Bandits: Bayesian Bandits and Thompson Sampling

Lucas Janson and Sham Kakade

CS/Stat 184: Introduction to Reinforcement Learning

Fall 2022

Today

- Feedback from last lecture
- Recap
- Instance-dependent regret of UCB
- Bayesian bandit
- Thompson sampling

Feedback from feedback forms

1. Thank you to everyone who filled out the forms!
2. Just a few people filled out form, and net zero on the pace
3. Added slide numbers

Today

- ✓ Feedback from last lecture
- Recap
- Instance-dependent regret of UCB
- Bayesian bandit
- Thompson sampling

Recap

- Pure greedy, pure exploration, ETC, ε-greedy achieve suboptimal worst-case regret
- UCB uses Optimism in the Face of Uncertainty (OFU) principle and achieves *optimal* rate $\tilde{O}(\sqrt{T})$ of worst-case regret
- Instance-dependent regret should be more informative than worst-case regret, but we haven’t actually bounded it for UCB yet

Today

- ✓ Feedback from last lecture
- ✓ Recap
- Instance-dependent regret of UCB
- Bayesian bandit
- Thompson sampling

Instance-dependent regret for UCB: strategy

(Reminder from last time)

1. Now that we can incorporate information about the $\mu^{(k)}$, we'll try to precisely bound how often each suboptimal arm k is sampled, $N_T^{(k)}$
2. To do that, we'll use the uniform Hoeffding bound to see how often the UCB for $k^\star$ is (with high probability) higher than the UCB for k
3. Then we'll multiply $N_T^{(k)}$ by the suboptimality of arm k, and sum this over the arms k to get the total regret

Instance-dependent regret for UCB

By uniform Hoeffding: w/p $\geq 1 - \delta$,

$\text{UCB}_t^{(k^\star)} \geq \mu^{(k^\star)} = \mu^\star$, and $\forall k$, $\text{UCB}_t^{(k)} := \hat{\mu}_t^{(k)} + \sqrt{\ln(2KT/\delta)/2N_t^{(k)}}$

$$:= \hat{\mu}_t^{(k)} + B_t^{(k)} \leq \mu^{(k)} + 2B_t^{(k)}$$

Denote $g_k := \mu^\star - \mu^{(k)}$ the *gap* between the best arm and arm k's mean

$\Rightarrow$ if $B_t^{(k)} < g_k/2$, then $\text{UCB}_t^{(k^\star)} > \text{UCB}_t^{(k)}$

When is $B_t^{(k)} < g_k/2$?

Instance-dependent regret for UCB (cont'd)

From last slide: w/p $\geq 1 - \delta$, $\forall t, k$ such that $N_t^{(k)} > 2\ln(2KT/\delta)/g_k^2$,

$\mathrm{UCB}_t^{(k^\star)} > \mathrm{UCB}_t^{(k)}$ (arm k not pulled at time t) $\Rightarrow$ $1_{\{a_t = k\}} = 0$

$$\mathrm{Regret}_T = \sum_{k=1}^{K} (\mu^\star - \mu^{(k)}) N_T^{(k)}$$

UCB regret with large g_k

$$\text{Regret}_T \leq \sum_{k=1}^{K} \frac{2\ln(2KT/\delta)}{g_k} \text{ w/p } \geq 1-\delta$$

Logarithmic in T: seems *much* better than worst-case lower-bound of $\Omega(\sqrt{T})$

But need to think about g_k to be sure

When all g_k are large relative to $\sqrt{1/T}$:

$$\sum_{k=1}^{K} \frac{2\ln(2KT/\delta)}{g_k} \leq K\frac{2\ln(2KT/\delta)}{\min_k g_k} \ll 2K\ln(2KT/\delta)\sqrt{T}$$

Instance-dependent bound indeed much better!

Idea: CLT says that with T steps, we'll easily find best arm if it's better by $\gg \sqrt{1/T}$ so basically we make relatively few mistakes

UCB regret with small g_k

If $\min_k g_k$ is much smaller than $\sqrt{1/T}$:

$$\sum_{k=1}^{K} \frac{2\ln(2KT/\delta)}{g_k} \geq \frac{2\ln(2KT/\delta)}{\min_k g_k} \gg 2\ln(2KT/\delta)\sqrt{T}$$

Way worse than worst-case upper-bound of $\tilde{O}(\sqrt{T})$...

But can match worst-case upper-bound by splitting arms into two groups:

$$\{k : g_k \leq \sqrt{1/T}\} \quad \text{and} \quad \{k : g_k > \sqrt{1/T}\}$$

$$\text{Regret}_T = \sum_{\{k:g_k\leq\sqrt{1/T}\}} g_k N_T^{(k)} + \sum_{\{k:g_k>\sqrt{1/T}\}} g_k N_T^{(k)}$$

UCB regret with VERY small g_k

Of course, if $\nu^{(1)} = \cdots = \nu^{(K)}$ and hence $\mu^{(1)} = \cdots = \mu^{(K)}$, then $\text{Regret}_T = 0\ldots$

neither bound is tight

$$\text{Regret}_T = \sum_{k=1}^{K} g_k N_t^{(k)} \le \max_k g_k \sum_{k=1}^{K} N_t^{(k)} = T \max_k g_k$$

Tighter than other bounds when $\max_k g_k \ll \frac{\ln(T)}{T}$, i.e., for small g_k and/or small T

Reasonable to expect Regret_T to scale like T times worst arm regret for *any algorithm* when it's too hard to distinguish the arms!

Summary: instance-dependent analysis gives more nuanced bounds on regret

Questions about UCB

1. Can we get rid of T in the algorithm so we don't have to know the time horizon?

   Yes: a more careful analysis allows to essentially replace T with t.

2. How to choose δ, since it impacts the algorithm and the regret bound?

   No satisfying answer that I know of to this.

3. What if we have prior information about the arms before collecting the data?

   There are heuristics for incorporating such information into UCB, but no single obvious and natural way to do so; Thompson sampling will though!

4. OFU principle seems reasonable, but why does it work?

   We will try to answer this today.

Today

- ✓ Feedback from last lecture
- ✓ Recap
- ✓ Instance-dependent regret of UCB
- Bayesian bandit
- Thompson sampling

Bayesian bandit

A Bayesian bandit augments the bandit environment we've been working in so far with a prior distribution on the unknown reward distributions: $\pi(\nu^{(1)}, \ldots, \nu^{(K)})$

E.g., in a Bernoulli bandit, each $\nu^{(k)}$ is entirely characterized by its mean $\mu^{(k)} = \mathbb{P}_{r \sim \nu^{(k)}}(r = 1)$, so a prior on the $\nu^{(k)}$ is equivalent to a prior on the $\mu^{(k)}$

One such prior, since all the $\mu^{(k)}$ are bounded between 0 and 1, is the prior that is *Uniform* on the unit hypercube, i.e., $(\mu^{(1)}, \ldots, \mu^{(K)}) =: \boldsymbol{\mu} \sim$ Uniform$([0,1]^K)$

Note that the Bernoulli bandit reduced everything unknown about the bandit system to a K-dimensional vector $\boldsymbol{\mu}$

Without the Bernoulli assumption, we may need many more dimensions to describe the possible distributions, and hence have to define a much higher-dimensional prior

Bayesian Bernoulli bandit

The really nice thing about a Bayesian bandit is that we can use Bayes rule to exactly characterize our uncertainty about the reward distributions at every time step.

Example: Bayesian Bernoulli bandit

1. At $t = 0$, we have no data, and the distribution of the reward distributions is simply given by the prior on the reward parameters $\boldsymbol{\mu}$:

$$\mathbb{P}(\boldsymbol{\mu}) = \pi(\boldsymbol{\mu})$$

($\mathbb{P}$ will sometimes denote a continuous density instead of a true probability, e.g., for $\boldsymbol{\mu} \sim$ Uniform$([0,1]^K)$, we would write $\mathbb{P}(\boldsymbol{\mu}) = 1_{\{0 \le \mu^{(k)} \le 1 \ \forall k\}}$)

Bayesian Bernoulli bandit (cont'd)

1. At $t = 0$, $\mathbb{P}(\boldsymbol{\mu}) = \pi(\boldsymbol{\mu})$
2. At $t = 1$, we have one data point $r_0 \sim$ Bernoulli($\mu^{(a_0)}$), and the distribution of $\boldsymbol{\mu}$ gets updated via Bayes rule:

Bayesian Bernoulli bandit (cont'd)

1. At $t = 0$, $\mathbb{P}(\boldsymbol{\mu}) = \pi(\boldsymbol{\mu})$
2. At $t = 1$, we have one data point $r_0 \sim$ Bernoulli$(\mu^{(a_0)})$, and the distribution of $\boldsymbol{\mu}$ gets updated via Bayes rule:

$$\mathbb{P}(\boldsymbol{\mu} \mid r_0, a_0) = \frac{\mathbb{P}(r_0 \mid a_0, \boldsymbol{\mu})\mathbb{P}(\boldsymbol{\mu})}{\int_{\tilde{\boldsymbol{\mu}} \in [0,1]^K} \mathbb{P}(r_0 \mid a_0, \tilde{\boldsymbol{\mu}})\mathbb{P}(\tilde{\boldsymbol{\mu}}) d\tilde{\boldsymbol{\mu}}}$$

Bayesian Bernoulli bandit (cont'd)

1. At $t = 0$, $\mathbb{P}(\boldsymbol{\mu}) = \pi(\boldsymbol{\mu})$
2. At $t = 1$, we have one data point $r_0 \sim$ Bernoulli($\mu^{(a_0)}$), and the distribution of $\boldsymbol{\mu}$ gets updated via Bayes rule:
$$\mathbb{P}(\boldsymbol{\mu} \mid r_0, a_0) = 2(\mu^{(a_0)})^{r_0}(1 - \mu^{(a_0)})^{1-r_0}$$
3. At $t = 2$, we have another data point $r_1 \sim$ Bernoulli($\mu^{(a_1)}$), and we can update the distribution of $\boldsymbol{\mu}$ again via Bayes rule, treating $\mathbb{P}(\boldsymbol{\mu} \mid r_0, a_0)$ as the prior

$\vdots$

Bayes rule at time step t gives us a distribution (called the posterior distribution)

$$\mathbb{P}(\boldsymbol{\mu} \mid r_0, a_0, r_1, a_1, \ldots, r_{t-1}, a_{t-1})$$

that exactly characterizes our uncertainty about $\boldsymbol{\mu}$. We can use this to choose a_t!

Today

- ✓ Feedback from last lecture
- ✓ Recap
- ✓ Instance-dependent regret of UCB
- ✓ Bayesian bandit
- Thompson sampling

Thompson sampling

Bayesian bandit environment means that at every time step, we know the distribution of the arm reward distributions conditioned on everything we've seen so far

In particular, we know the exact probability, given everything we've seen so far, that each arm is the true optimal arm, i.e.,

$\forall k,$ we know $\mathbb{P}(k = k^\star \mid r_0, a_0, \ldots, r_{t-1}, a_{t-1})$

<u>Thompson sampling</u>: sample from this distribution to determine next arm to pull

For $t = 0, \ldots, T-1$:

$a_t \sim$ distribution of $k^\star \mid r_0, a_0, \ldots, r_{t-1}, a_{t-1}$

(In practice, usually draw a sample $\boldsymbol{\mu}_t \sim$ distribution of $\boldsymbol{\mu} \mid r_0, a_0, \ldots, r_{t-1}, a_{t-1}$ and then compute $a_t = \arg\max_k \mu_t^{(k)}$, which is the same thing as $a_t \sim$ distribution of $k^\star \mid r_0, a_0, \ldots, r_{t-1}, a_{t-1}$)

That's it! Statistically, this is a super simple and elegant algorithm

(though computationally, it may not be easy to update the posterior at each time step)

Thompson sampling (cont'd)

Thompson sampling: $a_t \sim$ distribution of $k^\star \mid r_0, a_0, \ldots, r_{t-1}, a_{t-1}$

Why is this a good idea?

A good tradeoff of exploration vs exploitation should:

a) Sample the optimal arm as much as possible (duh)
b) Ensure arms that might still be optimal aren't overlooked
c) Not waste undue time on less promising arms

Intuitively: want to sample arms proportionally to how promising they are

This is exactly what Thompson sampling does, where "promising" is encoded very naturally as: "the probability that the arm is the optimal arm, given all the data so far"

No arbitrary δ tuning parameter, but do have to choose prior π

π can often be chosen "uninformatively" to a default prior such as the uniform, or can encode nuanced prior information/belief about the arms' reward distributions

Thompson sampling (cont’d)

Thompson sampling samples arms proportionally to how promising they are

Note this sampling is much more sophisticated than, say, ε-greedy, which really just samples according to 2 categories: “most promising” and “other”

But it’s also quite different from UCB, whose OFU approach doesn’t really involve “sampling” at all, i.e., every a_t for UCB is a *deterministic* function of the previous data

My interpretation: OFU provides a simple heuristic to accomplish what Thompson sampling does by design, namely, sample arms according to how promising they are

Thompson sampling can do this because of the Bayesian bandit: assuming a prior on the reward distributions makes the arm means random, otherwise it wouldn’t even make sense to talk about “the probability that an arm is the best arm”

Although derived from the Bayesian bandit, Thompson sampling has excellent practical performance across bandit problems, whether or not they are Bayesian!

Today

- ✓ Feedback from last lecture
- ✓ Recap
- ✓ Instance-dependent regret of UCB
- ✓ Bayesian bandit
- ✓ Thompson sampling

Today's summary:

Instance-dependent regret

- More descriptive than worst-case analysis
- UCB can do much better than worst-case $\Omega(\sqrt{T})$ regret in many cases

Bayesian bandit

- Adds an additional assumption of prior on reward distributions
- Bayes rule gives exact running uncertainty quantification for any algorithm

Thompson sampling

- Samples optimal arm from its (posterior) distribution
- Achieves excellent performance in practice

Next time:

- Gittins index

1-minute feedback form: https://bit.ly/3RHtlxy