}_1)$ when q concentrates on such meta-experts. This motivates the following prior that is defined through a Markov process:

$$\widehat{p}_1(e) = \pi(e(1)) \prod_{t=2}^{T} \pi(e(t)|e(t-1))$$

where $\pi(i) = 1/N$ for $i \in [N]$ is the initial distribution and

$$\pi(i|j) = \begin{cases} (1-\beta) & \text{if } i = j \\ \beta/(N-1) & \text{else} \end{cases}$$

for $i, j \in [N]$ and some parameter $\beta \in [0, 1]$ is the transition probability from j to i. In other words, one can imagine that each meta-expert e is created by first drawing an initial expert $e(1) \sim \pi(\cdot)$ (that is, uniformly), and then transiting to the next expert one by one with $e(t) \sim \pi(\ \cdot\ |e(t-1))$ (that is, with probability $1-\beta$ stay at the same expert, otherwise transit to one of other $N-1$ experts uniformly at random). Note that this is merely a thought experiment to define the prior $\widehat{p}_1$ – the algorithm is not actually doing this kind of sampling, nor is the environment.

It is then clear that the smaller the parameter β, the larger the probability of creating a meta-expert with a small number of switches. Specifically, the prior for e with $\sum_{t=2}^{T} \mathbf{1}\{e(t) \neq e(t-1)\} = S-1$ is at exactly $\frac{1}{N}(1-\beta)^{T-S}(\frac{\beta}{N-1})^{S-1}$. Therefore, if q concentrates on e then

$$\mathrm{KL}(q, \widehat{p}_1) = -\ln \widehat{p}_1(e) \leq S \ln N + (T-S) \ln\left(\frac{1}{1-\beta}\right) + S \ln\left(\frac{1}{\beta}\right)$$

where the last two terms is minimized when $\beta = S/T$ with minimum value $T \cdot H(S/T)$ for the binary entropy function $H(\rho) = (1-\rho)\ln\frac{1}{1-\rho} + \rho \ln\frac{1}{\rho}$. One can also verify[1] the fact $H(\rho) \leq \rho(1 + \ln\frac{1}{\rho})$ and therefore $\mathrm{KL}(q, \widehat{p}_1) = \mathcal{O}\left(S \ln\left(\frac{NT}{S}\right)\right)$ Plugging this into Eq. (1) we again obtain switching regret $\mathcal{O}\left(\sqrt{TS \ln\left(\frac{NT}{S}\right)}\right)$.

The discussion above shows that the specific prior addresses the linear regret issue. Next we show how it also allows efficient implementation. Indeed, keep in mind that ultimately we only care about getting $p_t(i) = \sum_{e:e(t)=i} \widehat{p}_t(e)$ but only the individual $\widehat{p}_t(e)$. To compute $p_t(i)$, first note that

$$p_{t+1}(i) = \sum_{e:e(t+1)=i} \widehat{p}_{t+1}(e) \propto \sum_{e:e(t+1)=i} \widehat{p}_t(e) \exp(-\eta \widehat{\ell}_t(e)) = \sum_{j=1}^{N} \left(\sum_{\substack{e:e(t)=j \\ e(t+1)=i}} \widehat{p}_t(e) \right) \exp(-\eta \ell_t(j)).$$

[1] Indeed this is true because $\ln\frac{1}{1-\rho} = \ln\left(1 + \frac{\rho}{1-\rho}\right) \leq \frac{\rho}{1-\rho}$.

Next, we claim that

$$\sum_{\substack{e:e(t)=j\\e(t+1)=i}} \widehat{p}_t(e) = \begin{cases} (1-\beta)p_t(j) & \text{if } j=i, \\ \frac{\beta}{N-1}p_t(j) & \text{else.} \end{cases} \tag{2}$$

To see this, notice that

$$\sum_{i=1}^{N} \sum_{\substack{e:e(t)=j\\e(t+1)=i}} \widehat{p}_t(e) = \sum_{e:e(t)=j} \widehat{p}_t(e) = p_t(j)$$

and also for any $i \neq j$,

$$\frac{\sum\limits_{\substack{e:e(t)=j\\e(t+1)=j}} \widehat{p}_t(e)}{\sum\limits_{\substack{e:e(t)=j\\e(t+1)=i}} \widehat{p}_t(e)} = \frac{\sum\limits_{\substack{e:e(t)=j\\e(t+1)=j}} \pi(e(1)) \left(\prod\limits_{\tau=2}^{t} \pi(e(\tau)|e(\tau-1))\right) \pi(j|j) \exp(-\eta \sum\limits_{\tau<t} \ell_\tau(e(\tau)))}{\sum\limits_{\substack{e:e(t)=j\\e(t+1)=i}} \pi(e(1)) \left(\prod\limits_{\tau=2}^{t} \pi(e(\tau)|e(\tau-1))\right) \pi(i|j) \exp(-\eta \sum\limits_{\tau<t} \ell_\tau(e(\tau)))} = \frac{1-\beta}{\frac{\beta}{N-1}}$$

which together implies Eq. (2). We therefore continue with

$$\begin{aligned} p_{t+1}(i) &\propto \sum_{j=1}^{N} \left(\sum_{\substack{e:e(t)=j\\e(t+1)=i}} \widehat{p}_t(e) \right) \exp(-\eta \ell_t(j)) \\ &= (1-\beta) p_t(i) \exp(-\eta \ell_t(i)) + \frac{\beta}{N-1} \sum_{j \neq i} p_t(j) \exp(-\eta \ell_t(j)) \\ &= (1-\alpha) p_t(i) \exp(-\eta \ell_t(i)) + \frac{\alpha}{N} \sum_{j=1}^{N} p_t(j) \exp(-\eta \ell_t(j)) \qquad (\text{define } \alpha = \tfrac{N\beta}{N-1}) \end{aligned}$$

which implies

$$p_{t+1}(i) = (1-\alpha) \frac{p_t(i) \exp(-\eta \ell_t(i))}{\sum_{j=1}^{N} p_t(j) \exp(-\eta \ell_t(j))} + \frac{\alpha}{N} \tag{3}$$

With p_1 being the uniform distribution, this provides an efficient and recursive formula for computing p_t! Note that the multiplicative update form of Hedge plays a key role in this derivation. Update rule Eq. (3) is called the fixed-share algorithm [Herbster and Warmuth, 1998]. Despite the somewhat complicated derivation, the final algorithm is in fact extremely simple. When $\alpha = 0$, fixed-share simply recovers Hedge. When $\alpha \neq 0$, fixed-share is mixing some amount of uniform exploration into Hedge, but in a recursive way. Indeed, fixed-share is different from the following update rule which mixes some uniform exploration into Hedge directly

$$p_t(i) \propto (1-\alpha) \frac{\exp(-\eta \sum_{\tau=1}^{t-1} \ell_\tau(i))}{\sum_{j=1}^{N} \exp(-\eta \sum_{\tau=1}^{t-1} \ell_\tau(j))} + \frac{\alpha}{N}.$$

The difference is in fact crucial – in the above update rule, the losses at different time are still treated equally, while in fixed-share, by expanding the recursive formula one can see that roughly speaking, the most recent loss is weighted by $(1-\alpha)$, the second most recent loss is weighted by $(1-\alpha)^2$, so on and so forth. This is very important for getting switching regret or in general for non-stationary environments since recent data is intuitively more useful than data obtained a long time ago.

As a final remark, one can also prove interval regret and even dynamic regret for fixed-share (details omitted). In all cases, parameter tunning (η and α) is a problem, but there are also approaches to fix it.

References

Mark Herbster and Manfred K Warmuth. Tracking the best expert. *Machine learning*, 32(2):151–178, 1998.

Lecture 12

Instructor: Haipeng Luo

1 The Multi-armed Bandit Problem

All the topics we have discussed so far consider problem with full information feedback. Staring from this lecture, we will move on to the more challenging settings with partial information feedback. The classic example of such problems is the *multi-armed bandit* problem [Lai and Robbins, 1985], and here we discuss an adversarial version introduced in [Auer et al., 2002].

The problem models the situation where a gambler sequentially pull the arm of one of the slot machines in a casino, with the hope of maximizing reward. A slot machine is sometimes called a "one-armed bandit", and hence the name multi-armed bandit for this problem. Formally, there are K arms/actions available for a learner, and at each time $t = 1, \ldots, T$,

1. the learner picks an action $a_t \in [K]$ while simultaneously the environment decides the loss vector $\ell_t \in [0,1]^K$,
2. the learner then suffers and observes (only) the loss $\ell_t(a_t)$.

Clearly, this is simply a partial information version of the expert problem, with the difference being that the learner has to actually pick one action at each round and then observe only the loss for this action but not the whole loss vector ℓ_t. For convention, we move from the notation i and N to a and K to denote a specific action and the total number of actions respectively.

For simplicity we only consider oblivious environment and thus one can equivalently think of the loss vectors as generated ahead of time before the game starts (possibly randomly though). We measure the algorithm's performance by the expected regret

$$\mathbb{E}\left[\mathcal{R}_T\right] = \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(a_t)\right] - \min_{a\in[K]} \sum_{t=1}^{T} \ell_t(a),$$

where the expectation is with respect to the randomness of the algorithm.

The challenge of this problem (or in general all partial information problem) is the well-known exploitation-exploration tradeoff. Indeed, on one hand, it's tempting to pick actions that have suffered small losses before (exploitation), but on the other hand, there is also an incentive to pick other actions just to see whether they can admit even smaller losses (exploration).

But since the problem is so close to the expert problem, let's first see whether we can somehow use an expert algorithm to solve it. The obvious obstacle is that we do not have the whole loss vector to feed to an expert algorithm. However, suppose we pick a_t according to a distribution p_t, then we can construct an estimator for the loss vector in the following way

$$\widehat{\ell}_t(a) = \frac{\ell_t(a)}{p_t(a)}\mathbf{1}\{a = a_t\} = \begin{cases} \frac{\ell_t(a_t)}{p_t(a_t)} & \text{if } a = a_t, \\ 0 & \text{else.} \end{cases}$$

This simple trick is called inverse propensity score weighting or simply importance weighting. Apparently the estimator is computable using the available information, and more importantly, it is unbiased: for any $a \in [K]$,

$$\mathbb{E}_t[\widehat{\ell}(a)] = (1 - p_t(a)) \times 0 + p_t(a)\frac{\ell_t(a)}{p_t(a)} = \ell_t(a)$$

where $\mathbb{E}_t[\cdot]$ is the conditional expectation with respect to the random draw of a_t given the past. Therefore, since we only care about expected regret (at least for now), it seems like we can simply use the prediction of an arbitrary expert algorithm p_t to draw a_t, and then feed $\widehat{\ell}_t$ to the algorithm. Indeed, we have for any $a \in [K]$,

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(a_t)\right] - \sum_{t=1}^{T} \ell_t(a) = \mathbb{E}\left[\sum_{t=1}^{T} \left\langle p_t, \widehat{\ell}_t \right\rangle - \sum_{t=1}^{T} \widehat{\ell}_t(a)\right]$$

where the last term is exactly the (expected) regret of the expert algorithm. We have showed that the optimal regret for the expert problem is $\mathcal{O}(\sqrt{T \ln K})$. Does this mean we have come up with a simple algorithm for the multi-armed bandit with regret $\mathcal{O}(\sqrt{T \ln K})$?

The answer is no – what we missed in the above argument is the fact that the range of the losses that the expert algorithm receives is no longer in $[0, 1]$! In fact, it could potentially be very large due to the importance weighting and thus the regret is no longer just $\mathcal{O}(\sqrt{T \ln K})$. As a simple fix, we can try to enforce a lower bound on the importance weight by doing a small amount of uniform exploration

$$p_t = (1 - \alpha)\widehat{p}_t + \frac{\alpha}{K}\mathbf{1} \tag{1}$$

where $\widehat{p}_t$ is now the prediction of the expert algorithm, $\mathbf{1}$ is the all-one vector, and α is some parameter to be specified later. Then clearly we have $\widehat{\ell}(a) \leq K/\alpha$ and thus if we feed the expert algorithm with $\frac{\alpha}{K}\widehat{\ell}_t \in [0, 1]^K$, we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \left\langle p_t, \widehat{\ell}_t \right\rangle - \sum_{t=1}^{T} \widehat{\ell}_t(a)\right] &\leq \frac{\alpha}{K}\mathbb{E}\left[\sum_{t=1}^{T} \widehat{\ell}_t(a_t)\right] + \mathbb{E}\left[\sum_{t=1}^{T} \left\langle \widehat{p}_t, \widehat{\ell}_t \right\rangle - \sum_{t=1}^{T} \widehat{\ell}_t(a)\right] \\
&\leq \alpha T + \frac{K}{\alpha}\mathbb{E}\left[\sum_{t=1}^{T} \left\langle \widehat{p}_t, \frac{\alpha}{K}\widehat{\ell}_t \right\rangle - \sum_{t=1}^{T} \frac{\alpha}{K}\widehat{\ell}_t(a)\right] \\
&= \mathcal{O}\left(\alpha T + \frac{K}{\alpha}\sqrt{T \ln K}\right)
\end{aligned}$$

where the second step is by $\mathbb{E}_t[\widehat{\ell}_t(a_t)] = \sum_{a=1}^{K} p_t(a)\frac{\ell_t(a)}{p_t(a)} = \sum_{a=1}^{K} \ell_t(a) \leq K$ and the last step is by applying the regret bound of the expert algorithm. Finally, by picking the optimal α we achieve a regret bound of order $\mathcal{O}(T^{\frac{3}{4}} K^{\frac{1}{2}} (\ln K)^{\frac{1}{4}})$, which is much larger than the optimal bound for the full information setting. Therefore, although the importance weighted estimator is unbiased and we do only care about regret in expectation, the range or really the variance of the estimator still matters.

2 The Exp3 Algorithm

Can we do better than the approach discussed above? It turns out that the answer is yes, and the solution is simply by using Hedge as the expert algorithm in the above reduction, *without even mixing the uniform distribution*. To see this, note that the potential-based proof of Hedge does not use the fact that the losses are in $[0, 1]$ at all to arrive at the following

$$\sum_{t=1}^{T} \left\langle p_t, \widehat{\ell}_t \right\rangle - \sum_{t=1}^{T} \widehat{\ell}_t(a) \leq \frac{\ln K}{\eta} + \eta \sum_{t=1}^{T} \sum_{a=1}^{K} p_t(a)\widehat{\ell}_t(a)^2. \tag{2}$$

Noting that the variance (or rather the second moment) of the estimator is $\mathbb{E}_t[\widehat{\ell}_t(a)^2] = \frac{\ell_t(a)^2}{p_t(a)}$, we continue with

$$\mathbb{E}\left[\sum_{t=1}^{T} \left\langle p_t, \widehat{\ell}_t \right\rangle - \sum_{t=1}^{T} \widehat{\ell}_t(a)\right] \leq \frac{\ln K}{\eta} + \eta\left[\sum_{t=1}^{T} \sum_{a=1}^{K} p_t(a)\frac{\ell_t(a)^2}{p_t(a)}\right] \leq \frac{\ln K}{\eta} + TK\eta,$$

which means with the optimal tuning $\eta = \sqrt{(\ln K)/(TK)}$ the regret is only $\mathcal{O}(\sqrt{TK \ln K})$, much better than the previous $\mathcal{O}(T^{\frac{3}{4}})$ bound! This is yet another example of the power of adaptive regret

Algorithm 1: Exp3

Input: learning rate $\eta > 0$

Initialization: let $\widehat{L}_0$ be the all-zero vector

for $t = 1, \dots, T$ **do**

compute $p_t \in \Delta(K)$ such that $p_t(a) \propto \exp(-\eta \widehat{L}_{t-1}(a))$

play $a_t \sim p_t$ and observe its loss $\ell_t(a_t)$

update $\widehat{L}_t = \widehat{L}_{t-1} + \widehat{\ell}_t$ where $\widehat{\ell}_t(a) = \frac{\ell_t(a)}{p_t(a)} \mathbf{1}\{a = a_t\},\ \forall a \in [K]$

bounds, and it is in fact pretty magical that bound (2) can automatically deal with the large variance issue of the estimators.

This algorithm (summarized in Algorithm 1) is called Exp3 (which stands for Exponential-weight for Exploration and Exploitation) is the first and arguably most important algorithm for adversarial multi-armed bandit. Its regret bound is summarized in the following theorem for completeness.

Theorem 1. *With the optimal tuning Exp3 ensures* $\mathbb{E}[\mathcal{R}_T] = \mathcal{O}(\sqrt{TK \ln K})$.

It is worth noting that although there is no explicit exploration (like Eq. (1)) in Exp3, the algorithm is in fact doing some implicit exploration. Indeed, whenever an arm a_t is pulled (maybe due to exploitation), its weight for the next round is always not increased no matter what the loss vector ℓ_t is, which will then encourage the algorithm to explore other actions next round. This is due to the structure of the estimator $\widehat{\ell}_t$ so that only the picked action a_t could have non-zero loss, while all the other actions have estimated loss 0.

3 Lower Bounds

The regret bound of Exp3 is showing that the price of bandit feedback is only a $\sqrt{K}$ factor compared to full information feedback (ignoring logarithmic terms). Is this optimal?

Intuitively it should be. Consider the following very informal argument. Suppose the losses are all generated independently and uniformly from $\{0, 1\}$. For any fixed algorithm, there must be an arm that is pulled no more than T/K times by this algorithm. Now suppose the environment is modified so that the loss of this arm follows a Bernoulli distribution with parameter $1/2 - \sqrt{K/T}$, which is not distinguishable from the uniform distribution information-theoretically with only T/K samples. Then the algorithm should not be aware of this change and still pull this arm no more than T/K rounds, leading to an expected regret $(T - T/K)\sqrt{K/T} \approx \sqrt{TK}$.

The following theorem makes the argument above formal with a probabilistic argument, similar to the lower bound proof for the expert problem.

Theorem 2. *For any multi-armed bandit algorithm $\mathcal{A}$, there exists a sequence of loss vectors s.t.*

$$\mathbb{E}_{\mathcal{A}}[\mathcal{R}_T] = \Omega(\sqrt{TK})$$

where we use $\mathbb{E}_{\mathcal{A}}[\cdot]$ to denote the expectation with respect to the randomness of $\mathcal{A}$.

Proof. Consider randomly generating an environment in the following way: first draw an action uniformly at random to be the "good" action; then for all $t \in [T]$ and $a \in [K]$ independently generate $\ell_t(a)$ whose distribution is a Bernoulli with parameter $1/2 - \epsilon$ if a is the good action (ϵ to be specified later), or uniform on $\{0, 1\}$ otherwise. Let $\mathbb{E}_\star[\cdot]$ be the expectation with respect to the random draw of such environment. Our goal is to prove

$$\mathbb{E}_\star[\mathbb{E}_{\mathcal{A}}[\mathcal{R}_T]] = \Omega(\sqrt{TK}) \tag{3}$$

which clearly implies the theorem.

The first step to prove Eq. (3) is to realize that $\mathbb{E}_\star[\mathbb{E}_{\mathcal{A}}[\mathcal{R}_T]] = \mathbb{E}_{\mathcal{A}}[\mathbb{E}_\star[\mathcal{R}_T]]$ and thus it is enough to show that for any deterministic algorithm, $\mathbb{E}_\star[\mathcal{R}_T]$ has the same lower bound. Note that for a deterministic algorithm, a_t is completely determined by $\widetilde{\ell}_{1:t-1}$, a shorthand for $\ell_1(a_1), \dots, \ell_{t-1}(a_{t-1})$.

Now let $\mathbb{E}_a[\cdot]$ denote the conditional expectation given that the good action is a, we have

$$\begin{aligned}
\mathbb{E}_\star[\mathcal{R}_T] &= \frac{1}{K}\sum_{a=1}^{K}\mathbb{E}_a\left[\sum_{t=1}^{T}\ell_t(a_t) - \min_{a^\star\in[K]}\sum_{t=1}^{T}\ell_t(a^\star)\right] \\
&\geq \frac{1}{K}\sum_{a=1}^{K}\mathbb{E}_a\left[\sum_{t=1}^{T}\ell_t(a_t) - \sum_{t=1}^{T}\ell_t(a)\right] = \frac{1}{K}\sum_{a=1}^{K}\mathbb{E}_a\left[\sum_{t:a_t\neq a}(\ell_t(a_t)-\ell_t(a))\right] \\
&\geq \frac{1}{K}\sum_{a=1}^{K}\mathbb{E}_a\left[\sum_{t:a_t\neq a}\epsilon\right] = \epsilon\left(T - \frac{1}{K}\sum_{a=1}^{K}\mathbb{E}_a\left[n_a\right]\right) \qquad (4)
\end{aligned}$$

where n_a is the number of times that a is picked by the algorithm. To upper bound the term $\mathbb{E}_a[n_a]$, we imagine a reference environment where every loss is an independent and uniform draw from $\{0,1\}$, and let $\mathbb{E}_0[\cdot]$ denote the corresponding expectation. Noting that n_a is a function of $\widetilde{\ell}_{1:T}$, with $\mathbb{P}_0, \mathbb{P}_1, \ldots, \mathbb{P}_K$ being the probability distribution of $\widetilde{\ell}_{1:T}$ under the corresponding environment, we can relate $\mathbb{E}_a[n_a]$ and $\mathbb{E}_0[n_a]$ as

$$\mathbb{E}_a[n_a] - \mathbb{E}_0[n_a] = \sum_{\widetilde{\ell}_{1:T}} n_a\left(\mathbb{P}_a(\widetilde{\ell}_{1:T}) - \mathbb{P}_0(\widetilde{\ell}_{1:T})\right) \leq T\sum_{\widetilde{\ell}_{1:T}}\left|\mathbb{P}_a(\widetilde{\ell}_{1:T}) - \mathbb{P}_0(\widetilde{\ell}_{1:T})\right| = T\left\|\mathbb{P}_a - \mathbb{P}_0\right\|_1$$

which by Pinskers inequality is bounded by $\sqrt{2\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_a)}$. We compute the KL term as follows:

$$\begin{aligned}
\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_a) &= \sum_{\widetilde{\ell}_{1:T}}\mathbb{P}_0(\widetilde{\ell}_{1:T})\ln\left(\frac{\mathbb{P}_0(\widetilde{\ell}_{1:T})}{\mathbb{P}_a(\widetilde{\ell}_{1:T})}\right) = \sum_{\widetilde{\ell}_{1:T}}\mathbb{P}_0(\widetilde{\ell}_{1:T})\ln\left(\frac{\prod_{t=1}^{T}\mathbb{P}_0(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}{\prod_{t=1}^{T}\mathbb{P}_a(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}\right) \\
&= \sum_{t=1}^{T}\sum_{\widetilde{\ell}_{1:t}}\mathbb{P}_0(\widetilde{\ell}_{1:t})\ln\left(\frac{\mathbb{P}_0(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}{\mathbb{P}_a(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}\right) = \sum_{t=1}^{T}\sum_{\widetilde{\ell}_{1:t}:a_t=a}\mathbb{P}_0(\widetilde{\ell}_{1:t})\ln\left(\frac{\mathbb{P}_0(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}{\mathbb{P}_a(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}\right) \\
&= \sum_{t=1}^{T}\sum_{\widetilde{\ell}_{1:t-1}:a_t=a}\mathbb{P}_0(\widetilde{\ell}_{1:t-1})\sum_{\widetilde{\ell}_t\in\{0,1\}}\mathbb{P}_0(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})\ln\left(\frac{\mathbb{P}_0(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}{\mathbb{P}_a(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}\right) \\
&= \frac{1}{2}\sum_{t=1}^{T}\mathbb{P}_0(a_t=a)\left(\ln\frac{1/2}{1/2+\epsilon} + \ln\frac{1/2}{1/2-\epsilon}\right) = \frac{\mathbb{E}_0[n_a]}{2}\ln\left(\frac{1}{1-4\epsilon^2}\right).
\end{aligned}$$

Therefore, we have by $\sum_{a=1}^{K}\mathbb{E}_0[n_a] = T$ and Cauchy-Schwarz inequality

$$\sum_{a=1}^{K}\mathbb{E}_a[n_a] \leq \sum_{a=1}^{K}\mathbb{E}_0[n_a] + T\sum_{a=1}^{K}\sqrt{\mathbb{E}_0[n_a]\ln\left(\frac{1}{1-4\epsilon^2}\right)} \leq T + T\sqrt{KT\ln\left(\frac{1}{1-4\epsilon^2}\right)}.$$

Plugging the above back to Eq. (4) shows

$$\mathbb{E}_\star[\mathcal{R}_T] \geq \epsilon T\left(1 - \frac{1}{K} - \sqrt{\frac{T}{K}\ln\left(\frac{1}{1-4\epsilon^2}\right)}\right) = \Omega\left(\epsilon T\left(1 - \epsilon\sqrt{\frac{T}{K}}\right)\right),$$

which proves Eq. (3) with the optimal ϵ and finishes the proof. □

Therefore we see that Exp3 is almost worst-case optimal. In the next lecture we will discuss algorithms that are exactly optimal up to constants.

References

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

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

Lecture 13

Instructor: Haipeng Luo

1 Optimal Multi-armed Bandit Algorithms

We have shown a lower bound of order $\Omega(\sqrt{TK})$ for the expected regret of any algorithm for the multi-armed bandit problem, and also that Exp3 ensures an expected regret of order $\mathcal{O}(\sqrt{TK \ln K})$. Can we close the $\sqrt{\ln K}$ gap in the upper and lower bounds?

The answer turns out to be yes, and the approach is again FTRL, but with special regularizers. Specifically, consider the following general FTRL algorithm for multi-armed bandit [Audibert and Bubeck, 2010, Abernethy et al., 2015]: draw $a_t \sim p_t$ with

$$p_{t+1} = \operatorname*{argmin}_{p \in \Delta(K)} \left\langle p, \sum_{\tau=1}^{t} \widehat{\ell}_\tau \right\rangle + \frac{1}{\eta}\psi(p)$$

where $\psi(p)$ is a regularizer and $\widehat{\ell}_t(a) = \frac{\ell_t(a)}{p_t(a)}\mathbf{1}\{a = a_t\}$ is the importance weighted estimator. Exp3 is clearly just a special case with $\psi(p)$ being the negative entropy. To derive the optimal algorithm, we will consider a family of FTRL instances by using the following regularizer

$$\psi(p) = \frac{1 - \sum_{a=1}^{K} p(a)^\beta}{1-\beta},$$

for a parameter $\beta \in (0,1)$. This is known as the *Tsallis entropy* and is in fact a generalization of the Shannon entropy since $\lim_{\beta \to 1} \frac{1-\sum_a p(a)^\beta}{1-\beta} = \sum_a p(a)\ln(p(a))$ by L'Hôpital's rule. Therefore the algorithm above can be seen as a generalization of the Exp3 algorithm. One can now verify that the algorithm admits the following update rule

$$\frac{1}{p_{t+1}(a)^{1-\beta}} = \frac{1-\beta}{\beta}\left(\lambda + \eta \sum_{\tau=1}^{t} \widehat{\ell}_\tau(a)\right), \quad \forall a \in [K] \tag{1}$$

for some constant λ such that p_{t+1} is a distribution. This constant λ comes from the Lagrangian multiplier and can be found efficiently by a simple binary search.

While it is possible to use the general FTRL analysis to derive the regret bound for this algorithm, it is in fact simpler to analyze it using the Online Mirror Descent (OMD) framework (see Homework 1). Recall that one way to write the OMD algorithm is

$$\begin{aligned} \nabla\psi(p'_{t+1}) &= \nabla\psi(p_t) - \eta\widehat{\ell}_t \\ p_{t+1} &= \operatorname*{argmin}_{p \in \Delta(K)} D_\psi(p, p'_{t+1}) \end{aligned}$$

where $D_\psi(p,q) = \psi(p) - \psi(q) - \langle \nabla\psi(q), p-q\rangle$ is the Bregman divergence associated with ψ. With ψ being the Tsallis entropy, one can verify $\nabla\psi(q)(a) = \frac{-\beta}{1-\beta}\frac{1}{q(a)^{1-\beta}}$ and

$$D_\psi(p,q) = \frac{1}{1-\beta}\sum_{a=1}^{K}\left(q(a)^\beta - p(a)^\beta + \frac{\beta}{q(a)^{1-\beta}}(p(a)-q(a))\right)$$

and the update rule becomes

$$\frac{1}{p'_{t+1}(a)^{1-\beta}} = \frac{1}{p_t(a)^{1-\beta}} + \frac{1-\beta}{\beta}\eta\widehat{\ell}_t(a) \tag{2}$$

$$\frac{1}{p_{t+1}(a)^{1-\beta}} = \frac{1}{p'_{t+1}(a)^{1-\beta}} + \lambda \tag{3}$$

where λ is again such that p_{t+1} is a distribution (different from the λ in Eq. (1) though) and can be computed by a binary search. This update rule is in fact equivalent to the FTRL update rule (1) since combining (2) and (3) iteratively leads to

$$\frac{1}{p_{t+1}(a)^{1-\beta}} = \frac{1}{p_t(a)^{1-\beta}} + \frac{1-\beta}{\beta}\eta\widehat{\ell}_t(a) + \lambda = \cdots = \frac{1-\beta}{\beta}\eta\left(\sum_{\tau=1}^{t}\widehat{\ell}_\tau(a)\right) + \lambda'$$

for some other normalization term λ'. This shows that the two algorithms are exactly the same and we can use the OMD analysis to analyze the algorithm, which is the focus for the rest of the section.

Recall that the key in the proof of Exp3 is the following bound: $\forall a^\star \in [K]$,

$$\sum_{t=1}^{T}\left\langle p_t, \widehat{\ell}_t\right\rangle - \widehat{\ell}_t(a^\star) \le \frac{\ln K}{\eta} + \eta\sum_{t=1}^{T}\sum_{a=1}^{K} p_t(a)\widehat{\ell}_t(a)^2, \tag{4}$$

and the last term deals with the large variance issue of the estimator automatically. With ψ being the Tsallis entropy, we can prove a generalized version of the bound:

Theorem 1. *As long as $\widehat{\ell}_t(a) \ge 0$ for all t and a, FTRL* (1) *or OMD* (2) (3) *ensures $\forall a^\star \in [K]$,*

$$\sum_{t=1}^{T}\left\langle p_t, \widehat{\ell}_t\right\rangle - \widehat{\ell}_t(a^\star) \le \frac{K^{1-\beta}-1}{\eta(1-\beta)} + \frac{\eta}{\beta}\sum_{t=1}^{T}\sum_{a=1}^{K} p_t(a)^{2-\beta}\widehat{\ell}_t(a)^2. \tag{5}$$

Note that the theorem does not require $\widehat{\ell}_t(a)$ to be the specific importance weighted estimator. By L'Hôpital's rule, we have $\lim_{\beta\to 1}\frac{K^{1-\beta}-1}{(1-\beta)} = \ln K$ and thus the bound above exactly recovers Eq. (4). However, the bound is actually slightly better and allows one to obtain the optimal regret as shown by the following corollary.

Corollary 1. *With $\widehat{\ell}_t$ being the importance weighted estimator, FTRL* (1) *or OMD* (2) (3) *ensures*

$$\mathbb{E}[\mathcal{R}_T] \le \frac{K^{1-\beta}-1}{\eta(1-\beta)} + \frac{\eta K^\beta T}{\beta}.$$

Therefore, by picking $\beta = 1/2$ and $\eta = 1/\sqrt{T}$, we obtain the optimal regret $\mathbb{E}[\mathcal{R}_T] = 4\sqrt{TK}$.

Proof. Recall that the conditional second moment of the estimator $\mathbb{E}_t[\widehat{\ell}_t(a)^2]$ is bounded by $1/p_t(a)$. Therefore, by taking expectation on both sides of Eq. (5), we arrive at

$$\mathbb{E}[\mathcal{R}_T] \le \frac{K^{1-\beta}-1}{\eta(1-\beta)} + \frac{\eta}{\beta}\sum_{t=1}^{T}\sum_{a=1}^{K} p_t(a)^{1-\beta}.$$

Applying Hölder's inequality to the last term

$$\sum_{a=1}^{K} p_t(a)^{1-\beta} \le \left(\sum_{a=1}^{K}(p_t(a)^{1-\beta})^{\frac{1}{1-\beta}}\right)^{1-\beta}\left(\sum_{a=1}^{K}1^{\frac{1}{\beta}}\right)^{\beta} \le K^\beta$$

finishes the proof. □

Clearly picking other constants such $\beta = 1/3$ (along with the optimal η) can also lead to a bound of the optimal order $\mathcal{O}(\sqrt{TK})$. It remains to prove Theorem 1.

Proof of Theorem 1. According to the OMD analysis, for any $q \in \Delta(K)$ we have

$$\begin{aligned}\eta \left\langle p_t - q, \widehat{\ell}_t \right\rangle &= D_\psi(q, p_t) - D_\psi(q, p'_{t+1}) + D_\psi(p_t, p'_{t+1}) \\ &\le D_\psi(q, p_t) - D_\psi(q, p_{t+1}) + D_\psi(p_t, p'_{t+1}),\end{aligned}$$

and thus

$$\sum_{t=1}^{T} \left\langle p_t - q, \widehat{\ell}_t \right\rangle \le \frac{D_\psi(q, p_1)}{\eta} + \frac{1}{\eta} \sum_{t=1}^{T} D_\psi(p_t, p'_{t+1}).$$

When q concentrates on one particular action, $D_\psi(q, p_1) = \frac{K^{1-\beta}-1}{(1-\beta)}$. Therefore, it remains to prove

$$D_\psi(p_t, p'_{t+1}) \le \frac{\eta^2}{\beta} \sum_{a=1}^{K} p_t(a)^{2-\beta} \widehat{\ell}_t(a)^2. \tag{6}$$

Indeed, by definition we have

$$\begin{aligned}D_\psi(p_t, p'_{t+1}) &= \frac{1}{1-\beta} \sum_{a=1}^{K} \left(p'_{t+1}(a)^\beta - p_t(a)^\beta + \frac{\beta}{p'_{t+1}(a)^{1-\beta}} (p_t(a) - p'_{t+1}(a)) \right) \\ &= \frac{1}{1-\beta} \sum_{a=1}^{K} \left((1-\beta) p'_{t+1}(a)^\beta - p_t(a)^\beta + \beta \left(\frac{1}{p_t(a)^{1-\beta}} + \frac{1-\beta}{\beta} \eta \widehat{\ell}_t(a) \right) p_t(a) \right) \\ &= \sum_{a=1}^{K} \left(p'_{t+1}(a)^\beta - p_t(a)^\beta + \eta p_t(a) \widehat{\ell}_t(a) \right). \end{aligned} \tag{7}$$

Now notice that

$$p'_{t+1}(a)^\beta = p_t(a)^\beta \left(\frac{p'_{t+1}(a)^{\beta-1}}{p_t(a)^{\beta-1}} \right)^{\frac{\beta}{\beta-1}} = p_t(a)^\beta \left(1 + \frac{1-\beta}{\beta} \eta p_t(a)^{1-\beta} \widehat{\ell}_t(a) \right)^{\frac{\beta}{\beta-1}},$$

and thus using the fact $(1+x)^\alpha \le 1 + \alpha x + \alpha(\alpha-1)x^2$ for any $x \ge 0$ and $\alpha < 0$,[1] we have

$$\begin{aligned}p'_{t+1}(a)^\beta &\le p_t(a)^\beta \left(1 - \eta p_t(a)^{1-\beta} \widehat{\ell}_t(a) + \frac{\eta^2}{\beta} p_t(a)^{2-2\beta} \widehat{\ell}_t(a)^2 \right) \\ &= p_t(a)^\beta - \eta p_t(a) \widehat{\ell}_t(a) + \frac{\eta^2}{\beta} p_t(a)^{2-\beta} \widehat{\ell}_t(a)^2.\end{aligned}$$

Plugging this into Eq. (7) proves Eq. (6) and thus the theorem. □

2 High Probability Bounds

So far we have only proven that the expected regret of Exp3 or the more general algorithm is nicely bounded. However, since online learning focuses more on sequentially playing the game without going back, it seems that the *expected* regret does not really say much about the performance of the algorithm for a particular run. To address this issue, we need to derive a bound on the actual regret that holds with high probability.

Due to the high variance of the importance weighted estimator, without any modification the approaches we have discussed cannot ensure the same regret bound with high probability. There are many fixes for this, but they all share the same idea of sacrificing a little bit of unbiasedness to lower the variance. Here, we discuss a simple strategy introduced in [Neu, 2015], which constructs loss estimators as

$$\widehat{\ell}_t(a) = \frac{\ell_t(a)}{p_t(a) + \gamma} \mathbf{1}\{a = a_t\}, \quad \forall a \in [K] \tag{8}$$

[1] This is because with $y = \ln(1+x)$, one has $(1+x)^\alpha = e^{\alpha y} \le 1 + \alpha y + \alpha^2 y^2$ due to $\alpha y < 0$. Further using inequalities $y = \ln(1+x) \ge x - x^2$ and $y = \ln(1+x) \le x$ proves the fact.

for some parameter $\gamma > 0$. The rest of the algorithm remains exactly the same: plug this new estimator into the update rule of Exp3 or FTRL/OMD with Tsallis entropy to obtain p_t and then sample $a_t \sim p_t$.

The new estimator makes a difference mostly when $p_t(a_t)$ is small – in this case the extra term γ makes the estimator much less dramatic. In general the estimator is underestimating the losses and the following important property holds (the proof can be found in [Neu, 2015] and is omitted here).

Lemma 1. *Let $c_1, \ldots, c_T \in [0, 2\gamma]^K$ be such that $c_t(a)$ is fixed given everything up to the beginning of time t. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$, we have*

$$\sum_{t=1}^{T}\sum_{a=1}^{K} c_t(a)\left(\widehat{\ell}_t(a) - \ell_t(a)\right) \le \ln\left(1/\delta\right).$$

Note that before when $\widehat{\ell}_t$ was unbiased, such inequality would not hold because the (large) variance plays a role in the martingale concentration bound. The key is that the new estimator is now an underestimation, making such one-sided inequality possible. Moreover, such one-sided inequality turns out to be all we need to prove a high probability bound.

Theorem 2. *With $\widehat{\ell}_t$ defined as in Eq.* (8)*, FTRL* (1) *or OMD* (2) (3) *ensures that for a fixed $a^\star \in [K]$, we have with probability at least $1-\delta$,*

$$\sum_{t=1}^{T} \ell_t(a) - \ell_t(a^\star) \le \frac{K^{1-\beta}-1}{\eta(1-\beta)} + \frac{\eta K^\beta T}{\beta} + \gamma T K + \frac{1}{2}\left(\frac{\eta}{\beta\gamma} + \frac{1}{\gamma} + 1\right)\ln\left(\frac{3}{\delta}\right).$$

Picking $\beta = 1/2$, $\eta = 1/\sqrt{T}$ and $\gamma = \sqrt{\frac{\ln(1/\delta)}{TK}}$ leads to $\mathcal{R}_T = \mathcal{O}(\sqrt{TK\ln(1/\delta)} + \ln(1/\delta))$.

Proof. Note that

$$\left\langle p_t, \widehat{\ell}_t \right\rangle = p_t(a_t)\frac{\ell_t(a_t)}{p_t(a_t)+\gamma} = \ell_t(a_t) - \gamma\frac{\ell_t(a_t)}{p_t(a_t)+\gamma} = \ell_t(a_t) - \gamma\sum_{a=1}^{K}\widehat{\ell}_t(a).$$

Therefore, by applying Theorem 1 which holds here since $\widehat{\ell}_t(a) \ge 0$, we have

$$\begin{aligned}
\sum_{t=1}^{T}\ell_t(a_t) &\le \sum_{t=1}^{T}\left(\left\langle p_t, \widehat{\ell}_t\right\rangle + \gamma\sum_{a=1}^{K}\widehat{\ell}_t(a)\right) \\
&\le \frac{K^{1-\beta}-1}{\eta(1-\beta)} + \sum_{t=1}^{T}\left(\widehat{\ell}_t(a^\star) + \frac{\eta}{\beta}\sum_{a=1}^{K} p_t(a)^{2-\beta}\widehat{\ell}_t(a)^2 + \gamma\sum_{a=1}^{K}\widehat{\ell}_t(a)\right) \\
&\le \frac{K^{1-\beta}-1}{\eta(1-\beta)} + \sum_{t=1}^{T}\left(\widehat{\ell}_t(a^\star) + \frac{\eta}{\beta}\sum_{a=1}^{K} p_t(a)^{1-\beta}\widehat{\ell}_t(a) + \gamma\sum_{a=1}^{K}\widehat{\ell}_t(a)\right)
\end{aligned}$$

where the last step is due to $p_t(a)\widehat{\ell}_t(a) \le 1$. We can now apply Lemma 1 to the last three terms with $c_t(a) \le 2\gamma$ being $2\gamma\mathbf{1}\{a = a^\star\}$, $2\gamma p_t(a)^{1-\beta}$, and 2γ respectively and a union bound to conclude that with probability at least $1-\delta$, the last three terms are bounded by

$$\sum_{t=1}^{T}\left(\ell_t(a^\star) + \frac{\eta}{\beta}\sum_{a=1}^{K} p_t(a)^{1-\beta}\ell_t(a) + \gamma\sum_{a=1}^{K}\ell_t(a)\right) + \frac{1}{2}\left(\frac{\eta}{\beta\gamma} + \frac{1}{\gamma} + 1\right)\ln\left(\frac{3}{\delta}\right).$$

Rearranging, using $\ell_t(a) \le 1$, and applying Hölder's inequality as in the proof of Corollary 1 finish the proof. □

References

Jacob D Abernethy, Chansoo Lee, and Ambuj Tewari. Fighting bandits with a new kind of smoothness. In *Advances in Neural Information Processing Systems 28*, 2015.

Jean-Yves Audibert and Sébastien Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11(Oct):2785–2836, 2010.

Gergely Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. In *Advances in Neural Information Processing Systems 28*, 2015.

Lecture 14

Instructor: Haipeng Luo

1 Stochastic Multi-armed Bandit

In the last two lectures we have seen algorithms for the *adversarial* multi-armed bandit problem when there is no assumption on how the loss vectors are generated. On the other hand, there is a also huge literature on the *stochastic* version of the multi-armed bandit problem, where each arm represents an unknown distribution and each pull of the arm generates an independent sample of the corresponding distribution.

While the problem setting is clearly just a special case of its adversarial version, the goal for stochastic bandit is usually to derive regret bounds that are distribution-dependent and in some situations stronger than the worst-case $\mathcal{O}(\sqrt{TK})$ bound. Moreover, although in the full information setting the stochastic assumption makes the problem much easier (indeed, FTL would solve the problem already), in the bandit setting, due to the lack of feedback the problem is still quite challenging even with the stochastic assumption.

Formally, we assume that for each action a, there is a unknown distribution $\mathcal{D}_a$ with mean $\mu(a)$ such that $\ell_1(a), \ldots, \ell_T(a)$ are independent samples of $\mathcal{D}_a$. Let $a^\star = \operatorname{argmin}_a \mu(a)$ be the optimal action in terms of the expected loss. For this problem we usually care about the a slightly different version of regret, called *pseudo-regret*, defined as

$$\bar{\mathcal{R}}_T = \mathbb{E}\left[\sum_{t=1}^{T} \left(\ell_t(a_t) - \ell_t(a^\star)\right)\right]$$

which is the expected regret against the fixed action $a^\star$ (instead of the empirically best action $\operatorname{argmin}_a \sum_t \ell_t(a)$), where the expectation is over the randomness of both the environment and the algorithm. Clearly the pseudo-regret can also be simplified as

$$\bar{\mathcal{R}}_T = \mathbb{E}\left[\sum_{t=1}^{T} \left(\mu(a_t) - \mu(a^\star)\right)\right] = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{a=1}^{K} \Delta_a \mathbf{1}\{a_t = a\}\right] = \sum_{a:\Delta_a>0} \Delta_a \mathbb{E}[n_T(a)]$$

where $\Delta_a = \mu(a) - \mu(a^\star)$ is called the suboptimality gap of action a and $n_t(a) = \sum_{\tau=1}^{t} \mathbf{1}\{a_\tau = a\}$ is the number of times action a has been pulled up to round t. Therefore, to analyze an algorithm in this setting, it boils down to bounding the term $\mathbb{E}[n_T(a)]$.

In the stochastic setting, the tradeoff between exploration and exploitation is perhaps even more intuitive. Let $\widehat{\mu}_t(a) = \frac{1}{n_t(a)} \sum_{\tau=1}^{t} \mathbf{1}\{a_\tau = a\}\ell_\tau(a)$ be the empirical mean of action a up to round t. Since the environment is stochastic, $\widehat{\mu}_t(a)$ could be a good approximation of $\mu(a)$ if $n_t(a)$ is large enough. Therefore, on one hand we want to exploit by picking the empirically best action $\operatorname{argmin}_a \widehat{\mu}_t(a)$, but on the other hand we also need to explore so that all actions are picked frequently enough and $\widehat{\mu}_t(a)$ is truly a good approximation of $\mu(a)$.

2 First Attempt: Explore-then-exploit

The simplest strategy to balance the tradeoff is to first perform pure exploration for a while, and then do pure exploitation and commit to a single action for the rest of the time. Formally, let T_0 be the number of exploration rounds to be specified later. The explore-then-exploit strategy is as follows:

1. For the first T_0 rounds, pick each action for T_0/K times (in an arbitrary order);
2. For the remaining $T - T_0$ rounds, always pick $\widehat{a} = \operatorname{argmin}_a \widehat{\mu}_{T_0}(a)$.

One can then show the following regret bound.

Theorem 1. *The pseudo-regret of explore-then-exploit is bounded as*

$$\bar{\mathcal{R}}_T \leq \sum_{a:\Delta_a>0} \left(\frac{T_0}{K} + 2T \exp\left(-\frac{T_0 \Delta_a^2}{8K} \right) \right) \Delta_a.$$

Proof. It suffices to prove that $\mathbb{E}[n_T(a)] \leq \frac{T_0}{K} + 2T \exp\left(-\frac{T_0\Delta_a^2}{8K}\right)$ for all a with $\Delta_a > 0$. Indeed, by the algorithm it is clear that

$$\mathbb{E}[n_T(a)] = \frac{T_0}{K} + (T - T_0)\mathbb{E}[\mathbf{1}\{\widehat{a} = a\}] = \frac{T_0}{K} + (T - T_0)\Pr(\widehat{a} = a),$$

and also $\Pr(\widehat{a} = a) \leq \Pr(\widehat{\mu}_{T_0}(a) \leq \widehat{\mu}_{T_0}(a^\star))$. Next note that if $\widehat{\mu}_{T_0}(a) \leq \widehat{\mu}_{T_0}(a^\star)$ happens, then one of the following two rare events must happen

$$\widehat{\mu}_{T_0}(a) \leq \mu(a) - \Delta_a/2$$
$$\widehat{\mu}_{T_0}(a^\star) \geq \mu(a^\star) + \Delta_a/2$$

since otherwise $\widehat{\mu}_{T_0}(a) > \mu(a) - \Delta_a/2 = \mu(a^\star) + \Delta_a/2 > \widehat{\mu}_{T_0}(a^\star)$. Now recall $\widehat{\mu}_{T_0}(a)$ ($\widehat{\mu}_{T_0}(a^\star)$) is the average of T_0/K i.i.d. samples of a distribution with mean $\mu(a)$ ($\mu(a^\star)$), and thus by Hoeffding's inequality (included at the end of the section) we know that the probability of each of the above two events is bounded by $\exp\left(-\frac{T_0\Delta_a^2}{8K}\right)$. A union bound then implies that $\Pr(\widehat{a} = a) \leq 2\exp\left(-\frac{T_0\Delta_a^2}{8K}\right)$, completing the proof. □

How should we choose the parameter T_0? For simplicity, let's consider the case when there are only two actions so that the optimal T_0 is such that $\frac{T_0}{2} + 2T \exp\left(-\frac{T_0\Delta^2}{16}\right)$ is minimized (Δ is the only non-zero gap). Direct calculations show that the optimal T_0 is $\frac{16}{\Delta^2} \ln\left(\frac{T\Delta^2}{4}\right)$ and the bound becomes

$$\frac{8}{\Delta}\left(1 + \ln\left(\frac{T\Delta^2}{4}\right)\right), \tag{1}$$

which only has a logarithmic dependence in T and is an instance-dependent bound that is better than the worst-case $\mathcal{O}(\sqrt{TK})$ bound as long as Δ is not too small. Note that this does not contradict with the lower bound $\Omega(\sqrt{TK})$ that we discussed previously. Indeed, recall that in the proof of the lower bound, the construction of the environment is also stochastic, but the gap is as small as $1/\sqrt{T}$.

The instance-dependent bounds we have seen before (mainly for the expert problem) are never worse than the worst-case bound, but bound (1) can actually be arbitrarily large if Δ is too small. However, while smaller Δ indeed increases the difficulty of distinguishing the best action from the suboptimal one, at the same time it also means that the picking the suboptimal action is not too terrible – it only incurs a regret Δ per round. This means bound (1) is loose for small Δ, but one can simply tighten it as

$$\min\left\{T\Delta,\ \frac{8}{\Delta}\left(1 + \ln\left(\frac{T\Delta^2}{4}\right)\right)\right\},$$

which is at most $\mathcal{O}(\sqrt{T \ln T})$ (by maximizing over Δ), meaning that the bound is never much worse than the one by using Exp3 or other adversarial bandit algorithms.

Bound (1) is in fact close to optimal for a fixed Δ, so this simple tradeoff between exploration and exploitation does work pretty well, at least in theory. However, the big caveat is that T_0 has to be tuned according to the suboptimality gaps, which are clearly unknown in practice. Moreover if T_0 is independent of the gap, one can show that the pseudo-regret can in fact be as large as $\Omega(T^{\frac{2}{3}})$. In the next section we will discuss an algorithm that addresses this issue completely.

Lemma 1 (Hoeffding's inequality)**.** *Let $X_1, \ldots, X_T \in [-B, B]$ for some $B > 0$ be independent random variables such that $\mathbb{E}[X_t] = 0$ for all $t \in [T]$, then we have for all $\delta \in (0, 1)$,*

$$\Pr\left(\sum_{t=1}^{T} X_t \geq B\sqrt{2T \ln \tfrac{1}{\delta}}\right) \leq \delta.$$

3 The UCB Algorithm

The classic algorithm for stochastic multi-armed bandit is the UCB (Upper Confidence Bound) algorithm [Auer et al., 2002], although since we use "losses" instead of "rewards" (which was used traditionally in [Auer et al., 2002]), the algorithm that we will discuss here is actually LCB (Lower Confidence Bound). For convention, we will still call it the UCB algorithm.

UCB applies a very important principle called "optimism in face of uncertainty", which is useful in many other stochastic problems with bandit feedback. The main idea of the principle is the following: among all plausible environments that are consistent with the data observed, assume the most favorable one is the true environment and act accordingly.

Let's apply this principle to stochastic multi-armed bandit. At time t, we have gather empirical averages $\widehat{\mu}_{t-1}(a)$ for each action a. What are the plausible environments, that is, the plausible values of the means $\mu(a)$, given this information? In light of Hoeffding's inequality, with high probability the mean $\mu(a)$ should be in the confidence interval (ignoring constants and logarithmic terms)

$$\left[\widehat{\mu}_{t-1}(a) - 1/\sqrt{n_{t-1}(a)}, \widehat{\mu}_{t-1}(a) + 1/\sqrt{n_{t-1}(a)}\right].$$

Having these plausible environments, we will then ask which is the most favorable one. Since our goal here is to suffer as less loss as possible, the best scenario is thus when $\mu(a)$ is exactly $\widehat{\mu}_{t-1}(a) - 1/\sqrt{n_{t-1}(a)}$ (called the lower confidence bound) for each a. Finally, we will simply be optimistic and assume that this is indeed the true environment and act according to it, which in this case will mean picking the action with the smallest lower confidence bound.

Formally, with constants and logarithmic terms carefully chosen, we define the lower confidence bound for action a at time t as

$$\mathrm{LCB}_t(a) = \widehat{\mu}_{t-1}(a) - 2\sqrt{\frac{\ln T}{n_{t-1}(a)}}.$$

Then at time t the UCB algorithm simply picks

$$a_t = \operatorname*{argmin}_{a \in [K]} \mathrm{LCB}_t(a).$$

First of all, note that $n_{t-1}(a)$ is initially 0, leading to negative infinity for $\mathrm{LCB}_t(a)$, which means the algorithm will be forced to pick each action once for the first K rounds. Afterwards, the two terms in $\mathrm{LCB}_t(a)$ are essentially playing the role of exploitation and exploration respectively since it suggests picking action with low empirical mean but penalized by how many times it has been selected. Whenever a suboptimal action is picked, its lower confidence bound will most likely go up and as a result it is less likely to be picked again in the future, which means optimism drives exploration. (Indeed, think about what happens to a pessimistic strategy that picks the action with the lowest *upper* confidence bound instead).

Notice that in contrast to randomized algorithms such as Exp3, both UCB and the explore-then-exploit strategy are deterministic algorithms – there is no randomness from the algorithms themselves. Importantly, UCB does not need to know the gaps Δ_a and is a very simple and practical algorithm (even the $\ln T$ term in $\mathrm{LCB}_t(a)$ can in fact be replaced by $\ln t$ to make the algorithm truly parameter-free). We finally prove the following bound for UCB that is in the same spirit of Eq. (1).

Theorem 2. *The pseudo-regret of UCB is bounded as*

$$\bar{\mathcal{R}}_T \leq \sum_{a:\Delta_a > 0} \left(\frac{16 \ln T}{\Delta_a} + 2\Delta_a\right)$$

Proof. Again it suffices to bound $\mathbb{E}[n_T(a)]$ by $\frac{16 \ln T}{\Delta_a^2} + 2$. Intuitively, for the first small number of rounds n (to be specified later), the concentration bounds are loose and there is nothing much to say. Therefore, we simply ignore these rounds and bound $\mathbb{E}[n_T(a)]$ by

$$n + \sum_{t=n+1}^{T} \Pr\left(a_t = a \text{ and } n_{t-1}(a) > n\right).$$

Note that n is similar to the number of pure exploration rounds T_0/K in the proof of explore-then-exploit, but the important thing is that n is merely for the analysis and is not a parameter of the algorithm. Similarly, the event $a_t = a$ happens only if one of the following two rare events happens

$$\begin{aligned} \text{LCB}_t(a^\star) &\geq \mu(a^\star) \\ \text{LCB}_t(a) &\leq \mu(a^\star) \end{aligned}$$

since otherwise $\text{LCB}_t(a) > \mu(a^\star) > \text{LCB}_t(a^\star)$ and a will not be picked according to the algorithm. Therefore we have by a union bound, $\Pr\left(a_t = a \text{ and } n_{t-1}(a) > n\right)$ is bounded by

$$\Pr\left(\text{LCB}_t(a^\star) \geq \mu(a^\star)\right) + \Pr\left(\text{LCB}_t(a) \leq \mu(a^\star) \text{ and } n_{t-1}(a) > n\right).$$

The first term, which is equivalent to

$$\Pr\left(\widehat{\mu}_{t-1}(a^\star) - \mu(a^\star) \geq 2\sqrt{\frac{\ln T}{n_{t-1}(a^\star)}}\right),$$

could be seemingly bounded by applying Hoeffding's inequality directly. However, one trap here is that $n_{t-1}(a)$ is actually also a random variable depending on the samples we observe. To deal with this subtle issue, we can imagine that there is a (infinite) sequence $X_1(a), X_2(a), \ldots$ of independent samples of $\mathcal{D}_a$ for each action a, and at time t the observed loss $\ell_t(a_t)$ is the $n_t(a_t)$-th sample of this sequence, that is, $\ell_t(a_t) = X_{n_t(a_t)}(a_t)$. With $\widetilde{\mu}_m(a) = \frac{1}{m}\sum_{k=1}^m X_m(a)$ being the average of the first m samples of this sequence, we then have $\widehat{\mu}_{t-1}(a) = \widetilde{\mu}_{n_{t-1}(a)}(a)$ and

$$\begin{aligned} &\Pr\left(\widehat{\mu}_{t-1}(a^\star) - \mu(a^\star) \geq 2\sqrt{\frac{\ln T}{n_{t-1}(a^\star)}}\right) \\ \leq\ &\Pr\left(\exists\, k \in [t-1] \ \text{ s.t. } \widetilde{\mu}_k(a^\star) - \mu(a^\star) \geq 2\sqrt{\frac{\ln T}{k}}\right) \\ \leq\ &\sum_{k=1}^{t-1} \Pr\left(\widetilde{\mu}_k(a^\star) - \mu(a^\star) \geq 2\sqrt{\frac{\ln T}{k}}\right), \end{aligned}$$

where each term is the last summation can now be bounded by $1/T^2$ using Hoeffding's inequality since k is fixed, and the summation is bounded by $1/T$.

For the second term $\Pr\left(\text{LCB}_t(a) \leq \mu(a^\star) \text{ and } n_{t-1}(a) > n\right)$, note that it is equivalent to

$$\begin{aligned} &\Pr\left(\widehat{\mu}_{t-1}(a) - 2\sqrt{\frac{\ln T}{n_{t-1}(a)}} \leq \mu(a^\star) \text{ and } n_{t-1}(a) > n\right) \\ =\ &\Pr\left(\Delta_a - 2\sqrt{\frac{\ln T}{n_{t-1}(a)}} \leq \mu(a) - \widehat{\mu}_{t-1}(a) \text{ and } n_{t-1}(a) > n\right) \end{aligned}$$

and thus by picking $n = \lfloor \frac{16 \ln T}{\Delta_a^2} \rfloor$, it is bounded by

$$\Pr\left(2\sqrt{\frac{\ln T}{n_{t-1}(a)}} \leq \mu(a) - \widehat{\mu}_{t-1}(a)\right),$$

which by the exact same argument as before is further bounded by $1/T$. This finishes the proof. □

It can be shown that the above bound for UCB is very close to optimal. Moreover, even though the bound can be arbitrarily large when the gaps are small, one can still show that the worst-case pseudo-regret for UCB is of order $\mathcal{O}(\sqrt{TK \ln T})$ (see Homework 3).

References

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

Lecture 15

Instructor: Haipeng Luo

1 Stochastic Linear Bandit

In this lecture we introduce yet another classic stochastic bandit model, called *stochastic linear bandit*, and discuss how to use the same principle of "optimism in face of uncertainty" to solve it. There is also a huge literature on this topic and the following discussions follow mostly [Abbasi-Yadkori et al., 2011].

The learning protocol is as follows: for each round $t = 1, \ldots, T$,

1. A set of actions $A_t \subset \mathbb{R}^d$ is revealed to the learner;
2. the learner picks an action $a_t \in A_t$ and observe its loss $c_t = \langle a_t, \theta^\star \rangle + \epsilon_t$ where $\theta^\star \in \mathbb{R}^d$ is an unknown parameter and $\epsilon_t \sim \mathcal{N}(0, 1)$ is independent standard Gaussian noise.

Let $a_t^\star = \operatorname{argmin}_{a \in A_t} \langle a, \theta^\star \rangle$ be the optimal action at time t. The pseudo-regret for this problem is defined as

$$\bar{\mathcal{R}}_T = \mathbb{E}\left[\sum_{t=1}^{T} \langle a_t - a_t^\star, \theta^\star \rangle\right]. \tag{1}$$

First of all, the stochastic multi-armed bandit model we discussed last time is clearly a special case with $d = K$, $A_t = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\}$ (that is, the standard basis of $\mathbb{R}^d$) and $\theta^\star = (\mu_1, \ldots, \mu_d)$ be the vector of loss means for the actions, except that for simplicity we only consider Gaussian noise now.

In general, the stochastic linear bandit model is much more powerful since it allows each action to come with an arbitrary "feature", and moreover the set of available actions can be different at different time. This allows the model to capture real-life problems such as building a personalized news recommendation system [Li et al., 2010]. In this example, each time t corresponds to a visit of some user to the website. The available news articles at that time as well as the user's information are then used to generate a feature vector for each article. Afterwards a linear bandit algorithm somehow selects an action and recommends the corresponding article to the user. The loss is then based on whether the user clicks on the recommended article or not. It is assumed that the expected loss of an action can be perfectly predicted by an unknown linear predictor $\theta^\star$, but generalization to nonlinear models is possible.

Note that because of the changing action sets, it only makes sense to define the pseudo-regret as in Eq. (1) so that it compares the expected loss of the algorithm to the expected loss of the best action *at each time*. This relates to the notion of dynamic regret discussed before. However, while in general sublinear dynamic regret is impossible, due to the stochastic assumption, regret of order $\mathcal{O}(\sqrt{T})$ is in fact achievable here as we will show soon.

Finally without loss of generality, we make two scaling assumptions: $\max_{a \in A_t} \|a\|_2 \leq 1$ for all t and $\|\theta^\star\|_2 \leq 1$.

2 LinUCB

Let's apply the same "optimism in face of uncertainty" principle to come up with an algorithm for this problem. Recall that the first step is to come up with the set of plausible environments that are

consistent with the observed data. The only parameter of the environment here is the linear predictor $\theta^\star$. So the first goal would be to come up with a confidence set Θ_t based on $a_1, c_1, \ldots, a_t, c_t$ so that $\theta^\star \in \Theta_t$ with high probability. With such a confidence set, similar to the UCB algorithm at time $t+1$ we optimistically assume that the loss for action $a \in A_{t+1}$ is

$$\mathrm{LCB}_{t+1}(a) = \min_{\theta \in \Theta_t} \langle a, \theta \rangle ,$$

and finally pick action $a_{t+1} = \operatorname{argmin}_{a \in A_{t+1}} \mathrm{LCB}_{t+1}(a)$.

It remains to come up with the confidence set Θ_t. First we need to figure out what the "center" of this set is. For UCB, the center of the confidence set is simply and naturally the empirical mean of losses. For linear bandit, note that we are observing $c_\tau \approx \langle a_\tau, \theta^\star \rangle$ for $\tau = 1, \ldots, t$. It is thus natural to perform least square regression to obtain an estimate of $\theta^\star$ as the center:

$$\widehat{\theta}_t = \operatorname*{argmin}_{\theta \in \mathbb{R}^d} \sum_{\tau=1}^{t} \left(\langle a_\tau, \theta \rangle - c_\tau \right)^2 .$$

By direct calculations one can verify that $\widehat{\theta}_t = M_t^{-1} \sum_{\tau=1}^t c_\tau a_\tau$ where $M_t = \sum_{\tau=1}^t a_\tau a_\tau^\top$ is the covariance matrix and is assumed to be invertible for now. Note that this is consistent with UCB: when $A_t = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\}$, M_t is a diagonal matrix with $M_t(i,i)$ being the number of times action i has been picked, and $\widehat{\theta}_t$ is exactly the vector of empirical means of actions.

By plugging $c_\tau = \langle a_\tau, \theta^\star \rangle + \epsilon_\tau$, we can also rewrite $\widehat{\theta}_t$ as

$$\widehat{\theta}_t = \left(M_t^{-1} \sum_{\tau=1}^{t} (\langle a_\tau, \theta^\star \rangle + \epsilon_\tau) a_\tau \right) = M_t^{-1} M_t \theta^\star + M_t^{-1} \sum_{\tau=1}^{t} \epsilon_\tau a_\tau = \theta^\star + M_t^{-1} Z_t$$

where $Z_t = \sum_{\tau=1}^t \epsilon_\tau a_\tau$. Next we need to figure out what Θ_t should look like around the center $\widehat{\theta}_t$. To get the intuition, we first ignore the fact that a_τ's are random variables and think of them as fixed vectors (all assumptions mentioned so far will be dropped eventually). Then we have that Z_t is a zero-mean d-dimensional Gaussian variable with covariance matrix

$$\mathbb{E}\left[Z_t Z_t^\top \right] = \sum_{\tau_1=1}^{t} \sum_{\tau_2=1}^{t} \mathbb{E}\left[\epsilon_{\tau_1} \epsilon_{\tau_2} \right] a_{\tau_1} a_{\tau_2}^\top = \sum_{\tau=1}^{t} \mathbb{E}\left[\epsilon_\tau^2 \right] a_\tau a_\tau^\top = M_t .$$

Therefore the random variable $M_t^{1/2}(\widehat{\theta}_t - \theta^\star)$ is actually distributed as $\mathcal{N}(0, I)$, the d-dimensional standard Gaussian. The question thus transfers to finding a region $S \in \mathbb{R}^d$ so that $\Pr(X \in S) \geq 1 - \delta$ if $X \sim \mathcal{N}(0, I)$. By standard results (specifically tail bounds of χ_d^2 distribution), S can be chosen as an ℓ_2-ball with squared radius $d + 2\sqrt{d \ln \frac{1}{\delta}} + 2 \ln \frac{1}{\delta}$. In other words,

$$\Pr\left(\left\| M_t^{1/2} (\widehat{\theta}_t - \theta^\star) \right\|_2^2 \leq d + 2\sqrt{d \ln \tfrac{1}{\delta}} + 2 \ln \tfrac{1}{\delta} \right) \geq 1 - \delta,$$

and the confidence set can thus be

$$\Theta_t = \left\{ \theta \in \mathbb{R}^d : \left\| M_t^{1/2} (\theta - \widehat{\theta}_t) \right\|_2^2 \leq d + 2\sqrt{d \ln \tfrac{1}{\delta}} + 2 \ln \tfrac{1}{\delta} \right\}$$

which is in fact an ellipsoid centered at $\widehat{\theta}_t$. Usually $\left\| M^{1/2} v \right\|_2 = \sqrt{v^\top M v}$ is compactly written as $\|v\|_M$, which is indeed a norm when M is positive definite. The set defined by $\|v\|_M \leq 1$ is the standard analytic form of an ellipsoid centered at the origin. The eigenvectors of M define the principal axes of the ellipsoid while the eigenvalues are the reciprocals of the squares of the semi-axes.

In the process of deriving such a ellipsoidal confidence set, we made two assumptions. First, M_t is invertible, which is not true until $a_1, \ldots, a_t$ span $\mathbb{R}^d$. This can be resolved by adding an ℓ_2-regularization to the least square regression

$$\widehat{\theta}_t = \operatorname*{argmin}_{\theta \in \mathbb{R}^d} \sum_{\tau=1}^{t} \left(\langle a_\tau, \theta \rangle - c_\tau \right)^2 + \lambda \|\theta\|_2^2 = M_t^{-1} \sum_{\tau=1}^{t} c_\tau a_\tau$$

where M_t is redefined as $\lambda I + \sum_{\tau=1}^{t} a_\tau a_\tau^\top$ for some parameter $\lambda > 0$ and is always invertible now. Similar confidence sets can be constructed based on this new M_t. However, what is more difficult to get rid of is the second assumption that a_t's are fixed and not random. Fortunately, with fancier probability tools, this can still be addressed. Specifically, the following lemma was proven in [Abbasi-Yadkori et al., 2011].

Lemma 1 (Confidence Ellipsoid)**.** *Let* $M_t = \lambda I + \sum_{\tau=1}^{t} a_\tau a_\tau^\top$, $\widehat{\theta}_t = M_t^{-1} \sum_{\tau=1}^{t} c_\tau a_\tau$, $\beta_t = \sqrt{\lambda} + \sqrt{2 \ln \frac{1}{\delta} + d \ln\left(1 + \frac{t}{d\lambda}\right)}$*, and*

$$\Theta_t = \left\{ \theta \in \mathbb{R}^d : \left\| \theta - \widehat{\theta}_t \right\|_{M_t} \leq \beta_t \right\}.$$

Then no matter how a_t*'s are chosen, with probability* $1 - \delta$*,* $\theta^\star \in \Theta_t$ *holds for all* t*.*

Finally, having this confidence set Θ_t, we can further simplify the algorithm by noting

$$\begin{aligned}
\mathrm{LCB}_{t+1}(a) &= \min_{\theta \in \Theta_t} \langle a, \theta \rangle = \min_{\|\theta - \widehat{\theta}_t\|_{M_t} \leq \beta_t} \langle a, \theta \rangle \\
&= \min_{\|\theta'\|_2 \leq \beta_t} \left\langle a, M_t^{-\frac{1}{2}} \theta' + \widehat{\theta}_t \right\rangle \qquad \text{(by changing variable } \theta' = M_t^{\frac{1}{2}}(\theta - \widehat{\theta}_t)\text{)} \\
&= \left\langle a, \widehat{\theta}_t \right\rangle + \min_{\|\theta'\|_2 \leq \beta_t} \left\langle M_t^{-\frac{1}{2}} a, \theta' \right\rangle \\
&= \left\langle a, \widehat{\theta}_t \right\rangle - \beta_t \|a\|_{M_t^{-1}}
\end{aligned}$$

and thus

$$a_{t+1} = \operatorname*{argmin}_{a \in A_{t+1}} \mathrm{LCB}_{t+1}(a) = \operatorname*{argmin}_{a \in A_{t+1}} \left(\left\langle a, \widehat{\theta}_t \right\rangle - \beta_t \|a\|_{M_t^{-1}} \right).$$

This algorithm is called by many names, such as LinUCB or OFUL (Optimism in Face of Uncertainty for Linear bandit). Very similar to UCB, the term $\left\langle a, \widehat{\theta}_t \right\rangle$ drives exploitation while the term $-\beta_t \|a\|_{M_t^{-1}}$ drives exploration of unobserved directions. Indeed, when $A_t = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\}$, one can verify that LinUCB has the same form of UCB.

3 Regret Analysis

We have so far directly applied the "optimism in face of uncertainty" principle to derive the LinUCB algorithm. The final step is to prove a regret bound for this algorithm.

Theorem 1. *If* $\lambda \geq 1$*, then the pseudo-regret of LinUCB is bounded as*

$$\bar{\mathcal{R}}_T \leq 2T\delta + \beta_T \sqrt{8dT \ln\left(1 + \frac{T}{\lambda d}\right)}.$$

Setting $\lambda = 1$ *and* $\delta = 1/T$ *leads to* $\bar{\mathcal{R}}_T = \mathcal{O}(d \ln(T/d) \sqrt{T})$.

Proof. Since $\langle a_t - a_t^\star, \theta^\star \rangle \leq |\langle a_t, \theta^\star \rangle| + |\langle a_t^\star, \theta^\star \rangle| \leq 2$, if suffices to show that under the event $\theta^\star \in \Theta_t$ for all t (which happens with probability $1 - \delta$ according to Lemma 1), we have $\bar{\mathcal{R}}_T \leq \beta_T \sqrt{8dT \ln\left(1 + \frac{T}{\lambda d}\right)}$. Indeed, notice that for any a,

$$\left| \left\langle a, \theta^\star - \widehat{\theta}_t \right\rangle \right| \leq \left\| \theta^\star - \widehat{\theta}_t \right\|_{M_t} \|a\|_{M_t^{-1}} \leq \beta_t \|a\|_{M_t^{-1}}.$$

Therefore with $a = a_{t+1}$ and $a = a_{t+1}^\star$, we have

$$\langle a_{t+1}, \theta^\star \rangle \leq \left\langle a_{t+1}, \widehat{\theta}_t \right\rangle + \beta_t \|a_{t+1}\|_{M_t^{-1}}$$

and

$$\left\langle a_{t+1}^\star, \theta^\star \right\rangle \geq \left\langle a_{t+1}^\star, \widehat{\theta}_t \right\rangle - \beta_t \left\| a_{t+1}^\star \right\|_{M_t^{-1}} \geq \left\langle a_{t+1}, \widehat{\theta}_t \right\rangle - \beta_t \|a_{t+1}\|_{M_t^{-1}}$$

where the last inequality is by the algorithm. Combining the above two inequalities we have

$$\left\langle a_{t+1} - a^\star_{t+1}, \theta^\star \right\rangle \leq 2\beta_t \left\| a_{t+1} \right\|_{M_t^{-1}} \leq 2\beta_T \left\| a_{t+1} \right\|_{M_t^{-1}}.$$

With the trivial bound shown at the beginning and $\beta_T \geq \sqrt{\lambda} \geq 1$, we have

$$\left\langle a_{t+1} - a^\star_{t+1}, \theta^\star \right\rangle \leq 2\beta_T \min\{1, \left\| a_{t+1} \right\|_{M_t^{-1}}\}.$$

and therefore by Cauchy-Schwarz inequality and $\min\{1, x\} \leq 2\ln(1+x)$, the regret is bounded by

$$\sqrt{T\sum_{t=1}^{T}(\langle a_t - a^\star_t, \theta^\star\rangle)^2} \leq \beta_T \sqrt{4T\sum_{t=1}^{T}\min\left\{1, \|a_t\|^2_{M_{t-1}^{-1}}\right\}}$$
$$\leq \beta_T \sqrt{8T\sum_{t=1}^{T}\ln\left(1+\|a_t\|^2_{M_{t-1}^{-1}}\right)} = \beta_T \sqrt{8T\ln\prod_{t=1}^{T}\left(1+\|a_t\|^2_{M_{t-1}^{-1}}\right)}.$$

Next by the fact that $\det(AB) = \det(A)\det(B)$ and $I + vv^\top$ has only two eigenvalues 1 and $1 + \|v\|_2^2$, we have

$$\det(M_T) = \det(M_{T-1} + a_T a_T^\top) = \det(M_{T-1}^{\frac{1}{2}}(I + M_{T-1}^{-\frac{1}{2}} a_T a_T^\top M_{T-1}^{-\frac{1}{2}}) M_{T-1}^{\frac{1}{2}})$$
$$= \det(M_{T-1})\det(I + M_{T-1}^{-\frac{1}{2}} a_T a_T^\top M_{T-1}^{-\frac{1}{2}}) = \det(M_{T-1})(1 + \|a_T\|^2_{M_{T-1}^{-1}})$$
$$= \cdots = \det(M_0)\prod_{t=1}^{T}\left(1 + \|a_t\|^2_{M_{t-1}^{-1}}\right).$$

It thus remains to show $\ln\frac{\det(M_T)}{\det(M_0)} \leq d\ln\left(1 + \frac{T}{\lambda d}\right)$. This is because by AM-GM inequality,

$$\det(M_T) \leq \left(\frac{\mathrm{TR}(M_T)}{d}\right)^d = \left(\frac{\lambda d + \sum_{t=1}^{T}\mathrm{TR}(a_t a_t^\top)}{d}\right)^d = \left(\lambda + \frac{\sum_{t=1}^{T}\mathrm{TR}(a_t^\top a_t)}{d}\right)^d \leq \left(\lambda + \frac{T}{d}\right)^d.$$

This finishes the proof together with $\det(M_0) = \lambda^d$. □

This regret bound for LinUCB has a linear dependence on the dimension d, which was shown to be optimal in the worst case (but suboptimal for the special case of multi-armed bandit where the dependence should be $\sqrt{d}$).

As a final remark, note that just as UCB, one can also derive a "gap-dependent" regret bound for LinUCB. Specifically, let the minimal suboptimal gap be

$$\Delta = \min_{t\in[T]}\min_{a\in A_t : \langle a - a^\star_t, \theta^\star\rangle > 0} \langle a - a^\star_t, \theta^\star\rangle.$$

Then one has either $\Delta \leq \langle a_t - a^\star_t, \theta^\star\rangle$ or $\langle a_t - a^\star_t, \theta^\star\rangle = 0$, and therefore

$$\bar{\mathcal{R}}_T \leq \frac{1}{\Delta}\sum_{t=1}^{T}(\langle a_t - a^\star_t, \theta^\star\rangle)^2,$$

where the last summation can be upper bounded in the exact same way as the proof above. This shows a regret of order $\mathcal{O}\left(\frac{(d\ln(T/d))^2}{\Delta}\right)$.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24*, 2011.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

Lecture 16

Instructor: Haipeng Luo

1 Adversarial Linear Bandit and Exp2

Previously we have discussed bandit problems under some stochastic assumptions. In this lecture we come back to adversarial setting and discuss the adversarial linear bandit problem, a natural generalization of adversarial multi-armed bandit. Specifically, the setting is as follows: at each time $t = 1, \ldots, T$,

1. learner picks action $a_t \in A \subset \mathbb{R}^d$ while simultaneously environment picks $\ell_t \in \mathbb{R}^d$;
2. learner suffers and observes $a_t^\top \ell_t$.

We assume that $\max_{a \in A} \|a\|_2 \leq B$ for some constant $B > 0$, $|a_t^\top \ell_t| \leq 1$, and A is a finite set with cardinality $|A| = K$. For simplicity we also assume that the environment is oblivious. The expected regret of the learner is

$$\mathbb{E}[\mathcal{R}_T] = \mathbb{E}\left[\sum_{t=1}^T a_t^\top \ell_t\right] - \min_{a \in A} \sum_{t=1}^T a^\top \ell_t.$$

Compared to the stochastic linear bandit, the main difference is that the action set A for the learner is now fixed while the loss vector ℓ_t is changing over time. Adversarial multi-armed bandit a clearly a special case of this model with A being the standard basis of $\mathbb{R}^d$. However, on the other hand one can also see linear bandit as a special case of multi-armed bandit with K actions, completely ignoring the underlying linear structure of the losses. This gives a trivial solution with regret of order $\mathcal{O}(\sqrt{TK})$, independent of d. However, as we will see soon, by exploiting the linear structure, one can in fact achieve regret $\mathcal{O}(\sqrt{dT \ln K})$, much better than the trivial solution as long as $d \ll K$.

Nevertheless, we can still borrow the idea of Exp3, the classic solution for adversarial multi-armed bandit. Recall that Exp3 is simply feeding the Hedge algorithm with unbiased loss estimators. Due to the similarity, it is natural to try the same idea here:

1. play $a_t \sim p_t \in \Delta(K)$ and observe $a_t^\top \ell_t$;
2. construct unbiased loss estimator $\widehat{\ell}_t$ based on $a_t^\top \ell_t$;
3. update $p_{t+1}(a) \propto \exp(\eta \sum_{\tau=1}^t a^\top \widehat{\ell}_t)$.

Note that instead of estimating the loss of each action, here we estimate the underlying loss vector ℓ_t directly. The key is therefore to come up with this estimator. Recall that in the last lecture we also constructed some estimator for the true parameter $\theta^\star$ for stochastic linear bandit, and it was simply based on least square regression using observed data. The difficulty here is that we have only one single observation about ℓ_t. However, because of the randomization in picking a_t, it turns out that one can construct the following estimator using a very similar formula as for stochastic linear bandit

$$\widehat{\ell}_t = M_t^{-1} a_t a_t^\top \ell_t \quad \text{where} \quad M_t = \sum_{a \in A} p_t(a) a a^\top = \mathbb{E}_{a \sim p_t}\left[a a^\top\right].$$

Note that although ℓ_t appears in this formula, the dependence is only through $a_t^\top \ell_t$, a quantity that we indeed observe. M_t is assumed to be invertible here, which is equivalent to assuming A is full

rank. Unlike the case for stochastic linear bandit, this is in fact without loss of generality. Indeed, if A is not full rank, then before the game starts one can do a preprocessing step to project the actions into a subspace with lower dimension. (Alternatively, one can also simply replace "inverse" by "pseudo-inverse" and verify that it does not change the final results.)

With $\mathbb{E}_t$ being the conditional expectation with respect to the random draw of a_t, direct calculations show that this estimator is indeed unbiased:

$$\mathbb{E}_t[\widehat{\ell}_t] = M_t^{-1}\mathbb{E}\left[a_t a_t^\top\right]\ell_t = M_t^{-1}M_t\ell_t = \ell_t.$$

Also, when A is the standard basis of $\mathbb{R}^d$, M_t is a diagonal matrix with $M_t(a,a) = p_t(a)$ and thus this recovers the importance weighted estimator used in Exp3.

There is one more detail we need to take care of before applying the result of Hedge. Recall that in the analysis of Hedge, we use the inequality $e^{-x} \le 1 - x + x^2$ for $x \ge 0$ where x corresponds to $\eta a^\top \widehat{\ell}_t$ here. While for multi-armed bandit this is indeed non-negative (or at least can be made to be nonnegative by shifting the losses), this is not true anymore for the general linear case, even if all $a \in A$ and ℓ_t have nonnegative coordinates.

Fortunately, the inequality in fact holds whenever $x \ge -1$. So at least negativity is not necessarily an issue. We do, however, still need to control the magnitude of $\eta a^\top \widehat{\ell}_t$. We will ensure this by enforcing an explicit exploration. Specifically, let $q \in \Delta(K)$ be a fixed exploration distribution over the actions in A and γ be some exploration parameter. We modify the algorithm as (first two steps remain the same):

1. play $a_t \sim p_t \in \Delta(K)$ and observe $a_t^\top \ell_t$;
2. construct unbiased loss estimator $\widehat{\ell}_t = M_t^{-1} a_t a_t^\top \ell_t$;
3. update $p'_{t+1}(a) \propto \exp(\eta \sum_{\tau=1}^t a^\top \widehat{\ell}_t)$;
4. compute $p_{t+1} = (1-\gamma)p'_{t+1} + \gamma q$.

With the explicit exploration, we can show that the magnitude of $|a^\top \widehat{\ell}_t|$ is controlled by the minimum eigenvalue of $\mathbb{E}_{a\sim q}[aa^\top]$ as shown by the following lemma.

Lemma 1. *If $\eta \le \frac{\gamma \lambda_{min}}{B^2}$ where λ_{min} is the minimum eigenvalue of $\mathbb{E}_{a\sim q}[aa^\top]$, then $\eta|a^\top \widehat{\ell}_t| \le 1$.*

Proof. Let $M_t = \sum_{i=1}^d \lambda_i v_i v_i^\top$ be the eigendecomposition of M_t so that $\lambda_1 \le \ldots, \lambda_d$. Note that because $M_t = (1-\gamma)\mathbb{E}_{a\sim p'_t}[aa^\top] + \gamma \mathbb{E}_{a\sim q}[aa^\top]$, its smallest eigenvalue λ_1 is lower bounded by $\gamma\lambda_{\min}$. Therefore, we have

$$|a^\top \widehat{\ell}_t| = |a^\top M_t^{-1} a_t||a_t^\top \ell_t| \le |a^\top M_t^{-1} a_t| = \left|\sum_{i=1}^d \frac{1}{\lambda_i}(a^\top v_i)(a_t^\top v_i)\right| \le \frac{B^2}{\lambda_1} \le \frac{B^2}{\gamma\lambda_{\min}},$$

where the first inequality is by the assumption $|a_t^\top \ell_t| \le 1$ and the second inequality is by Cauchy-Schwarz inequality

$$\left|\sum_{i=1}^d \frac{1}{\lambda_i}(a^\top v_i)(a_t^\top v_i)\right| \le \frac{1}{\lambda_1}\sum_{i=1}^d \left|(a^\top v_i)(a_t^\top v_i)\right| \le \frac{1}{\lambda_1}\sqrt{\left(\sum_{i=1}^d (a^\top v_i)^2\right)\left(\sum_{i=1}^d (a_t^\top v_i)^2\right)}$$

$$= \frac{1}{\lambda_1}\sqrt{\left(a^\top\left(\sum_{i=1}^d v_i v_i^\top\right)a\right)\left(a_t^\top\left(\sum_{i=1}^d v_i v_i^\top\right)a_t\right)} = \frac{1}{\lambda_1}\|a\|_2\|a_t\|_2 \le \frac{B^2}{\lambda_1}.$$

This finishes the proof. □

Roughly speaking, the lemma above implies that it is desirable to pick q such that it explores every direction with reasonable probability. The resulting algorithm is called by many names in different works, such as Exp2 (Expanded Exp) or GeometricHedge [Dani et al., 2008, Cesa-Bianchi and Lugosi, 2012, Bubeck et al., 2012]. We are now ready to prove the following regret bound.

Theorem 1. *If $\eta \leq \frac{\gamma\lambda_{min}}{B^2}$, then Exp2 ensures*

$$\mathbb{E}\left[\sum_{t=1}^{T} a_t^\top \ell_t\right] - \min_{a\in A}\sum_{t=1}^{T} a^\top \ell_t \leq \frac{\ln K}{\eta} + 2\gamma T + \eta T d.$$

Setting $\eta = \sqrt{\frac{\ln K}{\left(\frac{2B^2}{\lambda_{min}}+d\right)T}}$ and $\gamma = \frac{B^2\eta}{\lambda_{min}}$ leads to a regret of order $\mathcal{O}\left(\sqrt{\left(\frac{2B^2}{\lambda_{min}}+d\right)T\ln K}\right)$.

Proof. By Lemma 1 and the analysis of Hedge, we have for any $a_\star \in A$,

$$\sum_{t=1}^{T}\sum_{a\in A} p'_t(a)(a^\top \widehat{\ell}_t) - \sum_{t=1}^{T} a_\star^\top \widehat{\ell}_t \leq \frac{\ln K}{\eta} + \eta\sum_{t=1}^{T}\sum_{a\in A} p'_t(a)(a^\top \widehat{\ell}_t)^2.$$

Plugging $p'_t(a) = \frac{p_t(a)-\gamma q(a)}{1-\gamma}$, multiplying both sides by $1-\gamma$, and rearranging give

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{a\in A} p_t(a)(a^\top \widehat{\ell}_t) - \sum_{t=1}^{T} a_\star^\top \widehat{\ell}_t \\
&\leq \frac{(1-\gamma)\ln K}{\eta} + \gamma\sum_{t=1}^{T}\sum_{a\in A} q(a)(a^\top \widehat{\ell}_t) - \gamma\sum_{t=1}^{T} a_\star^\top \widehat{\ell}_t + \eta\sum_{t=1}^{T}\sum_{a\in A}(p_t(a)-\gamma q(a))(a^\top \widehat{\ell}_t)^2 \\
&\leq \frac{\ln K}{\eta} + \gamma\sum_{t=1}^{T}\sum_{a\in A} q(a)(a^\top \widehat{\ell}_t) - \gamma\sum_{t=1}^{T} a_\star^\top \widehat{\ell}_t + \eta\sum_{t=1}^{T}\sum_{a\in A} p_t(a)(a^\top \widehat{\ell}_t)^2.
\end{aligned}$$

By the unbiasedness of $\widehat{\ell}_t$, taking expectation on both sides and using $|a^\top \ell_t| \leq 1$ lead to

$$\mathbb{E}\left[\sum_{t=1}^{T} a_t^\top \ell_t\right] - \min_{a\in A}\sum_{t=1}^{T} a^\top \ell_t \leq \frac{\ln K}{\eta} + 2\gamma T + \eta\sum_{t=1}^{T}\sum_{a\in A}\mathbb{E}\left[p_t(a)(a^\top \widehat{\ell}_t)^2\right].$$

To bound the last term, note that

$$\begin{aligned}
&\mathbb{E}_t\left[p_t(a)(a^\top \widehat{\ell}_t)^2\right] = p_t(a)\mathbb{E}_t\left[(a_t^\top \ell_t)^2 a^\top M_t^{-1} a_t a_t^\top M_t^{-1} a\right] \\
&\leq p_t(a) a^\top M_t^{-1}\mathbb{E}_t\left[a_t a_t^\top\right] M_t^{-1} a = p_t(a) a^\top M_t^{-1} a = \mathrm{TR}(M_t^{-1}(p_t(a) a a^\top))
\end{aligned}$$

and thus

$$\sum_{a\in A}\mathbb{E}_t\left[p_t(a)(a^\top \widehat{\ell}_t)^2\right] \leq \mathrm{TR}(M_t^{-1} M_t) = d,$$

which completes the proof. □

Note that the sum of the eigenvalues of $\mathbb{E}_{a\sim q}[aa^\top]$ is bounded by B^2 (since $\mathrm{TR}(\mathbb{E}_{a\sim q}[aa^\top]) = \mathbb{E}_{a\sim q}[\mathrm{TR}(aa^\top)]$). Therefore, if the eigenvalues of $\mathbb{E}_{a\sim q}[aa^\top]$ are all close to each other, we have $\lambda_{\min} = \Omega(B^2/d)$, which then leads to a regret bound of $\mathcal{O}(\sqrt{dT\ln K})$, proven to be optimal in [Dani et al., 2008] (note that the parameter B in fact does not play a role in the optimal regret). This again suggests that q should uniformly explore different directions in $\mathbb{R}^d$.

In general, it turns out that one can always find a q over a subset of A with special geometric properties such that $\lambda_{\min} = \Omega(B^2/d)$ [Bubeck et al., 2012]. However, for many examples (such as those we discuss in the next section), simply setting q to be a uniform distribution over A is enough.

2 Examples

The most important example of linear bandit is the combinatorial bandit problem, where $A \subset \{0,1\}^d$ represents a set of combinatorial concepts chosen from d basic elements. Examples include spanning trees, paths, cuts, Hamiltonian cycles, permutations, and many more. Below we will apply the general results to two of these problems by computing $\lambda_{\min}$ in each case. In both examples

q is uniform over A and we will drop the subscript $a \sim q$ in the expectation for conciseness. We will use the fact $\lambda_{\min} = \min_{\|v\|_2=1} v^\top \mathbb{E}\left[aa^\top\right] v = \min_{\|v\|_2=1} \mathbb{E}\left[(a^\top v)^2\right]$ and

$$\mathbb{E}\left[(a^\top v)^2\right] = \mathbb{E}\left[\left(\sum_{i=1}^d a(i)v(i)\right)^2\right] = \mathbb{E}\left[\sum_{i=1}^d a(i)^2 v(i)^2 + \sum_{i\neq j} a(i)a(j)v(i)v(j)\right]$$
$$= \sum_{i=1}^d \Pr(a(i)=1)v(i)^2 + \sum_{i\neq j} \Pr(a(i)=a(j)=1)v(i)v(j).$$

Note that in general the time complexity of the algorithm is $\mathcal{O}(K)$ per round, which is often prohibitively large. However, in some cases one can actually implement the algorithm much more efficiently using techniques such as dynamic programming.

Hypercube. The first example is when A is the entire hypercube $\{0,1\}^d$. This corresponds to a setting where there are d items and each time we can pick any subset of them and observe the sum of the losses of the selected items. Now note that $\Pr(a(i)=1) = 1/2$ and $\Pr(a(i)=a(j)=1) = 1/4$, we have for any v with $\|v\|_2 = 1$,

$$\mathbb{E}\left[(a^\top v)^2\right] = \frac{1}{2}\|v\|_2^2 + \frac{1}{4}\sum_{i\neq j} v(i)v(j) = \frac{1}{4}\|v\|_2^2 + \frac{1}{4}\left(\sum_{i=1}^d v(i)\right)^2 \geq \frac{1}{4}.$$

The minimum is achievable as long as $\sum_{i=1}^d v(i) = 0$, which means $\lambda_{\min} = 1/4$. Together with $B = \sqrt{d}$ and $K = 2^d$, this implies a regret of order $\mathcal{O}(d\sqrt{T})$ for Exp2.

m-sets. The next example is when $A = \{a \in \{0,1\}^d : \|a\|_1 = m\}$, that is, each time we can only pick exactly m items. Similarly, noting that $\Pr(a(i)=1) = \binom{d-1}{m-1}/\binom{d}{m}$ and $\Pr(a(i)=a(j)=1) = \binom{d-2}{m-2}/\binom{d}{m}$, we have for any v with $\|v\|_2 = 1$

$$\mathbb{E}\left[(a^\top v)^2\right] = \frac{\binom{d-1}{m-1}}{\binom{d}{m}}\|v\|_2^2 + \frac{\binom{d-2}{m-2}}{\binom{d}{m}}\sum_{i\neq j} v(i)v(j)$$
$$= \left(\frac{m}{d} - \frac{m(m-1)}{d(d-1)}\right)\|v\|_2^2 + \frac{m(m-1)}{d(d-1)}\left(\sum_{i=1}^d v(i)\right)^2 \geq \frac{m(d-m)}{d(d-1)}.$$

Together with $K = \binom{d}{m}$ and $B = \sqrt{m}$, as long as $m = o(d)$ the regret of Exp2 is $\mathcal{O}(\sqrt{dmT\ln\frac{d}{m}})$.

References

Sébastien Bubeck, Nicolò Cesa-Bianchi, and Sham Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *25th Annual Conference on Learning Theory*, 2012.

Nicolò Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

Varsha Dani, Sham M Kakade, and Thomas P Hayes. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems 21*, 2008.

Lecture 17

Instructor: Haipeng Luo

1 Adversarial Linear Bandit and FTRL

In the last lecture we discussed the Exp2 algorithm for adversarial linear bandit. The problem of Exp2 is that in general it does not admit an efficient algorithm (in some cases even finding the right exploration distribution is computationally expensive). This time we discuss a different and efficient approach from the seminal work [Abernethy et al., 2008] that borrows a deep and beautiful idea from convex optimization.

Recall the linear bandit problem under an adversarial environment: for each $t = 1, \ldots, T$,

1. learner picks action $w_t \in \Omega \subset \mathbb{R}^d$ while simultaneously environment picks $\ell_t \in \mathbb{R}^d$;
2. learner suffers and observes $w_t^\top \ell_t$ (assume $|w^\top \ell_t| \leq 1$ for any $w \in \Omega$).

Note that we switch to the notation w for an action and Ω for the set of actions to highlight its connection to the OCO setting and the fact that Ω is a compact convex set (instead of a discrete set as for Exp2). Once again we assume that the environment is oblivious and aim to minimize expected regret:

$$\mathbb{E}[\mathcal{R}_T] = \mathbb{E}\left[\sum_{t=1}^{T} w_t^\top \ell_t\right] - \min_{w \in \Omega} \sum_{t=1}^{T} w^\top \ell_t.$$

In the full information setting we have seen the OGD algorithm, an instance of FTRL, that works for general OCO problems. Here we will again consider using FTRL with some regularizer ψ and naturally feed it with some loss estimators $\widehat{\ell}_t$:

$$w_{t+1} = \operatorname*{argmin}_{w \in \Omega} \sum_{\tau=1}^{t} w^\top \widehat{\ell}_\tau + \frac{1}{\eta}\psi(w).$$

There are two main difficulties in this approach that are closely related to each other. The first one is about how to construct the loss estimators. If one looks at the way we construct estimators for Exp3 or Exp2, it is clear that randomization is the key. Therefore, a natural idea is to explore randomly around w_t (computed according to FTRL), instead of exactly playing w_t. One possibility is to simply explore a small ball centered at w_t. This is a reasonable strategy as we will see in the next lecture, but not an optimal one. The reason is that if w_t is close to the boundary of Ω in one direction, then the exploration ball needs to be very small, which limits the exploration in all other directions.

Another difficulty is in choosing the regularizer. In previous examples of FTRL, we have used some regularizer that is strongly convex in some norm $\|\cdot\|$ and have shown that the regret depends on the dual norm of the gradients of the loss functions, which in this case is $\|\widehat{\ell}_t\|_\star$. Since $\widehat{\ell}_t$ in general can have large coordinates (just think about the importance weighted estimators), it is important that the dual norm somehow magically cancels these large coordinates. Indeed, we have seen similar phenomenon in the analysis of Exp3 and Exp2.

It turns out that both difficulties can be simultaneously addressed using one special kind of regularizer, call *self-concordant barriers*, which is also the key concept behind the classic optimization

Algorithm 1: SCRiBLe

Input: learning rate $\eta > 0$ and a ν-self-concordant function ψ
for $t = 1, \dots, T$ **do**
  compute $w_t = \operatorname{argmin}_{w \in \Omega} \sum_{\tau=1}^{t-1} w^\top \widehat{\ell}_\tau + \frac{1}{\eta}\psi(w)$
  compute eigendecomposition $\nabla^2 \psi(w_t) = \sum_{i=1}^d \lambda_i v_i v_i^\top$
  sample $i_t \in [d]$ and $\sigma_t \in \{-1, +1\}$ uniformly at random
  play $\widetilde{w}_t = w_t + \frac{\sigma_t}{\sqrt{\lambda_{i_t}}} v_{i_t}$ and observe $\widetilde{w}_t^\top \ell_t$
  construct estimator $\widehat{\ell}_t = d(\widetilde{w}_t^\top \ell_t)\sigma_t \sqrt{\lambda_{i_t}} v_{i_t}$

algorithm *interior point method*. Instead of stating the definition of self-concordant barriers immediately, which might not be the most intuitive way to understand why it is helpful here, we will defer its definition to the last section and first state on-the-fly some useful properties of self-concordant barriers as we explain and analyze the algorithm.

The first property is about how the Hessian of a self-concordant barrier stretches the space. Specifically, for a point $w \in \text{int}(\Omega)$ ($\text{int}(\Omega)$ denotes the interior of Ω), we define a norm associated with the Hessian of ψ at w as $\|x\|_w = \|x\|_{\nabla^2\psi(w)} = \sqrt{x^\top \nabla^2 \psi(w) x}$ for any $x \in \mathbb{R}^d$. This is indeed a norm since a self-concordant barrier is strictly convex such that $\nabla^2\psi(w)$ is positive definite for any $w \in \text{int}(\Omega)$. The *Dikin ellipsoid* centered at w with radius r is then defined as the ellipsoid $\mathcal{E}_r(w) = \{x \in \mathbb{R}^d : \|x - w\|_w \le r\}$.

Property 1. *If ψ is a self-concordant barrier on Ω, then $\mathcal{E}_1(w) \subset \Omega$ for any $w \in$ int(Ω).*

In other words, the Hessian of a self-concordant barrier stretches the space in a way so that the unit Dikin ellipsoid is always contained in the action set. This implies that given w_t, we can safely explore within the Dikin ellipsoid $\mathcal{E}_1(w_t)$, and it has the hope of better making use the available space than simply using a ball.

Specifically, we will simply uniformly sample one of the end points of the principal axes of the ellipsoid and play this point. In other words, if $\sum_{i=1}^d \lambda_i v_i v_i^\top$ is the eigendecomposition of $\nabla^2\psi(w_t)$, we will then play $\widetilde{w}_t = w_t + \frac{\sigma_t}{\sqrt{\lambda_{i_t}}} v_{i_t}$ where $\sigma_t \in \{-1, +1\}$ is a uniformly random sign and $i_t \in [d]$ is also chosen uniformly at random. It is clear that in expectation we are playing the point w_t, that is $\mathbb{E}_t[\widetilde{w}_t] = w_t$ ($\mathbb{E}_t$ denotes the conditional expectation with respect to the random draw of i_t and σ_t).

With this sampling scheme, we can construct the loss estimator as $\widehat{\ell}_t = d(\widetilde{w}_t^\top \ell_t)\sigma_t \sqrt{\lambda_{i_t}} v_{i_t}$ so that it lies in the direction of the chosen principal axis. This is indeed an unbiased loss estimator since

$$\mathbb{E}_t\left[\widehat{\ell}_t\right] = \frac{1}{d}\sum_{i=1}^d \left(d\sqrt{\lambda_i} v_i \cdot \frac{1}{2}\left(\sum_{\sigma \in \{-1,+1\}} (w_t^\top \ell_t)\sigma + \frac{\sigma^2 v_i^\top \ell_t}{\sqrt{\lambda_i}} \right) \right) = \left(\sum_{i=1}^d v_i v_i^\top \right) \ell_t = \ell_t.$$

This completes all the details of the algorithm (see Algorithm 1), which is called SCRiBLe (Self-Concordant Regularization in Bandit Learning).

2 Regret Analysis

In this section we prove a regret bound for SCRiBLe. The first step is to simply invoke the BTL lemma: for any $u \in \Omega$,

$$\sum_{t=1}^T (w_t - u)^\top \widehat{\ell}_t \le \frac{\psi(u) - \psi(w_1)}{\eta} + \sum_{t=1}^T (w_t - w_{t+1})^\top \widehat{\ell}_t. \tag{1}$$

The rest of the proof will (slightly) deviate from the proof that we have seen since ψ is not necessarily strongly convex. To deal with last term, we apply Hölder's inequality to get $(w_t - w_{t+1})^\top \widehat{\ell}_t \le \|w_t - w_{t+1}\|_{w_t} \|\widehat{\ell}_t\|^\star_{w_t}$ where $\|x\|^\star_{w_t} = \sqrt{x^\top [\nabla^2\psi(w_t)]^{-1} x}$. The term $\|\widehat{\ell}_t\|^\star_{w_t}$ is exactly the dual

norm term mentioned previously. One can verify that the way we construct $\widehat{\ell}_t$ and the dual norm once again "work well" together, leading to important cancellation:

$$\|\widehat{\ell}_t\|_{w_t}^\star = d|\widetilde{w}_t^\top \ell_t|\sqrt{\lambda_{i_t}}\sqrt{v_{i_t}^\top [\nabla^2\psi(w_t)]^{-1} v_{i_t}} = d|\widetilde{w}_t^\top \ell_t|\sqrt{\lambda_{i_t}}\sqrt{\frac{1}{\lambda_{i_t}} v_{i_t}^\top v_{i_t}} \leq d.$$

Next we will show that the algorithm is stable in the sense that $\|w_t - w_{t+1}\|_{w_t} \leq 8\eta\|\widehat{\ell}_t\|_{w_t}^\star$ so that the second term of Eq. (1) is simply bounded by $8\eta T d^2$. To this end let $F_t(w) = \sum_{\tau=1}^{t-1} w^\top \widehat{\ell}_\tau + \frac{1}{\eta}\psi(w)$ so that $w_t = \operatorname{argmin}_w F_t(w)$. Then we have one one hand by optimality of w_t,

$$\begin{aligned} F_{t+1}(w_t) - F_{t+1}(w_{t+1}) &= (w_t - w_{t+1})^\top \widehat{\ell}_t + F_t(w_t) - F_t(w_{t+1}) \\ &\leq (w_t - w_{t+1})^\top \widehat{\ell}_t \leq \|w_t - w_{t+1}\|_{w_t} \|\widehat{\ell}_t\|_{w_t}^\star, \end{aligned} \tag{2}$$

and on the other hand by Taylor's theorem there exists an point ξ on the segment connecting w_t and w_{t+1} such that

$$\begin{aligned} F_{t+1}(w_t) - F_{t+1}(w_{t+1}) &= \nabla F_{t+1}(w_{t+1})^\top (w_t - w_{t+1}) + \tfrac{1}{2}(w_t - w_{t+1})^\top \nabla^2 F_{t+1}(\xi)(w_t - w_{t+1}) \\ &\geq \tfrac{1}{2}(w_t - w_{t+1})\nabla^2 F_{t+1}(\xi)(w_t - w_{t+1}) \\ &\qquad \text{(by first order optimality condition)} \\ &= \tfrac{1}{2\eta}\|w_t - w_{t+1}\|_\xi^2. \end{aligned} \tag{3}$$

If ψ was strongly convex we would have a direct lower bound for the last term. For self-concordant barriers, we need to use another property, which says that within the unit Dikin ellipsoid, the Hessian of every point is pretty close.

Property 2. *If ψ is a self-concordant barrier on Ω, then $\|h\|_{w'} \geq \|h\|_w (1 - \|w - w'\|_w)$ for any $w \in$ int(Ω), $w' \in \mathcal{E}_1(w)$ and $h \in \mathbb{R}^d$.*

Therefore, if we can first show a weaker stability result: $\|w_t - w_{t+1}\|_{w_t} \leq 1/2$ (which also implies $\|w_t - \xi\|_{w_t} \leq 1/2$), then we can use this property to lower bound $\|w_t - w_{t+1}\|_\xi$ by $\|w_t - w_{t+1}\|_{w_t}(1 - \|w_t - \xi\|_{w_t}) \geq \frac{1}{2}\|w_t - w_{t+1}\|_{w_t}$. Combining with Eq. (2) and Eq. (3) would then finish the proof for $\|w_t - w_{t+1}\|_{w_t} \leq 8\eta\|\widehat{\ell}_t\|_{w_t}^\star$.

We now show that $\|w_t - w_{t+1}\|_{w_t} \leq 1/2$ is indeed true. Since w_{t+1} is the minimizer of the convex function F_{t+1}, it suffices to show that $F_{t+1}(w') \geq F_{t+1}(w_t)$ for all w' on the boundary of $\mathcal{E}_{1/2}(w_t)$, that is, $\|h\|_{w_t} = 1/2$ for $h = w' - w_t$. Indeed, using Taylor's theorem again, we have for some ξ lying on the segment connecting w_t and w',

$$\begin{aligned} F_{t+1}(w') &= F_{t+1}(w_t) + \nabla F_{t+1}(w_t)^\top h + \frac{1}{2} h^\top \nabla^2 F_{t+1}(\xi) h \\ &= F_{t+1}(w_t) + \widehat{\ell}_t^\top h + \nabla F_t(w_t)^\top h + \frac{1}{2\eta}\|h\|_\xi^2 \\ &\geq F_{t+1}(w_t) + \widehat{\ell}_t^\top h + \frac{1}{2\eta}\|h\|_{w_t}^2 (1 - \|w_t - \xi\|_{w_t})^2 \\ &\qquad \text{(by first order optimality condition and Property 2)} \\ &\geq F_{t+1}(w_t) - |\widehat{\ell}_t^\top h| + \frac{1}{32\eta} \\ &\geq F_{t+1}(w_t) - \|\widehat{\ell}_t\|_{w_t}^\star \|h\|_{w_t} + \frac{1}{32\eta} \geq F_{t+1}(w_t) - \frac{d}{2} + \frac{1}{32\eta}, \end{aligned}$$

and therefore as long as $\eta \leq \frac{1}{16d}$ we have $F_{t+1}(w') \geq F_{t+1}(w_t)$ and thus $w_{t+1} \in \mathcal{E}_{1/2}(w_t)$.

Finally, it remains the bound the first term of Eq. (1). While the most natural choice of u is simply the best fixed point in hindsight $w_\star = \operatorname{argmin}_{w\in\Omega} \sum_{t=1}^T w^\top \ell_t$, $\psi(u)$ in this case will actually be infinity (since $w_\star$ is on the boundary and a barrier is unbounded on the boundary as we will discuss soon). Fortunately, if ψ is a ν-self-concordant barrier for some parameter $\nu > 0$ (defined in the next section), then the following property holds:

Property 3. *If ψ is a ν-self-concordant barrier on Ω, then for any $\epsilon > 0$ and $u \in \Omega$ such that $u + \epsilon(u - w_1) \in \Omega$ (where $w_1 = \operatorname{argmin}_{w \in \Omega} \psi(w)$), we have $\psi(u) - \psi(w_1) \leq \nu \ln\left(\frac{1}{\epsilon} + 1\right)$.*

Therefore we can set $u = \frac{1}{1+\epsilon}(w_\star - w_1) + w_1$ so that $u + \epsilon(u - w_1) = w_\star$ and $\psi(u) - \psi(w_1) \leq \nu \ln\left(\frac{1}{\epsilon} + 1\right)$. Note that u is not so far away from $w_\star$ and

$$\sum_{t=1}^{T} (u - w_\star)^\top \ell_t \leq \frac{\epsilon}{1+\epsilon} \sum_{t=1}^{T} (w_1 - w_\star)^\top \ell_t \leq 2T\epsilon.$$

Finally noting that $\mathbb{E}_t[\widetilde{w}_t^\top \ell_t] = w_t^\top \ell_t = \mathbb{E}_t[w_t^\top \widehat{\ell}_t]$ and combining everything we have

$$\begin{aligned}\mathbb{E}[\mathcal{R}_T] &= \mathbb{E}\left[\sum_{t=1}^{T} \widetilde{w}_t^\top \ell_t\right] - \sum_{t=1}^{T} w_\star^\top \ell_t = 2T\epsilon + \mathbb{E}\left[\sum_{t=1}^{T} (w_t - u)^\top \widehat{\ell}_t\right] \\ &\leq 2T\epsilon + \frac{\nu}{\eta} \ln\left(\frac{1}{\epsilon} + 1\right) + 8\eta T d^2.\end{aligned}$$

Picking $\epsilon = 1/T$ and the optimal η we have thus proved the following theorem:

Theorem 1. *With $\eta = \min\left\{\frac{1}{16d}, \sqrt{\frac{\nu \ln T}{T d^2}}\right\}$ SCRiBLe ensures $\mathbb{E}[\mathcal{R}_T] = \mathcal{O}(d\sqrt{\nu T \ln T} + d\nu \ln T)$.*

3 Definition and Examples of Self-concordant Barriers

For completeness we now give the formal definition of ν-self-concordant barriers. A function $\psi : \Omega \to \mathbb{R}$ is a barrier if $\psi(w) \to +\infty$ when w approaches the boundary of Ω. A function ψ is self-concordant if it is C^3 (that is, third-order differentiable) and strictly convex and satisfies the following Lipschitz Hessian condition

$$|\nabla^3 \psi(w)[h, h, h]| \leq 2 \|h\|_w^3 \quad \text{for all } w \in \text{int}(\Omega) \text{ and } h \in \mathbb{R}^d$$

where $\nabla^3 \psi(w)[h, h, h]$ is a shorthand for $\sum_{i,j,k \in [d]} \frac{\partial^3 \psi(w)}{\partial w_i \partial w_j \partial w_k} h_i h_j h_k$. Finally a function ψ is a ν-self-concordant barrier if it is a self-concordant barrier and also satisfies the Lipschitz condition

$$|\nabla \psi(w)^\top h| \leq \sqrt{\nu} \|h\|_w \quad \text{for all } w \in \text{int}(\Omega) \text{ and } h \in \mathbb{R}^d.$$

Based on these definitions, one can prove all the three properties we mentioned above.

A seminal result of [Nesterov and Nemirovskii, 1994] states that there *always* exists a ν-self-concordant barrier with $\nu = O(d)$ for a closed convex set $\Omega \subset \mathbb{R}^d$. Canonical examples include the following:

- $\psi(w) = -\sum_{i=1}^{d} \ln w_i$ is a d-self-concordant barrier for $\Omega = \mathbb{R}^d_+$ (verify yourself);
- $\psi(w) = -\sum_{j=1}^{m} \ln(a_j^\top w - b_j)$ is an m-self-concordant barrier for the polytope $\Omega = \{w \in \mathbb{R}^d : a_j^\top w \geq b_j \text{ for } j = 1, \ldots, m\}$;
- $\psi(w) = -\ln(1 - \|w\|_2^2)$ is a 1-self-concordant barrier for the unit ball $\Omega = \{w \in \mathbb{R}^d : \|w\|_2 \leq 1\}$.

The existence of an $O(d)$-self-concordant barrier implies that the regret of SCRiBLe can be as small as $\mathcal{O}(d^{3/2}\sqrt{T \ln T})$ for any Ω (note that the minimax regret for this problem is actually $\mathcal{O}(d\sqrt{T})$). To efficiently implement SCRiBLe, it is not hard to see that one only needs to be able to compute the Hessian of ψ efficiently (see [Abernethy et al., 2008] for details). The best example to showcase the power of SCRiBLe is the online-shortest-path problem where Ω is simply a polytope (a set of flows) so one can use the above concrete barrier and obtain a very efficient algorithm, while Exp2 is very difficult to implement efficiently (indeed in this case uniform exploration does not work provably).

References

Jacob D Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *21st Annual Conference on Learning Theory*, 2008.

Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

Lecture 18

Instructor: Haipeng Luo

1 Bandit Convex Optimization

In this lecture we discuss the most general bandit problem: bandit convex optimization (BCO), which is basically the OCO setting with only bandit feedback. Specifically, for each $t = 1, \ldots, T$,

1. learner picks action $w_t \in \Omega \subset \mathbb{R}^d$ while simultaneously environment picks a convex loss function $f_t : \Omega \to [-1, 1]$;
2. learner suffers and observes $f_t(w_t)$.

Once again we assume that the environment is oblivious and aim to minimize expected regret:

$$\mathbb{E}[\mathcal{R}_T] = \mathbb{E}\left[\sum_{t=1}^{T} f_t(w_t)\right] - \sum_{t=1}^{T} f_t(w_\star).$$

where $w_\star = \operatorname{argmin}_{w\in\Omega} \sum_{t=1}^{T} f_t(w)$.

Recall that in the full information setting, there is no extra difficulty when f_t is convex compared to the case when f_t is linear, due to the so-called convexity trick: $f_t(w_t) - f_t(w_\star) \leq \nabla f_t(w_t)^\top (w_t - w_\star)$. In the bandit setting, however, this is no longer true because the only feedback is $f_t(w_t)$ while to apply a linear bandit algorithm one needs to observe $\nabla f_t(w_t)^\top w_t$. In fact, this is a very challenging problem and there are still many open problems unsolved.

The convexity trick above can still be helpful though. By now it is clear that one key technique to solve bandit problems is to come up with estimators. If we try to solve the problem directly, it appears that we need to construct an estimator for the function f_t given its value at only one point, which is very challenging. However, by the convexity trick it is clear that one only needs to construct an estimator $\widehat{g}_t$ for the gradient $\nabla f_t(w_t)$, which intuitively is much more manageable. Given such estimators, we can again execute FTRL

$$w_t = \operatorname*{argmin}_{w\in\Omega} \sum_{\tau=1}^{t-1} w^\top \widehat{g}_\tau + \frac{1}{\eta}\psi(w)$$

for some regularizer ψ and learning rate η. To construct these estimators, we need to make use of the following lemma:

Lemma 1. *Given a function f, an invertible matrix M and $\delta > 0$, define the smoothed version of f as $\widehat{f}(w) = \mathbb{E}_{b\sim\mathbb{B}^d}[f(w + \delta M b)]$ where b is a uniform sample of the d-dimensional unit ball $\mathbb{B}^d = \{b \in \mathbb{R}^d : \|b\|_2 \leq 1\}$. Then the following holds*

$$\nabla \widehat{f}(w) = \mathbb{E}_{s\sim\mathbb{S}^d}\left[\frac{d}{\delta} f(w + \delta M s) M^{-1} s\right] \tag{1}$$

where s is a uniform sample of the d-dimensional unit sphere $\mathbb{S}^d = \{s \in \mathbb{R}^d : \|s\|_2 = 1\}$.

We omit the proof here but one can simply verify this fact when $d = 1$ so that the unit ball is simply the segment $[-1, 1]$ and the unit sphere is simply two points -1 and 1. Indeed in this case, with F

being the antiderivative of f we have

$$\nabla \mathbb{E}_{b\sim\mathbb{B}^d}[f(w+\delta Mb)] = \frac{1}{2}\frac{d}{dw}\int_{-1}^{1} f(w+\delta Mb)db = \frac{1}{2\delta M}\frac{d}{dw}\left(F(w+\delta M) - F(w-\delta M)\right)$$
$$= \frac{1}{2\delta M}\left(f(w+\delta M) - f(w-\delta M)\right) = \mathbb{E}_{s\sim\mathbb{S}^d}\left[\frac{d}{\delta}f(w+\delta Ms)M^{-1}s\right].$$

This lemma directly implies a way to construct the gradient estimator $\widehat{g}_t$: draw a uniform sample s from the unit sphere, query the value of $f_t(w_t+\delta Ms)$ for some matrix M and δ by playing $\widetilde{w}_t = w_t + \delta Ms$, and then use $\widehat{g}_t = \frac{d}{\delta}f(w+\delta Ms)M^{-1}s$ as an unbiased estimator of the gradient of $\widehat{f}_t$.

Importantly, this is an unbiased estimator for the smoothed version of f_t but not f_t itself. This leads to one key difficulty in solving BCO using this approach: bias-variance tradeoff of the estimator, which is controlled by the parameter δ. When δ is close to 0, $\widehat{f}_t$ is very close to f_t but $\widehat{g}_t$ will have very large magnitude and large variance; on the other hand, when δ is large, the variance goes down while $\widehat{f}_t$ becomes very different from f_t. We will see how exactly one should tune δ later in the analysis.

The next step is to decide what M should be. The simplest choice is the identity. From a geometric viewpoint, this amounts to exploring the sphere centered at w_t with radius δ. Extra care needs to be taken to ensure that $\widetilde{w}_t$ is never outside the set Ω. Indeed, one of the first BCO algorithms [Flaxman et al., 2005] uses exactly this exploration scheme, together with $\psi(w) = \frac{1}{2}\|w\|_2^2$ so that FTRL is simply gradient descent.

However, as discussed last time, sampling uniformly in all directions might not be the best idea. We have seen that using a self-concordant barrier ψ together with a Dikin ellipsoid exploration scheme works well for linear bandit. Here we can in fact use the same idea (first proposed in [Saha and Tewari, 2011]). Specifically, at time t let $H_t = \nabla^2\psi(w_t)$ be the Hessian of ψ at w_t. If we let $M = H_t^{-\frac{1}{2}}$, then note that

$$\|\widetilde{w}_t - w_t\|_{w_t} = \delta\sqrt{s^\top H_t^{-\frac{1}{2}} H_t H_t^{-\frac{1}{2}} s} = \delta,$$

which means $\widetilde{w}_t$ is exactly on the surface of the Dikin ellipsoid $\mathcal{E}_\delta(w_t)$. Since $\mathcal{E}_1(w_t)$ is contained in Ω, we can safely choose any $\delta \in (0,1]$. See Algorithm 1 for the complete pseudocode.

2 Regret Analysis

We analyze the regret of Algorithm 1 in this section. Recall that with our choice of M, $\widehat{f}_t(w)$ is defined as $\mathbb{E}_{b\sim\mathbb{B}^d}\left[f_t\left(w+\delta H_t^{-\frac{1}{2}}b\right)\right]$. First note that FTRL guarantees a bound on the quantity $\mathbb{E}\left[\sum_{t=1}^T \widehat{f}_t(w_t) - \widehat{f}_t(u)\right] \le \mathbb{E}\left[\sum_{t=1}^T \nabla\widehat{f}_t(w_t)^\top(w_t-u)\right] = \mathbb{E}\left[\sum_{t=1}^T \widehat{g}_t^\top(w_t-u)\right]$ for any $u \in \Omega$. To connect this quantity to the actual regret $\mathbb{E}[\mathcal{R}_T]$, we decompose the regret into five terms and bound each of them separately:

$$\mathbb{E}[\mathcal{R}_T] = \underbrace{\mathbb{E}\left[\sum_{t=1}^T f_t(\widetilde{w}_t) - \widehat{f}_t(\widetilde{w}_t)\right]}_{A_1} + \underbrace{\mathbb{E}\left[\sum_{t=1}^T \widehat{f}_t(\widetilde{w}_t) - \widehat{f}_t(w_t)\right]}_{A_2}$$
$$+ \underbrace{\mathbb{E}\left[\sum_{t=1}^T \widehat{f}_t(w_t) - \widehat{f}_t(u)\right]}_{A_3} + \underbrace{\mathbb{E}\left[\sum_{t=1}^T \widehat{f}_t(u) - f_t(u)\right]}_{A_4} + \underbrace{\mathbb{E}\left[\sum_{t=1}^T f_t(u) - f_t(w_\star)\right]}_{A_5},$$

First, A_1 is simply non-positive by Jensen's inequality:

$$\widehat{f}_t(\widetilde{w}_t) = \mathbb{E}_{b\sim\mathbb{B}^d}\left[f_t\left(\widetilde{w}_t + \delta H_t^{-\frac{1}{2}}b\right)\right] \ge f_t\left(\widetilde{w}_t + \delta H_t^{-\frac{1}{2}}\mathbb{E}_{b\sim\mathbb{B}^d}[b]\right) = f_t(\widetilde{w}_t)$$

Algorithm 1: Variant of SCRiBLe for BCO

Input: parameter $\delta \in (0, 1]$, learning rate $\eta > 0$, and a ν-self-concordant function ψ
for $t = 1, \ldots, T$ **do**
  compute $w_t = \operatorname{argmin}_{w \in \Omega} \sum_{\tau=1}^{t-1} w^\top \widehat{g}_\tau + \frac{1}{\eta}\psi(w)$
  compute Hessian $H_t = \nabla^2 \psi(w_t)$ and sample $s_t \in \mathbb{S}^d$ uniformly at random
  play $\widetilde{w}_t = w_t + \delta H_t^{-\frac{1}{2}} s_t$ and observe $f_t(\widetilde{w}_t)$
  construct estimator $\widehat{g}_t = \frac{d}{\delta} f_t(\widetilde{w}_t) H_t^{\frac{1}{2}} s_t$

Next we look at term A_3, which is bounded by $\mathbb{E}\left[\sum_{t=1}^T \widehat{g}_t^\top (w_t - u)\right]$ as mentioned. Using the analysis from SCRiBLe, it is clear that as long as $\eta \le \frac{1}{16\|\widehat{g}_t\|_{w_t}^\star}$,

$$A_3 \le \frac{\psi(u) - \psi(w_1)}{\eta} + 8\eta \sum_{t=1}^T \|\widehat{g}_t\|_{w_t}^{\star 2}.$$

Note that the dual norm term is bounded as

$$\|\widehat{g}_t\|_{w_t}^{\star 2} = \frac{d^2}{\delta^2} f_t(\widetilde{w}_t)^2 \left(s_t^T H^{\frac{1}{2}} H^{-1} H^{\frac{1}{2}} s_t\right) \le \frac{d^2}{\delta^2}.$$

Together with the discussions from last lecture which says u should be chosen as $\frac{1}{1+\epsilon} w_\star + \frac{\epsilon}{1+\epsilon} w_1$ for some $\epsilon > 0$ so that $\psi(u) - \psi(w_1)$ is bounded by $\nu \ln\left(\frac{1}{\epsilon} + 1\right)$, we have

$$A_3 \le \frac{\nu \ln\left(\frac{1}{\epsilon} + 1\right)}{\eta} + \frac{8\eta d^2 T}{\delta^2},$$

as long as $\eta \le \frac{\delta}{16d}$. For simplicity we assume that T is large enough so that $\eta = \frac{\delta}{d}\sqrt{\frac{\nu}{T}\ln\left(\frac{1}{\epsilon}+1\right)} \le \frac{\delta}{16d}$ and with this η we have $A_3 = \mathcal{O}\left(\frac{d}{\delta}\sqrt{T\nu \ln\left(\frac{1}{\epsilon}+1\right)}\right)$.

With this specific choice of u, we can also bound the term A_5 using Jensen's inequality:

$$f_t(u) - f_t(w_\star) \le \frac{1}{1+\epsilon} f_t(w_\star) + \frac{\epsilon}{1+\epsilon} f_t(w_1) - f_t(w_\star) \le \epsilon |f_t(w_1) - f_t(w_\star)| \le 2\epsilon,$$

and thus $A_5 \le 2T\epsilon$. We will simply pick $\epsilon = 1/T$ so that A_5 is negligible.

Finally, to bound A_2 and A_4, we will make one additional assumption on f_t (although even without any more assumptions one can still prove some weak bounds as shown in [Flaxman et al., 2005]). We will consider two different choices of the assumption, each of which leads to a different rate in the end.

2.1 Lipschitzness Assumption

Assume f_t's are L-Lipschitz, that is, for all $w, w' \in \Omega$, $|f_t(w) - f_t(w')| \le L\|w - w'\|_2$. Note that this implies that $\widehat{f}_t$ is L-Lipschitz too:

$$|\widehat{f}_t(w) - \widehat{f}_t(w')| = \left|\mathbb{E}_{b \in \mathbb{B}^d}\left[f_t\left(w + \delta H_t^{-\frac{1}{2}} b\right) - f_t\left(w' + \delta H_t^{-\frac{1}{2}} b\right)\right]\right| \le L\|w - w'\|_2.$$

Therefore, we have

$$\widehat{f}_t(\widetilde{w}_t) - \widehat{f}_t(w_t) \le \delta L \left\|H_t^{-\frac{1}{2}} s_t\right\|_2 = \delta L \left\|w_t + H_t^{-\frac{1}{2}} s_t - w_t\right\|_2 \le \delta L \max_{w, w' \in \Omega} \|w - w'\|_2$$

where the last inequality holds since $w_t + H_t^{-\frac{1}{2}} s_t \in \mathcal{E}_1(w_t)$ is indeed in Ω as discussed. Letting $D = \max_{w,w' \in \Omega} \|w - w'\|_2$ denote the diameter of Ω, we have $A_2 \le \delta LDT$.

Similarly, we also have

$$\widehat{f}_t(u) - f_t(u) = \mathbb{E}_{b \sim \mathbb{B}^d}\left[f_t\left(u + \delta H_t^{-\frac{1}{2}} b\right) - f_t(u)\right] \le \delta L \mathbb{E}_{b \sim \mathbb{B}^d}\left[\left\|H_t^{-\frac{1}{2}} b\right\|_2\right] \le \delta LD,$$

and thus A_4 is also bounded by δLDT. Putting everything together we have proven

$$\mathbb{E}[\mathcal{R}_T] = \mathcal{O}\left(\delta LDT + \frac{d}{\delta}\sqrt{T\nu \ln T}\right),$$

where one can clearly see the tradeoff between the bias (the first term) and the variance (the second term) controlled by δ. With the optimal tuning of δ (assuming again that T is large enough so that $\delta \leq 1$) this shows $\mathbb{E}[\mathcal{R}_T] = \widetilde{\mathcal{O}}\left(\sqrt{dLD}\nu^{\frac{1}{4}}T^{\frac{3}{4}}\right)$ where the notation $\widetilde{\mathcal{O}}$ hides the small $\ln T$ terms.

2.2 Smoothness Assumption

Next we forget about the Lipschitzness assumption and make a different assumption that f_t's are β-smooth, that is, for any $w, w' \in \Omega$, $f_t(w) - f_t(w') \leq \nabla f_t(w')^\top (w - w') + \frac{\beta}{2}\|w - w'\|_2^2$, which also means that the gradient of f_t exists and is β-Lipschitz. Similarly, we can show that $\widehat{f}_t$ is β-smooth too:

$$\begin{aligned}
\widehat{f}_t(w) - \widehat{f}_t(w') &= \mathbb{E}_{b\sim\mathbb{B}^d}\left[f_t\left(w + \delta H_t^{-\frac{1}{2}} b\right) - f_t\left(w' + \delta H_t^{-\frac{1}{2}} b\right)\right] \\
&\leq \mathbb{E}_{b\sim\mathbb{B}^d}\left[\nabla f_t\left(w' + \delta H_t^{-\frac{1}{2}} b\right)\right]^\top (w - w') + \frac{\beta}{2}\|w - w'\|_2^2 \\
&= \nabla \mathbb{E}_{b\sim\mathbb{B}^d}\left[f_t\left(w' + \delta H_t^{-\frac{1}{2}} b\right)\right]^\top (w - w') + \frac{\beta}{2}\|w - w'\|_2^2 \\
&= \nabla \widehat{f}_t(w')^\top (w - w') + \frac{\beta}{2}\|w - w'\|_2^2 .
\end{aligned}$$

With this fact, A_2 and A_4 can both be bounded by $\frac{1}{2}\beta\delta^2 D^2 T$ since

$$\mathbb{E}\left[\widehat{f}_t(\widetilde{w}_t) - \widehat{f}_t(w_t)\right] \leq \mathbb{E}\left[\delta \nabla \widehat{f}_t(w_t)^\top H_t^{-\frac{1}{2}} s_t + \frac{\beta\delta^2}{2}\left\|H_t^{-\frac{1}{2}} s_t\right\|_2^2\right] \leq \frac{\beta\delta^2 D^2}{2}$$

and

$$\begin{aligned}
\widehat{f}_t(u) - f_t(u) &= \mathbb{E}_{b\sim\mathbb{B}^d}\left[f_t\left(u + \delta H_t^{-\frac{1}{2}} b\right) - f_t(u)\right] \\
&\leq \mathbb{E}_{b\sim\mathbb{B}^d}\left[\delta \nabla f_t(u)^\top H_t^{-\frac{1}{2}} b + \frac{\beta\delta^2}{2}\left\|H_t^{-\frac{1}{2}} b\right\|_2^2\right] \leq \frac{\beta\delta^2 D^2}{2}.
\end{aligned}$$

Putting everything together we have proven $\mathbb{E}[\mathcal{R}_T] = \mathcal{O}\left(\beta\delta^2 D^2 T + \frac{d}{\delta}\sqrt{T\nu \ln T}\right)$. With the optimal tuning of δ this becomes $\mathbb{E}[\mathcal{R}_T] = \widetilde{\mathcal{O}}\left((\beta\nu)^{\frac{1}{3}}(TdD)^{\frac{2}{3}}\right)$, improving the dependence on T from $T^{\frac{3}{4}}$ to $T^{\frac{2}{3}}$ compared to the Lipschitz case. Also, note that linear functions are smooth with $\beta = 0$. Therefore if f_t's are linear we can simply set $\delta = 1$ and recover the SCRiBLe regret bound $\mathcal{O}\left(d\sqrt{T\nu \ln T}\right)$. The two algorithms are slightly different in terms of sampling scheme though.

It turns out that these are all suboptimal results and the optimal regret for this problem is still $\mathcal{O}(\sqrt{T})$ (ignoring dependence on other parameters). A polynomial time algorithm to achieve this regret was only discovered very recently [Bubeck et al., 2016]. However, the algorithm is very complicated and far from being practical. Obtaining simple and optimal algorithms (even with extra assumptions) is still an important open problem.

References

Sébastien Bubeck, Ronen Eldan, and Yin Tat Lee. Kernel-based methods for bandit convex optimization. *arXiv preprint arXiv:1607.03084*, 2016.

Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth Annual ACM-SIAM symposium on Discrete algorithms*, 2005.

Ankan Saha and Ambuj Tewari. Improved regret guarantees for online smooth convex optimization with bandit feedback. In *The 14th International Conference on Artificial Intelligence and Statistics*, 2011.

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 $t = 1, \ldots, T$,

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 multi-armed 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.

Lecture 20

Instructor: Haipeng Luo

1 Toward Optimal and Efficient Contextual Bandit

We have discussed Epsilon-Greedy last time, an oracle-efficient but suboptimal algorithm for the i.i.d. contextual bandit problem. In this lecture we discuss an optimal but inefficient algorithm. It conveys important ideas based on which one can further derive both optimal and efficient algorithm that we will discuss next time.

Let's first recall what the key difficulty is in getting the optimal regret. A very reasonable template for an algorithm is to come up with a distribution over policies P_t at time t and then pick a_t according to $P_t(\cdot|x_t)$ but with a small amount of uniform exploration. To this end, define $P_t^\mu(\cdot|x_t)$ to be the mixture of $P_t(\cdot|x_t)$ and some uniform exploration so that

$$P_t^\mu(a|x_t) = (1 - K\mu)P_t(a|x_t) + \mu$$

for some parameter $\mu \leq 1/K$. Also recall the notation $\bar{\ell}(\pi) = \mathbb{E}_{(x,\ell)\sim\mathcal{D}}[\ell(\pi(x))]$ for the expected loss of a policy π and $\bar{\ell}_t(\pi) = \frac{1}{t}\sum_{\tau=1}^t \widehat{\ell}_\tau(\pi(x_\tau))$ for the empirical average loss where $\widehat{\ell}_\tau$ is the usual importance weighted estimators. The most important part of analyzing such algorithms is to understand the concentration of $\bar{\ell}_t(\pi)$. As shown before, the conditional variance of $\widehat{\ell}_t(\pi) - \bar{\ell}_t(\pi)$ is bounded as

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

Define for a distribution P and a policy π (and implicitly a marginal distribution over the contexts and a parameter μ)

$$V(P,\pi) = \mathbb{E}_x\left[\frac{1}{P^\mu(\pi(x)|x)}\right],$$

so that the conditional variance is simply bounded by $V(P_t,\pi)$. By Freedman's inequality, we have with probability at least $1-\delta$,

$$|\bar{\ell}_t(\pi) - \bar{\ell}(\pi)| \leq \mathcal{O}\left(\sqrt{\left(\frac{1}{t}\sum_{\tau=1}^t V(P_\tau,\pi)\right)\frac{\ln(1/\delta)}{t}} + \frac{\ln(1/\delta)}{\mu t}\right).$$

For Epsilon-Greedy, we simply bound each $V(P_\tau,\pi)$ by $1/\mu$, which then leads to $O(T^{\frac{2}{3}})$ regret. If we could ensure that $V(P_\tau,\pi)$ is much smaller, say a constant that is independent of T, then there is hope in getting $\mathcal{O}(\sqrt{T})$ regret (this is indeed the case in the full information case).

To get a sense of how small this quantity can be, one can first consider the special case where there is only one possible context x. In this case the policies are naturally grouped into (at most) K classes according to which action they pick given x. Then simply picking a distribution P over these policies such that $P(\cdot|x)$ is uniform would make $V(P,\pi) \leq K$ for any π.

When there are many different contexts, the argument above does not generalize. However, the conclusion turns out to be still true up a factor of two, as shown by the following lemma.

Lemma 1. *For any policy space Π, any context distribution, and any $\mu \leq 1/K$, there always exists a distribution $P \in \Delta(\Pi)$ such that $V(P,\pi) \leq 2K$ for all $\pi \in \Pi$.*

Proof. It is clear that the statement is equivalent to the following:

$$\min_{P\in\Delta(\Pi)}\max_{\pi\in\Pi} V(P,\pi)\le 2K.$$

We thus work on the minimax expression on the left. If $\mu \ge 1/(2K)$, then the statement trivially holds since $V(P,\pi)\le 1/\mu$. Below we assume $\mu \le 1/(2K)$ and thus $1-K\mu \ge 1/2$. First note that we can "linearize" the maximization part because maximization over the simplex can always be achieved by one of the elements:

$$\min_{P\in\Delta(\Pi)}\max_{\pi\in\Pi} V(P,\pi) = \min_{P\in\Delta(\Pi)}\max_{Q\in\Delta(\Pi)} \mathbb{E}_{\pi\sim Q}[V(P,\pi)].$$

Next we apply Sion's minimax theorem to swap the min and max

$$\min_{P\in\Delta(\Pi)}\max_{Q\in\Delta(\Pi)} \mathbb{E}_{\pi\sim Q}[V(P,\pi)] = \max_{Q\in\Delta(\Pi)}\min_{P\in\Delta(\Pi)} \mathbb{E}_{\pi\sim Q}[V(P,\pi)].$$

By picking a specific $P=Q$, the last expression is clearly bounded by $\max_{Q\in\Delta(\Pi)} \mathbb{E}_{\pi\sim Q}[V(Q,\pi)]$. Now note that

$$\begin{aligned}\mathbb{E}_{\pi\sim Q}[V(Q,\pi)] &= \mathbb{E}_x\left[\sum_{\pi\in\Pi}\frac{Q(\pi)}{Q^\mu(\pi(x)|x)}\right] \le \mathbb{E}_x\left[\sum_{\pi\in\Pi}\frac{Q(\pi)}{(1-K\mu)Q(\pi(x)|x)}\right]\\ &\le 2\mathbb{E}_x\left[\sum_{\pi\in\Pi}\frac{Q(\pi)}{Q(\pi(x)|x)}\right] = 2\mathbb{E}_x\left[\sum_{a=1}^K\sum_{\pi:\pi(x)=a}\frac{Q(\pi)}{Q(a|x)}\right] = 2K,\end{aligned}$$

which completes the proof. □

This lemma shows that we can always find a distribution that leads to low variance for the loss estimators. In other words, the distribution does a good job in terms of exploration. However, such a distribution says nothing about exploitation. Indeed, it is even independent of the observed losses.

One way to address this issue is to ensure that we only keep around "good" policies. Specifically, we start with the whole policy space $\Pi_1 = \Pi$; at each time t, we find a distribution P_t over Π_t that induces low variance (Lemma 1 shows that it always exists no matter what Π_t is); finally we remove all bad policies in Π_t based on what we have observed and obtain a new policy space Π_{t+1}. This final step can be done by simply checking how much worse a policy is in terms of the empirical performance compared to the empirically best policy, since we know that empirical data concentrates well to the truth due to the low variance of estimators. This algorithm is called Policy Elimination [Dudík et al., 2011] and is shown in Algorithm 1.

As mentioned this is not an efficient algorithm. Moreover, since the definition of V depends on the the unknown context distribution, it does not even seem to be a valid algorithm. However, the latter issue can be solved by simply replacing the context distribution by the empirical distribution, that is, a uniform distribution over $x_1,\dots,x_t$ at time t. The analysis remains similar except that one more step of concentration is needed now. For simplicity, we will skip this step and assume that the context distribution is known. The following theorem shows that Policy Elimination achieves the optimal regret (recall that regret is defined as $\mathcal{R}_T = \sum_{t=1}^T(\ell_t(a_t) - \bar\ell(\pi^\star))$.

Theorem 1. *Policy Elimination ensures* $\mathcal{R}_T = \widetilde{\mathcal{O}}\left(\sqrt{TK\ln(N/\delta) + K\ln(N/\delta)}\right)$ *with probability at least* $1-\delta$.

Proof. Clearly we have $\Pi_T \subset \cdots \subset \Pi_1 = \Pi$ and thus for any $\pi\in\Pi_t$ we have $V(P_\tau,\pi)\le 2K$ for any $\tau = 1,\dots,t$ by the algorithm. Therefore, by Freedman's inequality and union bound, we have with probability $1-\delta/2$, for all $t\in[T]$ and all $\pi\in\Pi_t$,

$$\begin{aligned}|\bar\ell_t(\pi) - \bar\ell(\pi)| &\le 2\sqrt{\left(\frac{1}{t}\sum_{\tau=1}^t V(P_\tau,\pi)\right)\frac{\ln(4NT/\delta)}{t}} + \frac{\ln(4NT/\delta)}{\mu t}\\ &\le 2\sqrt{\frac{2K\ln(4NT/\delta)}{t}} + \frac{\ln(4NT/\delta)}{\mu t} = \frac{\epsilon_t}{2}. \qquad (1)\end{aligned}$$

Algorithm 1: Policy Elimination

Input: failure probability $\delta \in (0,1)$

Initialization: let $\Pi_1 = \Pi$, $\epsilon_t = 4\sqrt{\frac{2K\ln(4NT/\delta)}{t}} + \frac{2\ln(4NT/\delta)}{\mu t}$, $\mu = \min\left\{\frac{1}{K}, \sqrt{\frac{\ln(TN/\delta)\ln T}{TK}}\right\}$

for $t = 1, \ldots, T$ **do**

find P_t such that $V(P_t, \pi) \leq 2K$ for all $\pi \in \Pi_t$

play $a_t \sim P_t^{\mu}(\cdot|x_t)$

update $\Pi_{t+1} = \left\{\pi \in \Pi_t : \bar{\ell}_t(\pi) \leq \bar{\ell}_t(\pi_t^\star) + \epsilon_t\right\}$ where $\pi_t^\star = \operatorname{argmin}_{\pi\in\Pi_t} \bar{\ell}_t(\pi)$

Conditioning on this event, we can show that $\pi^\star$ is never removed from the policy space: inductively assuming $\pi^\star \in \Pi_t$, we have

$$\bar{\ell}_t(\pi^\star) \leq \bar{\ell}(\pi^\star) + \frac{\epsilon_t}{2} \leq \bar{\ell}(\pi_t^\star) + \frac{\epsilon_t}{2} \leq \bar{\ell}_t(\pi_t^\star) + \epsilon_t,$$

which means $\pi^\star$ will stay in Π_{t+1}. Finally, applying Azuma inequality we have with probability $1-\delta$,

$$\begin{aligned}
\sum_{t=1}^T \ell_t(a_t) &\leq \sum_{t=1}^T \mathbb{E}_{x_t,\ell_t,a_t}[\ell_t(a_t)] + \mathcal{O}\left(\sqrt{T\ln(1/\delta)}\right) \\
&\leq \sum_{t=1}^T \mathbb{E}_{x_t,\ell_t}\left[\sum_{\pi\in\Pi_t} P_t(\pi)\ell_t(\pi(x_t))\right] + TK\mu + \mathcal{O}\left(\sqrt{T\ln(1/\delta)}\right) \\
&= \sum_{t=1}^T \mathbb{E}_{\pi\sim P_t}[\bar{\ell}(\pi)] + TK\mu + \mathcal{O}\left(\sqrt{T\ln(1/\delta)}\right) \\
&\leq \sum_{t=1}^T \mathbb{E}_{\pi\sim P_t}[\bar{\ell}_{t-1}(\pi)] + \frac{1}{2}\sum_{t=2}^T \epsilon_{t-1} + TK\mu + \mathcal{O}\left(\sqrt{T\ln(1/\delta)}\right) && \text{(by Eq. (1))} \\
&\leq \sum_{t=1}^T \bar{\ell}_{t-1}(\pi^\star) + \frac{3}{2}\sum_{t=2}^T \epsilon_{t-1} + TK\mu + \mathcal{O}\left(\sqrt{T\ln(1/\delta)}\right) && \text{(since } \pi \in \Pi_t\text{)} \\
&\leq \sum_{t=1}^T \bar{\ell}(\pi^\star) + 2\sum_{t=2}^T \epsilon_{t-1} + TK\mu + \mathcal{O}\left(\sqrt{T\ln(1/\delta)}\right) && \text{(by Eq. (1))} \\
&\leq \sum_{t=1}^T \bar{\ell}(\pi^\star) + \mathcal{O}\left(\sqrt{TK\ln(TN/\delta)} + \frac{\ln(TN/\delta)\ln T}{\mu} + TK\mu\right),
\end{aligned}$$

which competes the proof with the optimal tuning of μ. □

References

Miroslav Dudík, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2011.

Lecture 21

Instructor: Haipeng Luo

1 Softening Policy Elimination

In this lecture we are finally ready to discuss the state-of-the-art algorithm for the i.i.d. contextual bandit problem, which is both optimal and oracle-efficient [Agarwal et al., 2014]. Recall that the idea of Policy Elimination is to find $P_t \in \Delta(\Pi_t)$ such that $V(P_t, \pi) \leq 2K$ for all $\pi \in \Pi_t$ where

$$V(P, \pi) = \mathbb{E}_x \left[\frac{1}{P^\mu(\pi(x)|x)} \right]$$

is essentially the variance of the loss estimators that we want to control. To obtain an efficient algorithm, we need to forget about the idea of removing policies from Π. So is it possible to ensure $V(P_t, \pi) \leq 2K$ for all $\pi \in \Pi$ while at the same time P_t puts most of the weights on good policies?

Unfortunately this is too strong of a requirement. For example, if there is a bad policy π_{bad} which always picks a bad action a_{bad} with loss 1 and no other policy ever picks a_{bad}, then

$$2K \geq V(P_t, \pi_{\text{bad}}) = \mathbb{E}_x \left[\frac{1}{P_t^\mu(a_{\text{bad}}|x)} \right] = \frac{1}{(1 - K\mu)P_t(\pi_{\text{bad}}) + \mu}$$

which implies that $P_t(\pi_{\text{bad}})$ will be pretty large assuming μ is small. This is clearly not a good algorithm.

From this example, however, we can see that the condition $V(P_t, \pi) \leq 2K$ should be somehow relaxed for bad policies. Just as in Policy Elimination, whether a policy is good or bad can be roughly determined by its empirical performance compared to the empirically best policy. Specifically, recall the notation $\bar{\ell}_t(\pi) = \frac{1}{t} \sum_{\tau=1}^{t} \widehat{\ell}_\tau(\pi(x_\tau))$ for the empirical average loss and $\pi_t^\star = \operatorname{argmin}_{\pi \in \Pi} \bar{\ell}_t(\pi)$ for the empirically best policy up to time t. Define empirical average regret for a policy π to be

$$\text{Reg}_t(\pi) = \bar{\ell}_t(\pi) - \bar{\ell}_t(\pi_t^\star).$$

We now relax the low-variance condition as: find P_t such that

$$V(P_t, \pi) \leq 2K + \beta \text{Reg}_{t-1}(\pi) \quad \forall\, \pi \in \Pi$$

for some parameter $\beta > 0$ to be specified later. Now there is hope to impose exploitation simultaneously. Specifically, we want $\sum_{\pi \in \Pi} P_t(\pi) \text{Reg}_{t-1}(\pi)$ to be as small as possible. How small can it be? The following lemma answers this question.

Lemma 1. *For any $\beta > 0$, there always exists a distribution $P \in \Delta(\Pi)$ such that*

$$\sum_{\pi \in \Pi} P(\pi) \text{Reg}_{t-1}(\pi) \leq \frac{2K}{\beta}$$
$$V(P, \pi) \leq 2K + \beta \text{Reg}_{t-1}(\pi) \quad \forall\, \pi \in \Pi.$$

Proof. Define function $F_t : \Delta(\Pi) \to \mathbb{R}_+$ as

$$F_t(P) = \sum_{\pi \in \Pi} P(\pi) \text{Reg}_{t-1}(\pi) + \frac{2}{\beta} \mathbb{E}_x \left[\sum_{a=1}^{K} \ln \frac{1}{P^\mu(a|x)} \right].$$

The claim is that the minimizer of $F_t(P)$, which always exists due to compactness of $\Delta(\Pi)$ and continuousness of F_t, satisfies both conditions. To see this, first notice that we can extend the function to a set of "sub-distributions" $\Delta(\Pi)' = \{P \in \mathbb{R}_+^N : \sum_{\pi\in\Pi} P(\pi) \leq 1\}$ and still have

$$\min_{P\in\Delta(\Pi)} F_t(P) = \min_{P\in\Delta(\Pi)'} F_t(P).$$

This is because for any sub-distribution $P \in \Delta(\Pi)'$, one can make it a distribution by increasing the weight for policy $\pi^\star_{t-1}$ until the weights sum up to 1. This will only decrease the function value since $\text{Reg}_{t-1}(\pi^\star_{t-1}) = 0$ and the second term of F_t is decreasing in any coordinate of P.

Next note that the derivate of F_t with respect to a policy π is

$$\nabla F_t(P)(\pi) = \text{Reg}_{t-1}(\pi) - \frac{2(1-K\mu)}{\beta} V(P,\pi).$$

Let P^* be a minimizer of F_t over $\Delta(\Pi)'$. By KKT conditions, we have

$$\text{Reg}_{t-1}(\pi) - \frac{2(1-K\mu)}{\beta} V(P^*,\pi) - \lambda_\pi + \lambda = 0 \tag{1}$$

for some Lagrangian multipliers $\lambda_\pi \geq 0$ and $\lambda \geq 0$. Multiply both sides by $P^*(\pi)$ and sum over $\pi \in \Pi$ gives

$$\begin{aligned}
\sum_{\pi\in\Pi} P^\star(\pi)\text{Reg}_{t-1}(\pi) &= \frac{2(1-K\mu)}{\beta} \sum_{\pi\in\Pi} P^*(\pi)V(P^*,\pi) + \sum_{\pi\in\Pi} P^*(\pi)\lambda_\pi - \lambda \quad (P^* \in \Delta(\Pi)) \\
&= \frac{2(1-K\mu)}{\beta} \sum_{\pi\in\Pi} P^*(\pi)V(P^*,\pi) - \lambda \qquad \text{(complementary slackness)} \\
&\leq \frac{2}{\beta}\mathbb{E}_x\left[\sum_{\pi\in\Pi} \frac{P^*(\pi)}{P^*(\pi(x)|x)}\right] - \lambda = \frac{2K}{\beta} - \lambda \leq \frac{2K}{\beta},
\end{aligned}$$

showing that P^* satisfies the first condition. Moreover, the last equality above also implies $\lambda \leq \frac{2K}{\beta}$ since $\text{Reg}_{t-1}(\pi) \geq 0$. Rearranging Eq. (1) thus gives

$$V(P^*,\pi) \leq \frac{\beta}{2(1-K\mu)}\left(\text{Reg}_{t-1}(\pi) + \lambda\right) \leq 2K + \beta\text{Reg}_{t-1}(\pi),$$

where we assume $\mu \leq \frac{1}{2K}$ so that $2(1-K\mu) \geq 1$ (since otherwise we trivially have $V(P^*,\pi) \leq 1/\mu \leq 2K$). This shows that $P^\star$ satisfies the second condition too. □

The question is now what β we should use. Assuming $\text{Reg}_{t-1}(\pi)$ concentrates well around the actual expected regret of π compared to $\pi^\star$,

$$\text{Reg}(\pi) \stackrel{\text{def}}{=} \bar{\ell}(\pi) - \bar{\ell}(\pi^\star),$$

which is exactly what we hope for, $\text{Reg}_{t-1}(\pi)$ should be at most a constant. A reasonable choice of β would then be of order $1/\mu$, since $V(P,\pi)$ is trivially bounded by $1/\mu$. In other words, when a policy π is good, which means $\text{Reg}_{t-1}(\pi)$ is close to zero, we still require $V(P,\pi)$ to be close to $2K$, while when the policy is bad, which means $\text{Reg}_{t-1}(\pi)$ is a large constant, then there is almost no requirement on $V(P,\pi)$ with this choice of β.

On the other hand, this means that the exploitation constraint is $\sum_{\pi\in\Pi} P(\pi)\text{Reg}_{t-1}(\pi) = \mathcal{O}(K\mu)$, which also makes sense because μ should be of order $1/\sqrt{T}$, and if the per round regret is of order $1/\sqrt{T}$, then the overall regret over T rounds is of order $\sqrt{T}$. With some specific constant (chosen based on the analysis), this leads to the final algorithm called ILOVETOCONBANDITS (see Algorithm 1).

2 Oracle-Efficiency

To discuss oracle-efficiency, keep in mind that as in Policy Elimination, the true context distribution in the definition of V can be replaced by the empirical distribution of observed contexts, that is, a

Algorithm 1: ILOVETOCONBANDITS (colloquially referred as Mini-monster)

Input: failure probability $\delta \in (0,1)$

Initialization: let $\mu = \min\left\{\frac{1}{K}, \sqrt{\frac{\ln(TN/\delta)}{TK}}\ln T\right\}$

for $t = 1, \ldots, T$ **do**

find P_t such that

$$\sum_{\pi\in\Pi} P_t(\pi)\mathrm{Reg}_{t-1}(\pi) \le 20K\mu$$

$$V(P_t,\pi) \le 2K + \frac{\mathrm{Reg}_{t-1}(\pi)}{10\mu} \quad \forall\, \pi \in \Pi.$$

play $a_t \sim P_t^{\mu}(\cdot|x_t)$

uniform distribution over $x_1, \ldots, x_{t-1}$ at time t. (For simplicity, the analysis of next section will assume that the true context distribution is known instead.)

According to the proof of Lemma 1, to find distribution P_t it suffices to solve the optimization problem $\mathrm{argmin}_{P\in\Delta(\Pi)} F_t(P)$ (in fact an approximate solution is enough). This is in fact very similar to FTRL with a special regularizer. To see how to solve it efficiently with the oracle, notice that the derivative of $F_t(P)$ with respect to a policy π can be written as (with $\beta = 1/(10\mu)$)

$$\nabla F_t(P)(\pi) = \frac{1}{t-1}\sum_{\tau=1}^{t-1}\left(\widehat{\ell}_\tau(\pi(x_\tau)) - \widehat{\ell}_\tau(\pi^\star_{t-1}(x_\tau))\right) - \frac{20\mu(1-K\mu)}{(t-1)}\sum_{\tau=1}^{t-1}\frac{1}{P^\mu(\pi(x_\tau)|x_\tau)}.$$

Since the part involving $\pi^\star_{t-1}$ is independent of π, if we feed the oracle with a training set

$$\mathcal{S} = \left\{\left(x_1, \widehat{\ell}_1 - \frac{20\mu(1-K\mu)}{P^\mu(\,\cdot\,|x_1)}\right), \cdots, \left(x_{t-1}, \widehat{\ell}_{t-1} - \frac{20\mu(1-K\mu)}{P^\mu(\,\cdot\,|x_{t-1})}\right)\right\},$$

we have $\mathrm{ERM}(\mathcal{S}) = \mathrm{argmin}_{\pi\in\Pi}\nabla F_t(P)(\pi)$. In other words, the oracle can tell us the minimum coordinate of the gradient of $F_t(P)$ for any P, which opens up many possibilities to utilize the theory of optimization to find P_t. For example, one can directly apply the Frank-Wolfe algorithm (also known as conditional gradient method). Specifically, for a constraint convex optimization problem $\min_{w\in\Omega} f(w)$, the Frank-Wolfe algorithm performs the following iterative updates (staring with an arbitrary $w_1 \in \Omega$):

$$v_k = \operatorname*{argmin}_{v\in\Omega}\langle v, \nabla f(w_k)\rangle$$
$$w_{k+1} = (1-\gamma_k)w_k + \gamma_k v_k$$

for some step-size γ_k (default choice is $2/(k+1)$). When Ω is the simplex, the first step is exactly to find the minimum coordinate of the gradient. Therefore, with the oracle we can implement the Frank-Wolfe algorithm to solve $\mathrm{argmin}_{P\in\Delta(\Pi)} F_t(P)$ efficiently. We omit the details on how many iterations are needed but it will be polynomial in T, K, and $\ln N$.

Importantly, notice that unlike gradient descent, Frank-Wolfe leads to a sparse solution: when Ω is the simplex, after k rounds w_k has only k non-zero coordinates (assuming w_1 concentrates on one element to start with). This means that P_t's are all sparse distributions and operations involving P_t, such as constructing the training set $\mathcal{S}$ and sampling a_t, are all efficient.

Instead of using Frank-Wolfe, another possibility is to do some kind of coordinate descent: iteratively use the oracle to fine the coordinate with minimum derivative and adjust the weight for this coordinate appropriately. This is exactly the method taken in [Agarwal et al., 2014]. In fact, with additional tricks that are specialized for this task, it was shown that over T rounds only $\mathcal{O}(\sqrt{T})$ oracle calls are needed, which also implies that all P_t's are $\mathcal{O}(\sqrt{T})$-sparse.

3 Regret Analysis

Finally in this section we prove that ILOVETOCONBANDITS enjoys optimal regret. The key is to show the following concentration results on regret.

Lemma 2. *With probability $1-\delta/2$, Algorithm 1 ensures that for all $t \in [T]$ and all $\pi \in \Pi$,*

$$\mathrm{Reg}(\pi) \le 2\mathrm{Reg}_t(\pi) + \epsilon_t \quad \textit{and} \quad \mathrm{Reg}_t(\pi) \le 2\mathrm{Reg}(\pi) + \epsilon_t$$

where $\epsilon_t = \frac{20C}{\mu t} + 15K\mu$ and $C = \ln\left(\frac{4NT}{\delta}\right)\ln T$.

Proof. By Freedman's inequality and a union bound, we have with probability $1-\delta/2$, for all $t \in [T]$, all $\pi \in \Pi$, and any $\lambda \in [0, \mu]$,

$$|\bar{\ell}_t(\pi) - \bar{\ell}(\pi)| \le \frac{\lambda}{t}\sum_{\tau=1}^{t} V(P_\tau, \pi) + \frac{\ln\left(\frac{4NT}{\delta}\right)}{\lambda t}.$$

Specifically picking $\lambda = \frac{\mu}{\ln T}$ gives

$$|\bar{\ell}_t(\pi) - \bar{\ell}(\pi)| \le \frac{\mu}{t\ln T}\sum_{\tau=1}^{t} V(P_\tau, \pi) + \frac{C}{\mu t}. \tag{2}$$

Now we use induction to prove the lemma. The base case $t = 0$ is trivial. Assuming the statement holds for all rounds before time t, we have by the algorithm

$$V(P_\tau, \pi) \le 2K + \frac{\mathrm{Reg}_{\tau-1}(\pi)}{10\mu} \le 2K + \frac{\mathrm{Reg}(\pi)}{5\mu} + \frac{\epsilon_{\tau-1}}{10\mu} \tag{3}$$

for all $\tau = 2, \dots, t$ and $V(P_1, \pi) \le 2K$. Therefore, we have

$$\begin{aligned}
\mathrm{Reg}(\pi) - \mathrm{Reg}_t(\pi) &= \bar{\ell}(\pi) - \bar{\ell}(\pi^\star) - \bar{\ell}_t(\pi) + \bar{\ell}_t(\pi_t^\star) \\
&\le \bar{\ell}(\pi) - \bar{\ell}(\pi^\star) - \bar{\ell}_t(\pi) + \bar{\ell}_t(\pi^\star) && \text{(by optimality of } \pi_t^\star\text{)} \\
&\le \frac{2C}{\mu t} + \frac{\mu}{t\ln T}\sum_{\tau=1}^{t}\left(V(P_\tau, \pi) + V(P_\tau, \pi^\star)\right) && \text{(by Eq. (2))} \\
&\le \frac{2C}{\mu t} + \frac{4K\mu}{\ln T} + \frac{\mathrm{Reg}(\pi)}{5\ln T} + \frac{1}{5t\ln T}\sum_{\tau=2}^{t}\epsilon_{\tau-1} && \text{(by Eq. (3) and } \mathrm{Reg}(\pi^\star) = 0\text{)} \\
&\le \frac{2C}{\mu t} + \frac{4K\mu}{\ln T} + \frac{\mathrm{Reg}(\pi)}{5\ln T} + \frac{8C}{\mu t} + \frac{3K\mu}{\ln T} && \text{(by plugging in } \epsilon_\tau\text{)} \\
&\le \frac{10C}{\mu t} + 7K\mu + \frac{\mathrm{Reg}(\pi)}{2} \le \frac{\epsilon_t}{2} + \frac{\mathrm{Reg}(\pi)}{2}.
\end{aligned}$$

Rearranging proves $\mathrm{Reg}(\pi) \le 2\mathrm{Reg}_t(\pi) + \epsilon_t$. Similarly, we also have

$$\begin{aligned}
\mathrm{Reg}_t(\pi) - \mathrm{Reg}(\pi) &= \bar{\ell}_t(\pi) - \bar{\ell}_t(\pi_t^\star) - \bar{\ell}(\pi) + \bar{\ell}(\pi^\star) \\
&\le \bar{\ell}_t(\pi) - \bar{\ell}_t(\pi_t^\star) - \bar{\ell}(\pi) + \bar{\ell}(\pi_t^\star) && \text{(by optimality of } \pi^\star\text{)} \\
&\le \frac{2C}{\mu t} + \frac{\mu}{t\ln T}\sum_{\tau=1}^{t}\left(V(P_\tau, \pi) + V(P_\tau, \pi_t^\star)\right) && \text{(by Eq. (2))} \\
&\le \frac{2C}{\mu t} + \frac{4K\mu}{\ln T} + \frac{\mathrm{Reg}(\pi)}{5\ln T} + \frac{\mathrm{Reg}(\pi_t^\star)}{5\ln T} + \frac{1}{5t\ln T}\sum_{\tau=2}^{t}\epsilon_{\tau-1} && \text{(by Eq. (3))} \\
&\le \frac{2C}{\mu t} + \frac{4K\mu}{\ln T} + \frac{\mathrm{Reg}(\pi)}{5\ln T} + \frac{\epsilon_t}{5\ln T} + \frac{8C}{\mu t} + \frac{3K\mu}{\ln T} && (4) \\
&\le \frac{14C}{\mu t} + 10K\mu + \mathrm{Reg}(\pi) \le \epsilon_t + \mathrm{Reg}(\pi),
\end{aligned}$$

where Step (4) uses the fact $\mathrm{Reg}(\pi) \le 2\mathrm{Reg}_t(\pi) + \epsilon_t$ just proven above with π set to $\pi_t^\star$, and also the fact $\mathrm{Reg}_t(\pi_t^\star) = 0$. Rearranging then proves $\mathrm{Reg}_t(\pi) \le 2\mathrm{Reg}(\pi) + \epsilon_t$ as well. □

The final regret bound is now a simple application of this lemma and the exploitation constraint of the algorithm.

Theorem 1. *Algorithm 1 ensures that with probability* $1-\delta$*, we have* $\mathcal{R}_T = \widetilde{\mathcal{O}}\left(\sqrt{TK\ln(N/\delta)}\right)$.

Proof. The first step is exactly the same as analyzing Policy Elimination: by Azuma's inequality we have with probability $1-\delta/2$,

$$\sum_{t=1}^{T} \ell_t(a_t) \leq \sum_{t=1}^{T} \sum_{\pi\in\Pi} P_t(\pi)\bar{\ell}(\pi) + TK\mu + \mathcal{O}\left(\sqrt{T\ln(1/\delta)}\right).$$

Conditioning on this event and the event stated in Lemma 2, which happen simultaneously with probability $1-\delta$, we have

$$\begin{aligned}
\mathcal{R}_T &\leq \sum_{t=1}^{T} \sum_{\pi\in\Pi} P_t(\pi)\mathrm{Reg}(\pi) + TK\mu + \mathcal{O}\left(\sqrt{T\ln(1/\delta)}\right) \\
&\leq 2\sum_{t=1}^{T} \sum_{\pi\in\Pi} P_t(\pi)\mathrm{Reg}_{t-1}(\pi) + \sum_{t=2}^{T} \epsilon_{t-1} + TK\mu + \mathcal{O}\left(\sqrt{T\ln(1/\delta)}\right) \\
&\leq 56TK\mu + \frac{40C\ln T}{\mu} + \mathcal{O}\left(\sqrt{T\ln(1/\delta)}\right), \qquad \text{(by the exploitation constraint)}
\end{aligned}$$

which is of order $\widetilde{\mathcal{O}}\left(\sqrt{TK\ln(N/\delta)}\right)$ with the optimal tuning of μ. □

References

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Lecture 22

Instructor: Haipeng Luo

1 Contextual Bandit with Adversarial Losses

We have seen oracle-efficient and optimal algorithms for the i.i.d. contextual bandit problems. Going beyond the i.i.d. assumption remains a challenging question and we will discuss some recent progress from [Rakhlin and Sridharan, 2016, Syrgkanis et al., 2016] in this lecture.

Specifically we study a hybrid setting where the contexts are i.i.d samples from a fixed but unknown distribution $\mathcal{D}$, while the losses can be adversarial. This could be a reasonable assumption for the personalized recommendation problem: users' contextual information such as gender might be relatively stationary (e.g. 40% men and 60% women for a shopping website), while the actual preferences for items might be changing more rapidly. We also assume that we can draw fresh examples from $\mathcal{D}$ as we want. This is a somewhat technical assumption but in some cases can still be very reasonable.

Recall that in Lecture 12 the very first attempt we tried for bandit was to reduce it to the expert problem in a blackbox way. Here we will take the same approach. Specifically, we first consider deriving oracle-efficient and optimal algorithms for the following full information problem: for each round $t = 1, \ldots, T$,

1. environment draws $x_t \sim \mathcal{D}$ and reveals x_t;
2. learner decides a distribution $p_t \in \Delta(K)$;
3. environment decides a loss vector $\widehat{\ell}_t \in \mathcal{L} \subset [0,1]^K$;
4. learner suffers loss $\left\langle p_t, \widehat{\ell}_t \right\rangle$ and observes $\widehat{\ell}_t$.

Here $\mathcal{L}$ is a special loss space to be discussed soon. The regret is defined as the difference between the loss of the algorithm and the best fixed policy from a policy class Π:

$$\mathcal{R}_T = \sum_{t=1}^{T} \left\langle p_t, \widehat{\ell}_t \right\rangle - \min_{\pi \in \Pi} \sum_{t=1}^{T} \widehat{\ell}_t(\pi(x_t)).$$

Now suppose we have an oracle-efficient algorithm to solve such a full information problem. Then for the hybrid contextual bandit problem, we can simply sample an action a_t according to $(1 - K\mu)p_t + \mu\mathbf{1}$, construct the usual importance-weighted estimator, and finally rescale it by μ and feed it to the full information algorithm as the loss vector $\widehat{\ell}_t$. It is then clear that the loss space $\mathcal{L}$ for this reduction has a special structure and can be defined as $\{ce_a : a \in [K], c \in [0,1]\}$ where $e_1, \ldots, e_K$ are the K-dimensional standard basis vectors.

Note that the optimal regret for this full information problem is $\mathcal{O}(\sqrt{T \ln N})$ (achieved by applying Hedge). Having an oracle-efficient algorithm for this problem with regret $\mathcal{O}(\sqrt{T \ln N})$ will then lead to expected regret $\mathcal{O}(T^{\frac{3}{4}} K^{\frac{1}{2}} (\ln N)^{\frac{1}{4}})$ [Rakhlin and Sridharan, 2016], by the exact same argument as in Lecture 12. This is clearly not the optimal regret. Recent work [Syrgkanis et al., 2016] improves the regret to $\mathcal{O}((TK)^{\frac{2}{3}} (\ln N)^{\frac{1}{3}})$, while getting the optimal regret with an oracle-efficient algorithm $\mathcal{O}(\sqrt{TK \ln N})$ is still open.

2 Relaxation-based Approach

We now discuss how to solve the full information problem. The approach is based on a minimax analysis. Specifically, first note that the optimal worst-case expected regret (with respect to the random contexts) can be written as a sequence of minimax expressions

$$\mathbb{E}_{x_1} \min_{p_1 \in \Delta(K)} \max_{\widehat{\ell}_1 \in \mathcal{L}} \cdots \mathbb{E}_{x_T} \min_{p_T \in \Delta(K)} \max_{\widehat{\ell}_T \in \mathcal{L}} \left(\sum_{\tau=1}^{T} \left\langle p_\tau, \widehat{\ell}_\tau \right\rangle - \min_{\pi \in \Pi} \sum_{\tau=1}^{T} \widehat{\ell}_\tau(\pi(x_\tau)) \right) \tag{1}$$

More generally, at the beginning of time t, having observed $x_{1:t-1}$ and $\widehat{\ell}_{1:t-1}$,[1] and assuming the player and the environment will both play optimally afterwards, the conditional expected regret is

$$\begin{aligned}
&\mathbb{E}_{x_t} \min_{p_t \in \Delta(K)} \max_{\widehat{\ell}_t \in \mathcal{L}} \cdots \mathbb{E}_{x_T} \min_{p_T \in \Delta(K)} \max_{\widehat{\ell}_T \in \mathcal{L}} \left(\sum_{\tau=1}^{T} \left\langle p_\tau, \widehat{\ell}_\tau \right\rangle - \min_{\pi \in \Pi} \sum_{\tau=1}^{T} \widehat{\ell}_\tau(\pi(x_\tau)) \right) \\
&= \sum_{\tau=1}^{t-1} \left\langle p_\tau, \widehat{\ell}_\tau \right\rangle + \underbrace{\mathbb{E}_{x_t} \min_{p_t \in \Delta(K)} \max_{\widehat{\ell}_t \in \mathcal{L}} \cdots \mathbb{E}_{x_T} \min_{p_T \in \Delta(K)} \max_{\widehat{\ell}_T \in \mathcal{L}} \left(\sum_{\tau=t}^{T} \left\langle p_\tau, \widehat{\ell}_\tau \right\rangle - \min_{\pi \in \Pi} \sum_{\tau=1}^{T} \widehat{\ell}_\tau(\pi(x_\tau)) \right)}_{\Phi(x_{1:t-1}, \widehat{\ell}_{1:t-1})}.
\end{aligned}$$

We denote the last term as $\Phi(x_{1:t-1}, \widehat{\ell}_{1:t-1})$, which can be seen as the optimal "regret" (in fact only the part of the regret that we can still control), starting from the state $(x_{1:t-1}, \widehat{\ell}_{1:t-1})$. Note that we have the following recursive relationship:

$$\Phi(x_{1:t-1}, \widehat{\ell}_{1:t-1}) = \mathbb{E}_{x_t} \min_{p_t \in \Delta(K)} \max_{\widehat{\ell}_t \in \mathcal{L}} \left(\left\langle p_t, \widehat{\ell}_t \right\rangle + \Phi(x_{1:t}, \widehat{\ell}_{1:t}) \right).$$

Also note that $\Phi(x_{1:T}, \widehat{\ell}_{1:T})$ is the negative benchmark $-\min_{\pi \in \Pi} \sum_{\tau=1}^{T} \widehat{\ell}_\tau(\pi(x_\tau))$ and $\Phi(\emptyset)$ is exactly the minimax regret Eq. (1). Moreover, using Φ we can derive the minimax optimal algorithm: pick p_t to be

$$\operatorname*{argmin}_{p \in \Delta(K)} \max_{\widehat{\ell}_t \in \mathcal{L}} \left(\left\langle p, \widehat{\ell}_t \right\rangle + \Phi(x_{1:t}, \widehat{\ell}_{1:t}) \right).$$

It is clear that the expected regret of this algorithm is bounded by $\Phi(\emptyset)$ since

$$\begin{aligned}
&\mathbb{E}_{x_t} \left[\left\langle p_t, \widehat{\ell}_t \right\rangle + \Phi(x_{1:t}, \widehat{\ell}_{1:t}) \right] \le \mathbb{E}_{x_t} \max_{\widehat{\ell}_t \in \mathcal{L}} \left(\left\langle p_t, \widehat{\ell}_t \right\rangle + \Phi(x_{1:t}, \widehat{\ell}_{1:t}) \right) \\
&= \mathbb{E}_{x_t} \min_{p_t \in \Delta(K)} \max_{\widehat{\ell}_t \in \mathcal{L}} \left(\left\langle p_t, \widehat{\ell}_t \right\rangle + \Phi(x_{1:t}, \widehat{\ell}_{1:t}) \right) = \Phi(x_{1:t-1}, \widehat{\ell}_{1:t-1})
\end{aligned} \tag{2}$$

and thus

$$\begin{aligned}
\mathbb{E}_{x_{1:T}}[\mathcal{R}_T] &= \mathbb{E}_{x_{1:T}} \left[\sum_{\tau=1}^{T} \left\langle p_\tau, \widehat{\ell}_\tau \right\rangle + \Phi(x_{1:T}, \widehat{\ell}_{1:T}) \right] \\
&\le \mathbb{E}_{x_{1:T-1}} \left[\sum_{\tau=1}^{T-1} \left\langle p_\tau, \widehat{\ell}_\tau \right\rangle + \Phi(x_{1:T-1}, \widehat{\ell}_{1:T-1}) \right] \le \cdots \le \Phi(\emptyset).
\end{aligned}$$

Therefore, the notion of Φ completely characterizes the optimal regret and algorithm. However, in general Φ is highly complicated and intractable, making the approach above only theoretically interesting. Nevertheless, in light of Eq. (2), if we can come up with a different and tractable function Rel and a strategy such that

$$\mathbb{E}_{x_t} \max_{\widehat{\ell}_t \in \mathcal{L}} \left(\left\langle p_t, \widehat{\ell}_t \right\rangle + \mathrm{Rel}(x_{1:t}, \widehat{\ell}_{1:t}) \right) \le \mathrm{Rel}(x_{1:t-1}, \widehat{\ell}_{1:t-1}) \tag{3}$$

and in addition $\Phi(x_{1:T}, \widehat{\ell}_{1:T}) \le \mathrm{Rel}(x_{1:T}, \widehat{\ell}_{1:T})$, then by the exact same argument we have $\mathbb{E}_{x_{1:T}}[\mathcal{R}_T] \le \mathrm{Rel}(\emptyset)$. Such function Rel is called a relaxation of Φ and is indeed an upper bound of Φ (check it yourself by a simple induction). The hope is thus Rel should be as small as possible so that the final regret bound $\mathrm{Rel}(\emptyset)$ is still of order $\mathcal{O}(\sqrt{T \ln N})$.

The question is now how to come up with a reasonable relaxation. To see this, we will simply let $\mathrm{Rel}(x_{1:T}, \widehat{\ell}_{1:T}) = \Phi(x_{1:T}, \widehat{\ell}_{1:T})$ and first see what $\mathrm{Rel}(x_{1:T-1}, \widehat{\ell}_{1:T-1})$ should be.

[1] We use the notation $z_{1:t}$ to denote the sequence $z_1, \ldots, z_t$.

2.1 Warm-up: Finding $\mathrm{Rel}(x_{1:T-1}, \widehat{\ell}_{1:T-1})$

Note that the existence of strategy p_t such that Eq. (3) holds implies

$$\mathbb{E}_{x_T} \min_{p_T \in \Delta(K)} \max_{\widehat{\ell}_T \in \mathcal{L}} \left[\left\langle p_T, \widehat{\ell}_T \right\rangle + \mathrm{Rel}(x_{1:T}, \widehat{\ell}_{1:T}) \right] \le \mathrm{Rel}(x_{1:T-1}, \widehat{\ell}_{1:T-1}).$$

We now work on the term on the left-hand side and relax it step by step:

$$\begin{aligned}
&\mathbb{E}_{x_T} \min_{p_T \in \Delta(K)} \max_{\widehat{\ell}_T \in \mathcal{L}} \left[\left\langle p_T, \widehat{\ell}_T \right\rangle + \Phi(x_{1:T}, \widehat{\ell}_{1:T}) \right] \\
&= \mathbb{E}_{x_T} \min_{p_T \in \Delta(K)} \max_{q_T \in \Delta(\mathcal{L})} \mathbb{E}_{\widehat{\ell}_T \sim q_T} \left[\left\langle p_T, \widehat{\ell}_T \right\rangle + \Phi(x_{1:T}, \widehat{\ell}_{1:T}) \right] \\
&= \mathbb{E}_{x_T} \max_{q_T \in \Delta(\mathcal{L})} \min_{p_T \in \Delta(K)} \mathbb{E}_{\widehat{\ell}_T \sim q_T} \left[\left\langle p_T, \widehat{\ell}_T \right\rangle + \Phi(x_{1:T}, \widehat{\ell}_{1:T}) \right] && \text{(Sion's minimax theorem)} \\
&= \mathbb{E}_{x_T} \max_{q_T \in \Delta(\mathcal{L})} \left(\left(\min_{p_T \in \Delta(K)} \mathbb{E}_{\widehat{\ell}'_T \sim q_T} \left[\left\langle p_T, \widehat{\ell}'_T \right\rangle \right] \right) + \mathbb{E}_{\widehat{\ell}_T \sim q_T} \left[\max_{\pi \in \Pi} - \sum_{t=1}^{T} \widehat{\ell}_t(\pi(x_t)) \right] \right) \\
&= \mathbb{E}_{x_T} \max_{q_T \in \Delta(\mathcal{L})} \mathbb{E}_{\widehat{\ell}_T \sim q_T} \left[\max_{\pi \in \Pi} \left(\left(\min_{p_T \in \Delta(K)} \mathbb{E}_{\widehat{\ell}'_T \sim q_T} \left[\left\langle p_T, \widehat{\ell}'_T \right\rangle \right] \right) - \widehat{\ell}_T(\pi(x_T)) - \sum_{t=1}^{T-1} \widehat{\ell}_t(\pi(x_t)) \right) \right] \\
&\le \mathbb{E}_{x_T} \max_{q_T \in \Delta(\mathcal{L})} \mathbb{E}_{\widehat{\ell}_T \sim q_T} \left[\max_{\pi \in \Pi} \left(\mathbb{E}_{\widehat{\ell}'_T \sim q_T} \left[\widehat{\ell}'_T(\pi(x_T)) \right] - \widehat{\ell}_T(\pi(x_T)) - \sum_{t=1}^{T-1} \widehat{\ell}_t(\pi(x_t)) \right) \right] \\
&\le \mathbb{E}_{x_T} \max_{q_T \in \Delta(\mathcal{L})} \mathbb{E}_{\widehat{\ell}_T, \widehat{\ell}'_T \sim q_T} \left[\max_{\pi \in \Pi} \left(\widehat{\ell}'_T(\pi(x_T)) - \widehat{\ell}_T(\pi(x_T)) - \sum_{t=1}^{T-1} \widehat{\ell}_t(\pi(x_t)) \right) \right] \\
&= \mathbb{E}_{x_T} \max_{q_T \in \Delta(\mathcal{L})} \mathbb{E}_{\widehat{\ell}_T, \widehat{\ell}'_T \sim q_T, \sigma} \left[\max_{\pi \in \Pi} \left(\sigma \left(\widehat{\ell}'_T(\pi(x_T)) - \widehat{\ell}_T(\pi(x_T)) \right) - \sum_{t=1}^{T-1} \widehat{\ell}_t(\pi(x_t)) \right) \right] \\
&\qquad \text{(σ is Rademacher variable, that is, uniformly drawn from $\{-1, 1\}$)} \\
&\le \mathbb{E}_{x_T} \max_{q_T \in \Delta(\mathcal{L})} \mathbb{E}_{\widehat{\ell}_T, \widehat{\ell}'_T \sim q_T, \sigma} \left[\max_{\pi \in \Pi} \left(\sigma \widehat{\ell}'_T(\pi(x_T)) - \frac{1}{2} \sum_{t=1}^{T-1} \widehat{\ell}_t(\pi(x_t)) \right) + \right. \\
&\qquad \left. \max_{\pi \in \Pi} \left(-\sigma \widehat{\ell}_T(\pi(x_T)) - \frac{1}{2} \sum_{t=1}^{T-1} \widehat{\ell}_t(\pi(x_t)) \right) \right] \\
&= \mathbb{E}_{x_T} \max_{q_T \in \Delta(\mathcal{L})} \mathbb{E}_{\widehat{\ell}_T \sim q_T, \sigma} \left[\max_{\pi \in \Pi} \left(2\sigma \widehat{\ell}_T(\pi(x_T)) - \sum_{t=1}^{T-1} \widehat{\ell}_t(\pi(x_t)) \right) \right] \\
&\qquad \text{(σ and $-\sigma$ follow the same distribution)} \\
&= \mathbb{E}_{x_T} \max_{\widehat{\ell}_T \in \mathcal{L}} \mathbb{E}_{\sigma} \left[\max_{\pi \in \Pi} \left(2\sigma \widehat{\ell}_T(\pi(x_T)) - \sum_{t=1}^{T-1} \widehat{\ell}_t(\pi(x_t)) \right) \right] \\
&= \mathbb{E}_{x_T} \max_{\widehat{\ell}_T \in \mathcal{L}'} \mathbb{E}_{\sigma} \left[\max_{\pi \in \Pi} \left(2\sigma \widehat{\ell}_T(\pi(x_T)) - \sum_{t=1}^{T-1} \widehat{\ell}_t(\pi(x_t)) \right) \right]
\end{aligned}$$

where $\mathcal{L}' = \{\mathbf{0}, e_1, \ldots, e_K\}$ and last step is because maximizers of a convex function are always on the boundary. To continue, note that if $\widehat{\ell}_T = e_a$ for some $a \in [K]$, we can construct e'_a such that $e'_a(a) = 1$ and $e'_a(a')$ when $a' \neq a$ is an independent Rademacher variable, so that $\mathbb{E}_{e'_a}[e'_a] = e_a$ and

$$\begin{aligned}
\mathbb{E}_{\sigma} \left[\max_{\pi \in \Pi} \left(2\sigma e_a(\pi(x_T)) - \sum_{t=1}^{T-1} \widehat{\ell}_t(\pi(x_t)) \right) \right] &= \mathbb{E}_{\sigma} \left[\max_{\pi \in \Pi} \left(2\sigma \mathbb{E}_{e'_a}[e'_a(\pi(x_T))] - \sum_{t=1}^{T-1} \widehat{\ell}_t(\pi(x_t)) \right) \right] \\
&\le \mathbb{E}_{\sigma, e'_a} \left[\max_{\pi \in \Pi} \left(2\sigma e'_a(\pi(x_T)) - \sum_{t=1}^{T-1} \widehat{\ell}_t(\pi(x_t)) \right) \right].
\end{aligned}$$

Now if one looks at the random vector $\sigma e'_a$, it is clear that each coordinate of it is an independent Rademacher variable. We denote such random vector by ϵ_T and continue the bound with

$$\mathbb{E}_{\epsilon_T}\left[\max_{\pi\in\Pi}\left(2\epsilon_T(\pi(x_T)) - \sum_{t=1}^{T-1}\widehat{\ell}_t(\pi(x_t))\right)\right].$$

Note that this is also an upper bound when $\widehat{\ell}_T = \mathbf{0}$ since

$$\begin{aligned}\mathbb{E}_{\epsilon_T}\left[\max_{\pi\in\Pi}\left(2\epsilon_T(\pi(x_T)) - \sum_{t=1}^{T-1}\widehat{\ell}_t(\pi(x_t))\right)\right] &\geq \max_{\pi\in\Pi}\left(\mathbb{E}_{\epsilon_T}\left[2\epsilon_T(\pi(x_T)) - \sum_{t=1}^{T-1}\widehat{\ell}_t(\pi(x_t))\right]\right)\\ &= \max_{\pi\in\Pi}\left(-\sum_{t=1}^{T-1}\widehat{\ell}_t(\pi(x_t))\right).\end{aligned}$$

Therefore, we have shown

$$\mathbb{E}_{x_T}\min_{p_T\in\Delta(K)}\max_{\widehat{\ell}_T\in\mathcal{L}}\left[\left\langle p_T, \widehat{\ell}_T\right\rangle + \mathrm{Rel}(x_{1:T}, \widehat{\ell}_{1:T})\right] \leq \mathbb{E}_{x_T,\epsilon_T}\left[\max_{\pi\in\Pi}\left(2\epsilon_T(\pi(x_T)) - \sum_{t=1}^{T-1}\widehat{\ell}_t(\pi(x_t))\right)\right],$$

and can thus denote the last term by $\mathrm{Rel}(x_{1:T-1}, \widehat{\ell}_{1:T-1})$.

2.2 Generalizing the Argument

Compared to $\mathrm{Rel}(x_{1:T}, \widehat{\ell}_{1:T})$, one can see that in $\mathrm{Rel}(x_{1:T-1}, \widehat{\ell}_{1:T-1})$ the loss for the last round $-\widehat{\ell}_T(\pi(x_T))$ is replaced by the random loss $2\epsilon_T(\pi(x_T))$. This motivates us to define

$$\begin{aligned}\mathrm{Rel}(x_{1:t}, \widehat{\ell}_{1:t}) &= \mathbb{E}_{x_{t+1,T},\epsilon_{t+1,T}}\left[\max_{\pi\in\Pi}\left(2\sum_{\tau=t+1}^{T}\epsilon_\tau(\pi(x_\tau)) - \sum_{\tau=1}^{t}\widehat{\ell}_\tau(\pi(x_\tau))\right)\right]\\ &= \mathbb{E}_{x_{t+1,T},\epsilon_{t+1,T}}\left[\Phi(x_{1:T}, \widehat{\ell}_{1:t}, 2\epsilon_{t+1,T})\right], \qquad (4)\end{aligned}$$

which is saying that we should replace all the future losses by $2\epsilon_\tau$ and this leads to the worst-case regret. To make this algorithmic, we need to find a strategy such that Eq. (3) holds. While the most natural choice is

$$p_t = \operatorname*{argmin}_{p\in\Delta(K)}\max_{\widehat{\ell}_t\in\mathcal{L}}\left(\left\langle p, \widehat{\ell}_t\right\rangle + \mathrm{Rel}(x_{1:t}, \widehat{\ell}_{1:t})\right).$$

This still does not lead to an oracle-efficient algorithm since computing Rel is intractable. However, it turns out that the following strategy suffices:

$$p_t = \mathbb{E}_{x_{t+1,T},\epsilon_{t+1,T}}\left[p_t(x_{t+1,T}, \epsilon_{t+1,T})\right] \qquad (5)$$

where

$$p_t(x_{t+1,T}, \epsilon_{t+1,T}) = \operatorname*{argmin}_{p\in\Delta(K)}\max_{\widehat{\ell}_t\in\mathcal{L}}\left(\left\langle p, \widehat{\ell}_t\right\rangle + \Phi(x_{1:T}, \widehat{\ell}_{1:t}, 2\epsilon_{t+1,T})\right). \qquad (6)$$

Before discussing the oracle-efficiency of this strategy, let's first verify Eq. (3) is indeed satisfied.

Theorem 1. *The relaxation defined in Eq.* (4) *and the strategy defined in Eq.* (5) *satisfy Eq.* (3)*.*

Proof. The left-hand side of Eq. (3) can be bounded as follows:

$$\begin{aligned}&\mathbb{E}_{x_t}\max_{\widehat{\ell}_t\in\mathcal{L}}\left(\left\langle p_t, \widehat{\ell}_t\right\rangle + \mathrm{Rel}(x_{1:t}, \widehat{\ell}_{1:t})\right)\\ &= \mathbb{E}_{x_t}\max_{\widehat{\ell}_t\in\mathcal{L}}\left(\mathbb{E}_{x_{t+1,T},\epsilon_{t+1,T}}\left[\left\langle p_t(x_{t+1,T}, \epsilon_{t+1,T}), \widehat{\ell}_t\right\rangle\right] + \mathbb{E}_{x_{t+1,T},\epsilon_{t+1,T}}\left[\Phi(x_{1:T}, \widehat{\ell}_{1:t}, 2\epsilon_{t+1,T})\right]\right)\\ &\leq \mathbb{E}_{x_{t:T},\epsilon_{t+1,T}}\max_{\widehat{\ell}_t\in\mathcal{L}}\left(\left\langle p_t(x_{t+1,T}, \epsilon_{t+1,T}), \widehat{\ell}_t\right\rangle + \Phi(x_{1:T}, \widehat{\ell}_{1:t}, 2\epsilon_{t+1,T})\right)\\ &= \mathbb{E}_{x_{t:T},\epsilon_{t+1,T}}\min_{p\in\Delta(K)}\max_{\widehat{\ell}_t\in\mathcal{L}}\left(\left\langle p, \widehat{\ell}_t\right\rangle + \Phi(x_{1:T}, \widehat{\ell}_{1:t}, 2\epsilon_{t+1,T})\right).\end{aligned}$$

By repeating the exact same argument in the warm-up section, the last quantity is bounded by $\mathrm{Rel}(x_{1:t-1}, \widehat{\ell}_{1:t-1})$. □

Algorithm 1: Water-filling

Input: $B_1 \leq \cdots \leq B_K$
Ouput: solution of $\operatorname{argmin}_{p \in \Delta(K)} \max_{a \in [K]} (p(a) + B_a)$
Initialization: let $p = \mathbf{0}$, $S = 1$, $B_{K+1} = +\infty$
for $i = 1, \ldots, K$ **do**
  $s = \min\{(B_{i+1} - B_i)i, S\}$
  for $j = 1, \ldots, i$ **do** $p(j) \leftarrow p(j) + s/i$
  $S \leftarrow S - s$

Finally, is this relaxation tight enough? In other words, how large is the regret bound $\mathrm{Rel}(\emptyset)$? In fact, $\mathrm{Rel}(\emptyset)$ is a variant of the *Rademacher complexity* of the policy class Π and can be shown to be at most $2\sqrt{2T \ln N}$. This means that the relaxation is very tight and the strategy above enjoys the optimal regret.

2.3 Oracle-efficiency

To see why the strategy is efficient, first note that since we only care about expected regret, there is no difference in playing p_t or $p_t(x_{t+1,T}, \epsilon_{t+1,T})$ with a random draw of $x_{t+1,T}, \epsilon_{t+1,T}$. It thus suffices to solve the optimization problem defined in Eq. (6). It is not hard to see that the maximum over $\mathcal{L}$ can only be obtained by one of the K basis vectors $e_1, \ldots, e_K$.[2] Therefore, with

$$B_a = \Phi(x_{1:T}, \widehat{\ell}_{1:t-1}, e_a, 2\epsilon_{t+1,T}),$$

which clearly can be computed by calling the oracle once, the optimization problem becomes

$$\operatorname*{argmin}_{p \in \Delta(K)} \max_{a \in [K]} (p(a) + B_a).$$

This can be simply solved by a so-called water-filling procedure. Specifically, assuming $B_1 \leq \cdots \leq B_K$ without loss of generality, the solution can be found by Algorithm 1. It is therefore clear that for each round the algorithm makes K oracle-calls and all the other operations run in time $\mathrm{poly}(T, K)$.

Note that the algorithm shares some similarity with FTPL, both of which hallucinate some fake data and solve an offline optimization problem involving both the observed data and the hallucinated data.

References

Alexander Rakhlin and Karthik Sridharan. Bistro: An efficient relaxation-based method for contextual bandits. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

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 29*, 2016.

[2] Indeed, the objective is convex in $\widehat{\ell}_t$ and thus the maximum over $\mathcal{L}$ can only be obtained by $\mathcal{L}'$. Moreover, one can verify that $\mathbf{0}$ can not be the maximum since the average of the other K choices is larger.