Machine Learning Theory (CS 6783)

Lecture 27: Contextual Bandit, Semi-Bandit

So far we have seen the Bandit problem, both the multi-armed bandit and linear bandit settings. In this lecture, we will briefly look at fancier versions of bandit problems. Specifically, we will look at a contextual version of bandit problem and whats called semi-bandit settings.

1 Contextual Bandit Problem

Multi-armed bandit problem is one where we have K actions or arms and each day we pull one and get losses based on what we pulled. This setting is great for very simple ad placement problem for instance. Example, we have K ads and we want a strategy to display ad that is more likely to be clicked. However, in more practical scenarios, we dont just place ad's without using other, "contextual" information. For instance, when placing ad, we use information like, what season it is, who the user is what is their history etc. The contextual bandit problem considers this more realistic scenario.

- For $t = 1$ to n
  - Nature produces context $x_t \in \mathcal{X}$
  - Algorithm picks arm $I_t \in [K]$ in a possibly randomized fashion while nature produces loss vector ℓ_t
  - Learner suffers loss $\ell_t[I_t]$

Goal: Minimize regret w.r.t. class of policies $\mathcal{F} \subset [K]^{\mathcal{X}}$ given by

$$\text{Reg}_n = \frac{1}{n}\sum_{t=1}^{n} \ell_t[I_t] - \inf_{f\in\mathcal{F}} \frac{1}{n}\sum_{t=1}^{n} \ell_t[f(x_t)]$$

That is, we want to do as well as the best policy in $\mathcal{F}$ which takes into account context x_t before picking an arm on each round. If $\mathcal{F}$ had just the K constant mappings of picking just each of the K arms ignoring context, then this problem is same as the multi-armed bandit one.

What is a strategy here? Well we could simply ignore the fact that there are only K arms/options on each round and just treat is as a multi-armed bandit problem over $|\mathcal{F}|$ arms (each policy is one arm). However in this case, our regret bounds would be at least $\sqrt{|\mathcal{F}|/n}$ and since we would like to consider rich class of policies $\mathcal{F}$ whose cardinality could possibly be very very large compared to n, this approach of plainly using the bandit algorithm would fail. However lets consider a more careful reuse of the exponential weights algorithm. Recall from the Bandits lecture that the key to getting the bandit result was to plug in unbiased estimates of losses into a more carefully analyzed full information algorithm. Let us have a second look at what we had there.

Changing notations from lecture 20 and rewriting the result from the lecture for exponential weights (mirror descent with entropy regularizer), we have that

$$n\text{Reg}_n(\tilde{\ell}_1, \ldots, \tilde{\ell}_n) \leq \frac{\eta}{2} \sum_{t=1}^{n} \sum_{f \in \mathcal{F}} \hat{y}_t[f] \tilde{\ell}_t[f]^2 + \frac{\log |\mathcal{F}|}{\eta}$$

where $\tilde{\ell}_1, \ldots, \tilde{\ell}_n$ are the losses over the experts in $\mathcal{F}$ for the n rounds that is fed to the full information algorithm. $\hat{y}_t$ is the distribution over the experts produced by the algorithm in round t. Now the main thing to note is that at round t, if we pick an expert $f \in \mathcal{F}$, we observe loss $\ell_t[f(x_t)]$. But this is loss of arm $f(x_t)$ and so in reality we observe loss of not just f for that round but rather all the other $f' \in \mathcal{F}$ that are such that $f'(x_t) = f(x_t)$. Lets make use of this information. Specifically, let us define our unbiased estimate of loss to be:

$$\tilde{\ell}_t[f] = \frac{1}{\sum_{f \in \mathcal{F}: f(x_t) = I_t} \hat{y}_t[f]} \mathbf{1}\{f(x_t) = I_t\} \ell_t[I_t]$$

Why is this an unbiased estimate? Well consider our algorithm. It draws first $g \sim \hat{y}_t$ and then uses action $I_t = g(x_t)$. Hence, for any $f \in \mathcal{F}$,

$$\begin{aligned}
\mathbb{E}_{g \sim \hat{y}_t}\left[\tilde{\ell}_t[f]\right] &= \mathbb{E}_{g \sim \hat{y}_t}\left[\frac{1}{\sum_{f \in \mathcal{F}: f(x_t) = g(x_t)} \hat{y}_t[f]} \mathbf{1}\{f(x_t) = g(x_t)\} \ell_t[g(x_t)]\right] \\
&= \sum_{g \in \mathcal{F}} \hat{y}_t[g] \frac{1}{\sum_{f \in \mathcal{F}: f(x_t) = g(x_t)} \hat{y}_t[f]} \mathbf{1}\{f(x_t) = g(x_t)\} \ell_t[g(x_t)] \\
&= \sum_{g \in \mathcal{F}: g(x_t) = f(x_t)} \hat{y}_t[g] \frac{1}{\sum_{f' \in \mathcal{F}: f'(x_t) = f(x_t)} \hat{y}_t[f']} \ell_t[f(x_t)] \\
&= \ell_t[f(x_t)] \frac{\sum_{g \in \mathcal{F}: g(x_t) = f(x_t)} \hat{y}_t[g]}{\sum_{f' \in \mathcal{F}: f'(x_t) = f(x_t)} \hat{y}_t[f']} \\
&= \ell_t[f(x_t)]
\end{aligned}$$

Hence if we use this new unbiased estimate, we have that:

$$\begin{aligned}
n\text{Reg}_n(\tilde{\ell}_1, \ldots, \tilde{\ell}_n) &\leq \frac{\eta}{2} \sum_{t=1}^{n} \sum_{f \in \mathcal{F}} \hat{y}_t[f] \left(\frac{1}{\sum_{f \in \mathcal{F}: f(x_t) = I_t} \hat{y}_t[f]} \mathbf{1}\{f(x_t) = I_t\} \ell_t[I_t]\right)^2 + \frac{\log |\mathcal{F}|}{\eta} \\
&= \frac{\eta}{2} \sum_{t=1}^{n} \sum_{f \in \mathcal{F}} \hat{y}_t[f] \left(\frac{1}{\sum_{f \in \mathcal{F}: f(x_t) = I_t} \hat{y}_t[f]}\right)^2 \mathbf{1}\{f(x_t) = I_t\} \ell_t^2[I_t] + \frac{\log |\mathcal{F}|}{\eta} \\
&= \frac{\eta}{2} \sum_{t=1}^{n} \left(\sum_{f \in \mathcal{F}: f(x_t) = I_t} \hat{y}_t[f]\right) \left(\frac{1}{\sum_{f \in \mathcal{F}: f(x_t) = I_t} \hat{y}_t[f]}\right)^2 \ell_t^2[I_t] + \frac{\log |\mathcal{F}|}{\eta} \\
&= \frac{\eta}{2} \sum_{t=1}^{n} \frac{\ell_t^2[I_t]}{\sum_{f \in \mathcal{F}: f(x_t) = I_t} \hat{y}_t[f]} + \frac{\log |\mathcal{F}|}{\eta}
\end{aligned}$$

Now taking expectation on both sides, we get:

$$\begin{aligned}
n\mathbb{E}\left[\mathrm{Reg}_n(\tilde{\ell}_1,\ldots,\tilde{\ell}_n)\right] &\leq \frac{\eta}{2}\sum_{t=1}^{n}\mathbb{E}_{I_t}\left[\frac{\ell_t^2[I_t]}{\sum_{f\in\mathcal{F}:f(x_t)=I_t}\hat{y}_t[f]}\right]+\frac{\log|\mathcal{F}|}{\eta}\\
&\leq \frac{\eta}{2}\sum_{t=1}^{n}\sum_{k=1}^{K}\left(\sum_{f\in\mathcal{F}:f(x_t)=k}\hat{y}_t[f]\right)\frac{\ell_t^2[k]}{\sum_{f\in\mathcal{F}:f(x_t)=k}\hat{y}_t[f]}+\frac{\log|\mathcal{F}|}{\eta}\\
&\leq \frac{\eta}{2}\sum_{t=1}^{n}\sum_{k=1}^{K}\ell_t^2[k]+\frac{\log|\mathcal{F}|}{\eta}\\
&\leq \frac{\eta K n}{2}+\frac{\log|\mathcal{F}|}{\eta}
\end{aligned}$$

Optimizing over η and recalling that expected regret of the bandit algorithm is upper bounded by expected regret of the full information algorithm we conclude that

$$\mathbb{E}\left[\mathrm{Reg}_n\right]\leq O\left(\sqrt{\frac{K\log|F|}{n}}\right)$$

Thus we can in fact get logarithmic dependence on the number of policies and a $\sqrt{K}$ dependence for regret bound. This algorithm is known as EXP4 algorithm. This algorithm does have the optimal regret bound up to constant factors.

1.1 Oracle Efficient Algorithms

A drawback of this algorithm though is that when $|\mathcal{F}|$ is very large, this algorithm is computationally inefficient. This is because, the algorithm maintains a distribution over $\mathcal{F}$ and the computation time per round is linear in $|\mathcal{F}|$. In general, this issue can be real. However, practically, to alleviate this issue, one might want to assume access to an ERM oracle. That is, given a sequence of instances, $x_1,\ell_1,\ldots,\ell_m,x_m$, we assume that we can efficiently compute the $f\in\mathcal{F}$ that minimizes empirical loss. That is an oracle that either exactly or approximately returns

$$\widehat{f}_{\mathrm{ERM}}=\operatorname*{argmin}_{f\in\mathcal{F}}\sum_{t=1}^{m}\ell_t[f(x_t)]$$

Now in general, assumption of access to such an ERM oracle may not mean we can minimize regret. However, if one assumes that context and loss are drawn from a fixed distribution, that is the iid stochastic setting, then it is not hard to see that one can achieve regret that goes to 0 and only has a poly-log dependence on $|\mathcal{F}|$. To see this, say n were known in advance and consider the algorithm that for first m rounds simply picks the uniform distribution over strategies and hence builds an unbiased estimate over loss vectors as $\hat{\ell}_t=Ke_{I_t}\ell_t[I_t]$, Now just as in multi-armed bandit setting, $(x_1,\hat{\ell}_1),\ldots,(x_m,\hat{\ell}_m)$ is an unbiased estimate of $(x_1,\ell_1),\ldots,(x_m,\ell_m)$. Now from Hoeffding Azuma inequality with union bound, one can argue that f_{ERM} obtained by running ERM oracle run on the estimate is order $\sqrt{K\log|F|/m}$ sub-optimal. Hence if we use this hypothesis for all future $n-m$ rounds we obtain the upper bound of order

$$\mathbb{E}\left[\mathrm{Reg}_n\right]\leq\frac{(n-m)}{n}\sqrt{K\log|F|/m}+\frac{m}{n}\leq\sqrt{K\log|F|/m}+\frac{m}{n}$$

Using $m = (K \log |F|)^{1/3} n^{2/3}$ we get, the final bound of order $O(K \log |F|/n)^{1/3}$. This algorithm is close in spirit to the so called epsilon-greedy algorithm which in every round explores uniformly with a small probability and explots by using ERM with remaining probability. This epsilon-greedy algorithm also attains the same upper bound as the one this above naive algorithm attains. It turns out that this regret bound is not optimal and one can in fact obtain the optimal rate of $\sqrt{K \log |F|/n}$ bound using the right algorithm. But we don't have the bandwidth to cover this algorithm in this course.

2 Semi-Bandit Problem

Consider the following scenario. For n days, we travel from home to office in the morning. Each day we can pick a route and our goal is to on average over the n days spend amount of time on road that is not much more than the best path from home to office. Now of course, one could think of this as a bandit problem where arms are paths and total time from home to office on that day is the loss for that path. However, this is not ideal for two reasons. First, number of paths from office to home could be prohibitively large and so the naive bandit bound might be too harsh. Second, and more importantly, when we take a route from home to office, we not only see the total delay on the route but also the delay on each road segment. Since paths might have overlapping road segments, there is more information that we might not be using to our advantage. Semi-bandits or combinatorial bandits are general problem setting that encapsulate such problems. Since online routing, the problem described above is the prototypical semi-bandit problem, we only discuss that problem here.

Input: Graph G = (V,E)

For $t = 1$ to n:

Learner picks a path represented by a subset $p_t \in \text{Paths} \subseteq \{0,1\}^E$

Adversary picks a cost $\ell_t \in [0,1]^E$ for every edge in the graph

Learner suffers linear loss $p_t^\top \ell_t$ and for every $e \in E$ s.t. $p_t[e] = 1$ observes $\ell_t[e]$

End for.

Goal: Minimize regret:

$$\text{Reg}_n = \frac{1}{n} \sum_{t=1}^{n} p_t^\top \ell_t - \min_{p \in \text{Paths}} \frac{1}{n} \sum_{t=1}^{n} p^\top \ell_t$$

To solve this problem, lets first try using exponential weights algorithm but with better unbiased estimate of losses. As expected, our set of experts is Paths and exponential weights algorithm will give us a distribution over paths to take. However, given a distribution $\hat{y}_t \in \Delta(\text{Paths})$, and a draw of a path $p_t \in \hat{y}_t$, for each edge $e \in E$, we can build an unbiased estimate of its specific loss (delay on that road segment) as:

$$\tilde{\ell}_t[e] = \frac{1}{\sum_{p \in \text{Paths: } p[e]=1} \hat{y}_t[p]} \mathbf{1}\{p_t[e] = 1\} \ell[e]$$

Clearly, for any $e \in E$, $\mathbb{E}_{p_t \sim \hat{y}_t}\left[\tilde{\ell}_t[e]\right] = \ell[e]$ since we are inversely weighing each edge by probability the edge is observed. Now, just like we have been doing, we can use this unbiased estimates along with reduction to full information algorithm, exponential weights in this case. To match the experts problem notation I will not overload $\tilde{\ell}$ notation. Specifically, we will use $\tilde{\ell}_t(p)$ to represent $\sum_{e \in E} p[e] \cdot \tilde{\ell}[e]$ That is:

$$
\begin{aligned}
n\mathbb{E}\left[\mathrm{Reg}_n\right] &= \mathbb{E}\left[\sum_{t=1}^{n} \mathbb{E}_{p \sim \hat{y}_t}\left[\tilde{\ell}_t(p)\right] - \min_{p \in \mathrm{Paths}} \sum_{t=1}^{n} \tilde{\ell}_t(p)\right] \\
&\leq \frac{\eta}{2} \sum_{t=1}^{n} \mathbb{E}_{p_t \sim \hat{y}_t}\left[\sum_{p \in \mathrm{Paths}} \hat{y}_t[p] \tilde{\ell}_t(p)^2\right] + \frac{\log|\mathrm{Paths}|}{\eta} \\
&\leq \frac{\eta}{2} \sum_{t=1}^{n} \mathbb{E}_{p_t \sim \hat{y}_t}\left[\sum_{p \in \mathrm{Paths}} \hat{y}_t[p] \left(\sum_{e \in E} p[e] \cdot \tilde{\ell}_t[e]\right)^2\right] + \frac{\log|\mathrm{Paths}|}{\eta} \\
&= \frac{\eta}{2} \sum_{t=1}^{n} \mathbb{E}_{p_t \sim \hat{y}_t}\left[\sum_{p \in \mathrm{Paths}} \hat{y}_t[p] \left(\sum_{e \in E} p[e] \cdot \frac{1}{\sum_{p \in \mathrm{Paths:}\ p[e]=1} \hat{y}_t[p]} \mathbf{1}\{p_t[e]=1\}\ell[e]\right)^2\right] + \frac{\log|\mathrm{Paths}|}{\eta} \\
&\leq \frac{\eta}{2} \sum_{t=1}^{n} \mathbb{E}_{p_t \sim \hat{y}_t}\left[\sum_{p \in \mathrm{Paths}} \hat{y}_t[p] \left(\sum_{e \in E: p_t[e]=1} \frac{p[e]}{\sum_{p \in \mathrm{Paths:}\ p[e]=1} \hat{y}_t[p]}\right)^2\right] + \frac{\log|\mathrm{Paths}|}{\eta} \\
&\leq \frac{\eta}{2} \sum_{t=1}^{n} \mathbb{E}_{p_t \sim \hat{y}_t}\left[\sum_{p \in \mathrm{Paths}} \frac{1}{\mathrm{length}(p_t)} \sum_{e \in E: p_t[e]=1} \hat{y}_t[p] \left(\mathrm{length}(p_t) \frac{p[e]}{\sum_{p \in \mathrm{Paths:}\ p[e]=1} \hat{y}_t[p]}\right)^2\right] + \frac{\log|\mathrm{Paths}|}{\eta} \\
&\leq \frac{\eta}{2} \sum_{t=1}^{n} \mathbb{E}_{p_t \sim \hat{y}_t}\left[\mathrm{length}(p_t) \cdot \sum_{p \in \mathrm{Paths}} \sum_{e \in E: p_t[e]=1} \hat{y}_t[p] \frac{p[e]}{\left(\sum_{p \in \mathrm{Paths:}\ p[e]=1} \hat{y}_t[p]\right)^2}\right] + \frac{\log|\mathrm{Paths}|}{\eta} \\
&= \frac{\eta}{2} \sum_{t=1}^{n} \mathbb{E}_{p_t \sim \hat{y}_t}\left[\mathrm{length}(p_t) \cdot \sum_{e \in E: p_t[e]=1} \sum_{p \in \mathrm{Paths}: p[e]=1} \hat{y}_t[p] \frac{1}{\left(\sum_{p \in \mathrm{Paths:}\ p[e]=1} \hat{y}_t[p]\right)^2}\right] + \frac{\log|\mathrm{Paths}|}{\eta} \\
&\leq \frac{\eta\ \max_{p \in \mathrm{Paths}} \mathrm{length}(p)}{2} \sum_{t=1}^{n} \mathbb{E}_{p_t \sim \hat{y}_t}\left[\sum_{e \in E: p_t[e]=1} \frac{1}{\sum_{p \in \mathrm{Paths:}\ p[e]=1} \hat{y}_t[p]}\right] + \frac{\log|\mathrm{Paths}|}{\eta} \\
&= \frac{\eta\ \max_{p \in \mathrm{Paths}} \mathrm{length}(p)}{2} \sum_{t=1}^{n} \sum_{q \in \mathrm{Paths}} \sum_{e \in E: q[e]=1} \hat{y}_t[q] \cdot \frac{1}{\sum_{p \in \mathrm{Paths:}\ p[e]=1} \hat{y}_t[p]} + \frac{\log|\mathrm{Paths}|}{\eta} \\
&= \frac{\eta\ \max_{p \in \mathrm{Paths}} \mathrm{length}(p)}{2} \sum_{t=1}^{n} \sum_{e \in E} \sum_{q \in \mathrm{Paths}: q[e]=1} \hat{y}_t[q] \cdot \frac{1}{\sum_{p \in \mathrm{Paths:}\ p[e]=1} \hat{y}_t[p]} + \frac{\log|\mathrm{Paths}|}{\eta}] \\
&= \frac{\eta\ \max_{p \in \mathrm{Paths}} \mathrm{length}(p)}{2} \sum_{t=1}^{n} \sum_{e \in E} 1 + \frac{\log|\mathrm{Paths}|}{\eta} \\
&= \frac{\eta\ \max_{p \in \mathrm{Paths}}\ \mathrm{length}(p)\ |E|\ n}{2} + \frac{\log|\mathrm{Paths}|}{\eta}
\end{aligned}
$$

Optimizing for η, we get,

$$\mathbb{E}\left[\mathrm{Reg}_n\right] \leq O\left(\sqrt{\frac{|E| \cdot \max_{p \in \mathrm{Paths}} \mathrm{length}(p) \cdot \log|\mathrm{Paths}|}{n}}\right)$$

This bound is much better, but we still have one caveat: We need to maintain a distribution over all paths and this is still computationally intensive. Just like for contextual bandits, we can also use the ERM oracle here. That is, given a sequence of delays on all road segment (full information) find the shortest path efficiently. But this is not really an assumption, we know algorithms for shortest path problem. The key idea to get an efficient algorithm for the semi-bandit problem is to replace exponential weights algorithm with another algorithm which is based on using ERM oracle and has the same guarantee:

$$n\mathbb{E}\left[\mathrm{Reg}_n\right] \leq \frac{\eta}{2}\sum_{t=1}^{n}\mathbb{E}_{p_t \sim \hat{y}_t}\left[\sum_{p \in \mathrm{Paths}} \hat{y}_t[p]\tilde{\ell}_t(p)^2\right] + \frac{|E|\log|\mathrm{Paths}|}{\eta}$$

Luckily for us, there is such an algorithm, called follow the perturbed leader that does exactly this. The algorithm is as simple as the following. At time t, for each edge the delay is the cumulative delay (based on estimates) for that edge so far, plus a random delay drawn according to exponential distribution with appropriate parameters. That is, on every round we simply solve a shortest path problem with delays given by past delays plus a random perturbation. This algorithm enjoys the type of bound mentioned above. In fact, if one replaces the exponential distribution for perturbation by the so called Gumbel distribution, the resulting algorithm in expectation is exactly exponential weights algorithm. In any case, its unfortunate that I wont be able to cover Follow The Perturbed Leader (FTPL) in class, it is rather beautiful! In any case, given that one case replace exponential weights by FTPL which can be implemented using ERM oracle and hence, one can obtain a computationally efficient algorithm as well.