Dynamic Pricing: Profit Maximization From "Bandit" Feedback

Aaron Roth

University of Pennsylvania

April 20 2021

Overview

- Last lecture, we gave an online auction for maximizing revenue in digital goods settings.

Overview

- Last lecture, we gave an online auction for maximizing revenue in digital goods settings.
- But it was an "auction" rather than a "pricing scheme" because bidders had to report their valuations.

Overview

- Last lecture, we gave an online auction for maximizing revenue in digital goods settings.
- But it was an "auction" rather than a "pricing scheme" because bidders had to report their valuations.
- More practical/realistic if we just post prices and let buyers make purchase decisions.

Overview

- Last lecture, we gave an online auction for maximizing revenue in digital goods settings.
- But it was an "auction" rather than a "pricing scheme" because bidders had to report their valuations.
- More practical/realistic if we just post prices and let buyers make purchase decisions.
- But also more complex, because we don't get the feedback needed to run the polynomial weights algorithm.

Overview

- Last lecture, we gave an online auction for maximizing revenue in digital goods settings.
- But it was an "auction" rather than a "pricing scheme" because bidders had to report their valuations.
- More practical/realistic if we just post prices and let buyers make purchase decisions.
- But also more complex, because we don't get the feedback needed to run the polynomial weights algorithm.
- This lecture: solve this kind of "censored" learning problem when bidders are drawn from a distribution.

Overview

- Last lecture, we gave an online auction for maximizing revenue in digital goods settings.
- But it was an "auction" rather than a "pricing scheme" because bidders had to report their valuations.
- More practical/realistic if we just post prices and let buyers make purchase decisions.
- But also more complex, because we don't get the feedback needed to run the polynomial weights algorithm.
- This lecture: solve this kind of "censored" learning problem when bidders are drawn from a distribution.
- Its also possible to solve the problem without the distributional assumption... Just more complicated.

Dynamic Pricing

- We can offer fixed prices, and just observe whether buyers take or leave them. (Not their values).

Dynamic Pricing

- We can offer fixed prices, and just observe whether buyers take or leave them. (Not their values).
- We know nothing about the instance at the start, but learn as we go (and can change prices as we learn).

Dynamic Pricing

- We can offer fixed prices, and just observe whether buyers take or leave them. (Not their values).
- We know nothing about the instance at the start, but learn as we go (and can change prices as we learn).

Definition

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.

A Learning Approach

- We continue to want to compete with the bext fixed price benchmark:
$$\mathrm{OPT} = \max_p p \cdot \Pr[v \geq p] \cdot n$$

A Learning Approach

- We continue to want to compete with the bext fixed price benchmark:
$$\mathrm{OPT} = \max_p p \cdot \Pr[v \geq p] \cdot n$$
- Our approach last lecture was to reduce the problem to an online learning problem, and solve it using the PW algorithm.

A Learning Approach

- We continue to want to compete with the bext fixed price benchmark:
$$\mathrm{OPT} = \max_p p \cdot \Pr[v \geq p] \cdot n$$
- Our approach last lecture was to reduce the problem to an online learning problem, and solve it using the PW algorithm.
- We'll try and do the same thing this lecture. We need to define a learning problem with more restricted feedback.

Bandit Problems

Definition

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 $r_{i_t}^t \sim \mathcal{D}_i$.

Bandit Problems

Definition

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

- Idea: "optimism in the face of uncertainty".

The idea

- Idea: "optimism in the face of uncertainty".
- We will quantify uncertainty about the mean payoff of each arm i by maintaining a confidence interval around its empirical estimate.

The idea

- Idea: "optimism in the face of uncertainty".
- We will quantify uncertainty about the mean payoff of each arm i by maintaining a confidence interval around its empirical estimate.
- We 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*.

The idea

- Idea: "optimism in the face of uncertainty".
- We will quantify uncertainty about the mean payoff of each arm i by maintaining a confidence interval around its empirical estimate.
- We 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*.
- This is being optimistic – imagining that each arm is as good as it could possibly be, consistent with the evidence.

Confidence Intervals

Theorem (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 $0 \le \delta \le 1$:

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

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

Regret

Theorem

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

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

Proof

- Observe that the widths of the confidence intervals w maintained by the UCB algorithm are defined such that (by the Chernoff-Hoeffding bound): for each t and i, with probability $1 - \delta/T$:

$$\mu_i \in [u_i^t, \ell_i^t].$$

Proof

- Observe that the widths of the confidence intervals w maintained by the UCB algorithm are defined such that (by the 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, with probability $1 - \delta$, simultaneously for all i and t:

$$\mu_i \in [u_i^t, \ell_i^t].$$

Proof

- Observe that the widths of the confidence intervals w maintained by the UCB algorithm are defined such that (by the 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, 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.

Proof

- Suppose at day t we play action i_t, obtaining expected payoff μ_{i_t}.

Proof

- 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:

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

Proof

- 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:

$$\mu_{i^*} - \mu_{i_t} \leq 2w(n_{i_t}^{t-1})$$

Proof

- 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:

$$\mu_{i^*} - \mu_{i_t} \leq 2w(n_{i_t}^{t-1})$$

- Or see picture...

Proof

So we can bound overall regret as:

$$Regret(T) \quad \leq \quad 2\sum_{t=1}^{T} w(n_{i_t}^{t-1})$$

Proof

So we can bound overall regret as:

$$\begin{aligned} Regret(T) &\leq 2\sum_{t=1}^{T} w(n_{i_t}^{t-1}) \\ &= 2\sum_{i=1}^{k}\sum_{n=1}^{n_i^T} w(n) \end{aligned}$$

Proof

So we can bound overall regret as:

$$\begin{aligned} Regret(T) &\leq 2\sum_{t=1}^{T} w(n_{i_t}^{t-1}) \\ &= 2\sum_{i=1}^{k}\sum_{n=1}^{n_i^T} w(n) \\ &\leq 2\sum_{i=1}^{k}\sum_{n=1}^{T/k} w(n) \end{aligned}$$

Proof

So we can bound overall regret as:

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

Proof

So we can bound overall regret as:

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

Proof

So we can bound overall regret as:

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

Dynamic Pricing

- We will pick a set k "arms", associating each one with a price from $K = \{\alpha, 2\alpha, 3\alpha, \ldots, 1\}$.

Dynamic Pricing

- 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.

Dynamic Pricing

- 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.
- For every price $p \in [0, 1]$, there is another price $p' \in K$ such that $p - \alpha \leq p' \leq p$.

Dynamic Pricing

- 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.
- For every price $p \in [0, 1]$, there is another price $p' \in K$ such that $p - \alpha \leq p' \leq p$.
- So in a setting with n buyers, we have:

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

Dynamic Pricing

- Using the guarantees of the UCB algorithm we have that except with probability δ:

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

$$\geq \mathrm{OPT} - \alpha n - O\left(\sqrt{\frac{n}{\alpha} \cdot \ln\left(\frac{n}{\delta}\right)}\right)$$

Dynamic Pricing

- Using the guarantees of the UCB algorithm we have that except with probability δ:

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

- Choosing

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

yields:

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

Dynamic Pricing

- Using the guarantees of the UCB algorithm we have that except with probability δ:

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

$$\geq \mathrm{OPT} - \alpha n - O\left(\sqrt{\frac{n}{\alpha} \cdot \ln\left(\frac{n}{\delta}\right)}\right)$$

- Choosing

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

yields:

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

- So if $\mathrm{OPT}(n) = \omega\left(n^{2/3}\log(n/\delta)^{1/3}\right)$, then $Revenue(UCB) \geq (1 - o(1))\mathrm{OPT}$.

Dynamic Pricing

- Using the guarantees of the UCB algorithm we have that except with probability δ:

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

$$\geq \mathrm{OPT} - \alpha n - O\left(\sqrt{\frac{n}{\alpha} \cdot \ln\left(\frac{n}{\delta}\right)}\right)$$

- Choosing

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

yields:

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

- So if $\mathrm{OPT}(n) = \omega\left(n^{2/3} \log(n/\delta)^{1/3}\right)$, then $Revenue(UCB) \geq (1 - o(1))\mathrm{OPT}$.
- For any non-trivial distribution, this is the case (since $\mathrm{OPT}(n)$ grows linearly with n).

Thanks!

See you next class — stay healthy, and wear a mask!