NETS 412: Algorithmic Game Theory April 11, 2017

Lecture 20

Lecturer: Aaron Roth *Scribe: Aaron Roth*

Dynamic Pricing: Profit Maximization From "Bandit" Feedback

In the last lecture, we thought about running an auction in the "online" setting, in which buyers arrive one at a time, and report their valuations. We came up with a dominant strategy truthful allocation/payment rule that we thought of as offering fixed price "take it or leave it" offers, and we proved that we could compete with the best fixed price in hindsight. However, we weren't really offering fixed prices, in the sense that we still required buyers to report "bids" by specifying their valuation. In this lecture, we will think about one way to simplify the process so that we can actually just offer a fixed price at every round, which the buyer is free to take or leave. The difficulty will be that in this model, we will get only limited feedback at each round. If we offer some price p_i, and the offer is taken, we learn only that $v_i \geq p_i$. Similarly, if the offer is not taken, we learn only that $v_i < p_i$. In both cases, we lack the counter-factual information about what *would* have happened had we offered different prices, which is needed in order to update the polynomial weights algorithm.

To simplify things, we will assume in this lecture that the valuations of arriving buyers are not arbitrary, but are drawn from some unknown distribution (although similar results are possible without this assumption).

Definition 1 *In a dynamic pricing setting, there are n buyers, each with valuation $v_i \in [0,1]$ drawn independently from some unknown distribution $\mathcal{D}$.*

1. *At time t, the seller sets some price $p_t \in [0,1]$.*
2. *Buyer t arrives with $v_t \sim \mathcal{D}$. If $v_t \geq p_t$, the buyer purchases the good, and the seller gets revenue p_t. Otherwise, the buyer declines to purchase the good, and the seller gets revenue 0.*

Our goal is to dynamically set prices so as to obtain revenue competitive with the best fixed price. In this case, since we have assumed buyers are drawn from a distribution, this is: $\text{OPT} = \max_p p \cdot \Pr[v \geq p] \cdot n$. Just as we used the polynomial weights algorithm as our work-horse in the last lecture, to solve this problem, we will design a general technique for obtaining no-regret solutions in a generic "bandit feedback[1]" setting.

Definition 2 *In the multi-armed bandit problem, there are k "arms" i, each of which is associated with a payoff distribution $\mathcal{D}_i$ over $[0,1]$ with mean μ_i. In rounds t, the algorithm chooses arm i_t and receives reward $r^t_{i_t} \sim \mathcal{D}_i$.*

The expected reward of the algorithm after T days is $\sum_{t=1}^{T} \mu_{i_t}$. The regret *of the algorithm is:*

$$Regret(T) = T \cdot \mu_{i^*} - \sum_{t=1}^{T} \mu_{i_t}$$

where $i^* = \arg\max_i \mu_i$ is the arm with highest expected reward.

The idea will be to be "optimistic in the face of uncertainty". The algorithm we propose will quantify its uncertainty about the mean payoff of each arm i, by maintaining a confidence interval around its empirical estimate. It will then behave greedily – but not by playing the arm with the highest empirical mean so far, but rather by playing the arm with the highest *upper confidence bound*. Think of this as being optimistic – pretending that each arm is as good as it could possibly be, consistent with the evidence – and then playing greedily. Before we describe the algorithm, we will recall a useful fact about how to compute confidence intervals.

[1]The problem we are studying is colloquially called the "multi-armed bandit problem". The terminology comes from Vegas – a slot machine is a "one-armed bandit". In the multi-armed bandit problem, imagine a slot-machine with k arms, each with a different reward distribution. The goal is to find a policy for pulling the arms that is competitive with pulling the best fixed arm in hindsight.

Theorem 3 (Chernoff-Hoeffding Bound) *Let $\mathcal{D}$ be any distribution over $[0,1]$ with mean μ, and let $X_1, \ldots, X_n \sim \mathcal{D}$ be independent draws. Then for any $0 \leq \delta \leq 1$:*

$$\Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| \leq \sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}\right] \geq 1 - \delta$$

We are now ready to describe the algorithm.

UCB(δ, T):

Define $w(n) = \sqrt{\frac{\ln\left(\frac{2T}{\delta}\right)}{2n}}$. Initialize empirical means $\hat{\mu}_i^0 \leftarrow 1/2$ and upper and lower confidence bounds $u_i^0 \leftarrow 1, \ell_i^0 \leftarrow 0$ for each arm i. Initialize play counts $n_i^t \leftarrow 0$ for each arm i.
for $t = 1$ to T **do**
Pick an arm $i_t \in \arg\max u_i^{t-1}$. Observe reward $r_{i_t}^t$.
Update: For each $i \neq i_t$, set $(\hat{\mu}_i^t, u_i^t, \ell_i^t, n_i^t) \leftarrow (\hat{\mu}_i^{t-1}, u_i^{t-1}, \ell_i^{t-1}, n_i^{t-1})$
For $i = i_t$, $n_i^t \leftarrow n_i^{t-1} + 1$, $\hat{\mu}_i^t \leftarrow \frac{n_i^t - 1}{n_i^t}\hat{\mu}_i^{t-1} + \frac{1}{n_i^t} r_i^t, u_i^t \leftarrow \hat{\mu}_i^t + w(n_i^t), \ell_i^t \leftarrow \hat{\mu}_i^t - w(n_i^t)$
end for

Theorem 4 *For any set of k arms, with probability $1 - \delta$, the UCB algorithm obtains regret:*

$$\mathit{Regret}(T) \leq O\left(\sqrt{k \cdot T \cdot \ln\left(\frac{T}{\delta}\right)}\right)$$

Proof We start by observing that the widths of the confidence intervals w maintained by the UCB algorithm are defined such that by a Chernoff-Hoeffding bound, for each t and i, with probability $1-\delta/T$, $\mu_i \in [u_i^t, \ell_i^t]$. Since there are T confidence intervals constructed over the run of the algorithm, by a union bound we know that with probability $1 - \delta$, simultaneously for all i and t, $\mu_i \in [u_i^t, \ell_i^t]$. For the rest of the argument, we will assume that this is the case.

Now suppose at day t we play action i_t, obtaining expected payoff μ_{i_t}. How much worse is this than μ_{i^*}, the expected payoff of the optimal arm? Since by definition $i_t = \arg\max_i u_i^{t-1}$, and because all of the confidence intervals are valid, we have:

$$\mu_{i_t} \geq \ell_{i_t}^{t-1} = u_{i_t}^{t-1} - 2w(n_{i_t}^{t-1}) \geq u_{i^*}^{t-1} - 2w(n_{i_t}^{t-1}) \geq \mu_{i^*} - 2w(n_{i_t}^{t-1})$$

So the regret incurred at round t is at most $2w(n_{i_t}^{t-1})$. Thus, we can bound the total cumulative regret:

$$\begin{aligned}
Regret(T) &\leq 2\sum_{t=1}^{T} w(n_{i_t}^{t-1}) \\
&= 2\sum_{i=1}^{k}\sum_{n=1}^{n_i^T} w(n) \\
&\leq 2\sum_{i=1}^{k}\sum_{n=1}^{T/k} w(n) \\
&= 2\sum_{i=1}^{k}\sum_{n=1}^{T/k} \sqrt{\frac{\ln\left(\frac{2T}{\delta}\right)}{2n}} \\
&= 2\sum_{i=1}^{k} \sqrt{\frac{\ln\left(\frac{2T}{\delta}\right)}{2}} \sum_{n=1}^{T/k} \frac{1}{\sqrt{n}} \\
&\leq O\left(\sqrt{k \cdot T \cdot \ln\left(\frac{T}{\delta}\right)}\right)
\end{aligned}$$

■

Ok – so lets use this tool to design a revenue maximizing auction. We will pick a set k "arms", associating each one with a price from $K = \{\alpha, 2\alpha, 3\alpha, \ldots, 1\}$. Note that $k = |K| = 1/\alpha$. The distribution on rewards for each arm p is simply the distribution on revenue when deploying a price p – realizing reward $r_p = p$ with probability $\Pr[v \geq p]$ and reward $r_p = 0$ otherwise. Note also that because for every price $p \in [0, 1]$, there is another price $p' \in K$ such that $p - \alpha \leq p' \leq p$, in a setting with n buyers, we have:

$$\max_{p \in K} p \cdot \Pr[v \geq p] \cdot n \geq \max_{p \in [0,1]} p \cdot \Pr[v \geq p] \cdot n - \alpha n$$

Combining this guarantee with the guarantee of the UCB algorithm, we have that except with probability δ:

$$Revenue(UCB) \geq \max_{p \in K} p \cdot \Pr[v \geq p] \cdot n - O\left(\sqrt{k \cdot n \cdot \ln\left(\frac{n}{\delta}\right)}\right) \geq \text{OPT} - \alpha n - O\left(\sqrt{\frac{n}{\alpha} \cdot \ln\left(\frac{n}{\delta}\right)}\right)$$

Choosing

$$\alpha = \left(\frac{\log(n/\delta)}{n}\right)^{1/3}$$

yields:

$$Revenue(UCB) \geq \text{OPT} - O\left(n^{2/3} \log(n/\delta)^{1/3}\right)$$

What this means is that if $\text{OPT} = \omega\left(n^{2/3} \log(n/\delta)^{1/3}\right)$, then $Revenue(UCB) \geq (1 - o(1))\text{OPT}$. We would typically expect this to be the case, because if a constant fraction of buyers purchase the good, the revenue should grow linearly with n.