LECTURE #18

LECTURER: ELAD HAZAN SCRIBE: JEREMY COHEN

1 Learning with Partial Feedback

Recall the "learning from expert advice" game: there are N experts indexed $i \in [N]$, and in the t-th round, the player picks one expert i_t and "suffers the loss" associated with that expert $f_t(i_t) \in \{0, 1\}$. The player's goal is to choose the experts $i_1, \ldots, i_T$ so as to minimize the *regret*: the difference between the total loss suffered by the experts whom the player selected, and the total loss suffered by expert i^* who was, in hindsight, the best:

$$\text{Regret} = \sum_{t=1}^{T} f_t(i_t) - \min_{i^* \in [N]} \sum_{t=1}^{T} f_t(i^*)$$

Crucially, in each round t of the "learning from expert advice" game, the player is given not only the loss of the expert whom she chose, which is a single number $f_t(i_t) \in \{0, 1\}$, but also the loss of *all the other experts*, which is a function $f_t : [N] \to \{0, 1\}$. The multiplicative weights (MW) algorithm exploited this additional information, by maintaining a "weight" for each expert and penalizing *all* experts who were wrong in round t, not just the expert i_t who was chosen. However, in many real-world problems that we may wish to model using online convex optimization, the "player" does not have access to the "loss" associated with decisions other than the one that the player made. For example, if a (hypothetical) Princeton undergraduate is trying to optimize his course schedule over his $T = 8$ semesters in college so as to minimize the number of essays that he has to write, he does not know how many essays were assigned in classes that he did not take. This motivates the so-called **multi-armed bandit** problem, a variant of "learning from expert advice" in which the player only observes the regret of the expert whom she chose.

1.1 The Multi-Armed Bandit Problem

We have N experts. In each round of play $t = 1, 2, \ldots T$, the player picks one expert $i_t \in [N]$ and suffers a loss $f_t(i_t) \in [0, 1]$. (Note that the loss here is real-valued, not binary as it was above.) The goal is to minimize regret, defined as above.

Recall that in "learning from expert advice," the multiplicative weights algorithm guaranteed a regret bounded by $\sqrt{T \log N}$. Is there an algorithm that can attain the same regret bound in the multi-armed bandit case? Unfortunately,

Theorem 1.1. *Any algorithm for the multi-armed bandit problem might attain $\Omega(N)$ regret in the worst case.*

Proof. Consider an situation where $N - 1$ of the experts always give a loss of 1, and one expert, chosen uniformly at random, always gives a loss of 0.

An optimal learner will keep trying experts until it finds the good one, thus:

$$\begin{aligned}
\mathbf{E}[\text{Regret}] &= \mathbf{E}\left[\sum_{t=1}^{T} f_t(i_t) - \min_{i^* \in [N]} \sum_{t=1}^{T} f_t(i^*)\right] \\
&= \mathbf{E}\left[\sum_{i=1}^{T} f_t(i_t)\right] \\
&= \mathbf{E}\left[\text{number of iterations to find good expert}\right] \\
&= \sum_{j=1}^{N} \mathbb{P}[\text{takes} \geq j \text{ iterations to find good expert }] \\
&= \frac{N-1}{N} + \frac{N-1}{N}\frac{N-2}{N-1} + \frac{N-3}{N-2}\frac{N-2}{N-1}\frac{N-1}{N} + \dots \\
&= \frac{N-1}{N} + \frac{N-2}{N} + \frac{N-3}{N} + \dots + \frac{1}{N} \\
&= \Omega\left(\frac{N^2}{N}\right) \\
&= \Omega(N)
\end{aligned}$$

1.2 A Simple Algorithm

Intuitively, any algorithm for the multi-armed bandit problem will have to navigate the tradeoff between "exploration" (learning which experts are best) and "exploitation" (playing those experts). One simple approach is to flip a coin on each round and, with probability δ, do an "explore" step, and with probability $1-\delta$, do an "exploit" step. Much like in the MW algorithm, we'll maintain a weight for each expert which will keep track of how "correct" that expert has been thus far. In the exploration phase, we'll play a random expert and adjust the weights, and in the exploitation phase, we'll play an expert according to the weights.

Since we can't update the weights according to the *actual* loss incurred by the all of the experts, we'll instead update the weights according to an unbiased estimator of the losses. Specifically, let $\hat{\ell}_t \in \mathbb{R}^N$ be a random variable whose expectation is

$$\mathbf{E}[\hat{\ell}_t(i)] = f_t(i)$$

Algorithm 1 Simple Bandits Algorithm

Let $\mathbf{x}_1 = \frac{1}{N}\mathbf{1}$
for t=1 to T **do**
  if Bernoulli(δ) is 1 **then**
    pick expert $i_t \in [N]$ uniformly at random
    set $\hat{\ell}_t(i) = \begin{cases} \frac{N}{\delta} f_t(i) & \text{if } i = i_t \\ 0 & \text{otherwise} \end{cases}$
  else
    pick expert $i_t \sim \mathbf{x}_t$
    set $\hat{\ell}_t(i) = 0$
  update $\mathbf{y}_{t+1}(i) = \mathbf{x}_t(i)\, e^{-\varepsilon \hat{\ell}_t(i)} \quad \mathbf{x}_{t+1} = \frac{\mathbf{y}_{t+1}}{||\mathbf{y}_{t+1}||_1}$

We claim that under this scheme, $\mathbf{E}[\hat{\ell}_t(i)] = f_t(i)$. Why?

$$\begin{aligned}
\mathbf{E}[\hat{\ell}_t(i)] &= \mathbf{E}[\hat{\ell}_t(i)|t \text{ is explore}] \Pr[t \text{ is explore}] + \mathbf{E}[\hat{\ell}_t(i)|t \text{ is exploit}] \Pr[t \text{ is exploit}] \\
&= \mathbf{E}[\hat{\ell}_t(i)|t \text{ is explore}](\delta) + (0)(1-\delta)) \\
&= \delta \, \mathbf{E}[\hat{\ell}_t(i)|t \text{ is explore}] \\
&= \delta \sum_{i=1}^{N} \mathbf{E}\left[\hat{\ell}_t(i)|t \text{ is explore}, i_t = i\right] \Pr\left[i_t = i\right] \\
&= \delta \left(\frac{N}{\delta} f_t(i)\right)\left(\frac{1}{N}\right) \\
&= f_t(i)
\end{aligned}$$

Theorem 1.2. *The expected regret of this simple algorithm is bounded by* $O(T^{\frac{3}{4}}\sqrt{N}(\log N)^{\frac{1}{4}})$.

Proof.

$$\begin{aligned}
\mathbf{E}[\text{Regret}] &= \mathbf{E}\left[\sum_{t=1}^{T} f_t(i_t) - \min_{i\in[N]} \sum_{t=1}^{T} f_t(i)\right] \\
&= \mathbf{E}\left[\sum_{t=1}^{T} \hat{\ell}_t(i_t) - \min_{i\in[N]} \sum_{t=1}^{T} \hat{\ell}_t(i)\right] \\
&= \mathbf{E}\left[\sum_{t \text{ is explore}} \hat{\ell}_t(i_t) + \sum_{t \text{ is exploit}} \hat{\ell}(i_t) - \min_{i\in[N]} \sum_{t=1}^{T} \hat{\ell}_t(i)\right]
\end{aligned}$$

since all of the "explore" steps may go horribly awry and incur the maximum loss of 1,

$$\begin{aligned}
&\leq \mathbf{E}\left[\text{num explore steps} + \sum_{t \text{ is exploit}} \hat{\ell}(i_t) - \min_{i\in[N]} \sum_{t=1}^{T} \hat{\ell}_t(i)\right] \\
&= \delta T + \mathbf{E}\left[\sum_{t \text{ is exploit}} \hat{\ell}(i_t) - \min_{i\in[N]} \sum_{t=1}^{T} \hat{\ell}_t(i)\right] \\
&\leq \delta T + \mathbf{E}\left[\sum_{t \text{ is exploit}} \hat{\ell}(i_t) - \min_{i\in[N]} \sum_{t \text{ is exploit}} \hat{\ell}_t(i)\right] \\
&\leq \delta T + \mathbf{E}\left[\text{regret of hedge algorithm after } T \text{ steps}\right] \\
&\leq \delta T + M\sqrt{8T\log N} \qquad \text{(OCO book page 82)}
\end{aligned}$$

where M is any upper bound on $\hat{\ell}_t(i)$. Since $\hat{\ell}_t(i) \leq \frac{N}{\delta} f_t(i) \leq \frac{N}{\delta}$, we have

$$\leq \delta T + \frac{1}{\delta}\sqrt{8TN^2\log N}$$

choosing $\delta = T^{-\frac{1}{4}}(N^2\log N)^{\frac{1}{4}}$

$$\begin{aligned}
&\leq (1+\sqrt{8})\, T^{\frac{3}{4}}(N^2\log N)^{\frac{1}{4}} \\
&= O(T^{\frac{3}{4}}\sqrt{N}(\log N)^{\frac{1}{4}})
\end{aligned}$$

1.3 Exp3

The simple algorithm presented above for the multi-armed bandit problem can be improved upon if we do away with the distinction between "explore" and "exploit" steps.

Algorithm 2 EXP3 Algorithm

Let $\mathbf{x}_1 = \frac{1}{N}\mathbf{1}$
for t=1 to T **do**
  choose $i_t \sim \mathbf{x}_t$ and play i_t
  set $\hat{\ell}_t(i) = \begin{cases} \frac{1}{\mathbf{x}_t(i_t)} f_t(i) & \text{if } i = i_t \\ 0 & \text{otherwise} \end{cases}$
  update $\mathbf{y}_{t+1}(i) = \mathbf{x}_t(i)\, e^{-\varepsilon \hat{\ell}_t(i)}$ $\qquad \mathbf{x}_{t+1} = \frac{\mathbf{y}_{t+1}}{||\mathbf{y}_{t+1}||_1}$

Theorem 1.3. *The regret of EXP3 is bounded by* $O(\sqrt{N \log N}\sqrt{T})$.

Is this the best regret bound one can attain for the bandit problem? No. In fact, the minimax rate for expected regret is $O(\sqrt{NT})$. In words, this means the following:

Theorem 1.4. *For the bandit problem:*

- *There exists an adversary that forces any algorithm to incur expected regret at least* $\Omega(\sqrt{NT})$.
- *There exists an algorithm that incurs at most* $O(\sqrt{NT})$ *expected regret against any adversary.*

1.4 Online Routing

One could model the online routing problem as a multi-armed bandit problem. Each of the N "arms" of the bandit is a path throughout the network; the loss function measures the time it takes a packet to travel along that path. This approach would work, but the number of paths throughout the network scales exponentially with the number of nodes. Can we do better?

Recall that the dimension of the flow polytope scales *polynomially* with the number of nodes in the network. If we could optimize over the flow polytope directly, we might obtain a better regret bound.

This motivates a more general setting, called **bandit convex optimization**, which is OCO minus the gradient.

2 Bandit Convex Optimization

In bandit convex optimization (BCO), as in online convex optimization, the player's goal is to play some $\mathbf{x}_t \in \mathcal{K}$ in the t-th round so as to minimize regret:

$$\text{Regret} = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x})$$

In BCO, however, the player is given far more limited feedback than in OCO: only a single number, $f_t(\mathbf{x}_t) \in \mathbb{R}$, instead of the whole loss function $f_t : \mathcal{K} \to \mathbb{R}$. Can we still do online learning without a gradient?

2.1 FKM Algorithm

The idea behind the FKM (Flaxman, Kalai, and McMahan) algorithm is to follow an unbiased estimator of the gradient.

For a function in one dimension $f : \mathbb{R} \to \mathbb{R}$, an unbiased estimator of the derivative is given by:

$$\tilde{f}'(x) = \begin{cases} \frac{1}{\delta} f(x+\delta) & \text{w.p. } \frac{1}{2} \\ -\frac{1}{\delta} f(x-\delta) & \text{w.p. } \frac{1}{2} \end{cases}$$

As $\delta \to 0$, the estimator $\tilde{f}'(x)$ tends toward $f'(x)$ in expectation:

$$\lim_{\delta \to 0} \mathbf{E}\left[\tilde{f}'(x)\right] = \lim_{\delta \to 0} \frac{f(x+\delta) - f(x-\delta)}{2\delta} = f'(x)$$

This approach also works in higher dimensions. For a function $f : \mathbb{R}^d \to \mathbb{R}$, an unbiased estimator of the gradient is given by

$$\tilde{\nabla} f_\delta(\mathbf{x}) = \frac{d}{\delta} f(\mathbf{x} + \delta \mathbf{v}) \mathbf{v} \quad \text{where} \quad \mathbf{v} \text{ drawn uniformly over } d\text{-dimensional unit sphere } \mathbb{S}_1$$

Stokes' theorem guarantees that $\tilde{\nabla} f_\delta(\mathbf{x})$ is an unbiased estimator of the gradient:

$$\lim_{\delta \to 0} \quad \mathbf{E}\left[\tilde{\nabla} f_\delta(\mathbf{x})\right] = \nabla f(\mathbf{x})$$

How can we cheaply sample a vector $\mathbf{v}$ uniformly over the unit sphere? (That is, $||\mathbf{v}|| = 1$ and all directions should be equally likely.) One way is to sample each element of $\mathbf{v}$ independently from the standard normal distribution, and then scale $\mathbf{v}$ to have unit norm.

Assume for simplicity that $\mathbf{0} \in \mathcal{K}$.

Algorithm 3 FKM Algorithm

set $\mathbf{y}_1 = \mathbf{0}$
for t = 1 to T **do**
  sample $\mathbf{u}_t \sim \mathbb{S}_1$
  set $\mathbf{x}_t = \mathbf{y}_t + \delta \mathbf{u}_t$ and play $\mathbf{x}_t$
  set $\mathbf{g}_t = \frac{d}{\delta} f_t(\mathbf{x}_t)\mathbf{u}_t$
  update $\mathbf{y}_{t+1} = \Pi_{\mathcal{K}_\delta}(\mathbf{y}_t - \eta\, \mathbf{g}_t)$

where $\mathcal{K}_\delta$ is defined as:

$$\mathcal{K}_\delta = \left\{ \mathbf{x} \in \mathbb{R}^d : \frac{\mathbf{x}}{1-\delta} \in \mathcal{K} \right\}$$

We project onto this "shrunken" decision set $\mathcal{K}_\delta$ instead of the original decision set $\mathcal{K}$ in order to ensure that $\mathbf{x}_t + \delta \mathbf{v}$ lies within the domain of $f_t : \mathcal{K} \to \mathbb{R}$.

Theorem 2.1. *The FKM algorithm attains a regret of* $O(d\, T^{\frac{3}{4}})$.

One can use the FKM algorithm for, say, the online routing problem, and the regret will scale polynomially, rather than exponentially, with the number of nodes in the network.