Bandits: UCB Regret, Bayesian Bandits, and Thompson Sampling

Lucas Janson

CS/Stat 184(0): Introduction to Reinforcement Learning

Fall 2024

Bandits: UCB Regret, Bayesian Bandits, and Thompson Sampling

Lucas Janson

CS/Stat 184(0): Introduction to Reinforcement Learning

Fall 2024

Today

- Feedback from last lecture
- Recap
- UCB regret analysis
- Regret lower-bound
- Bayesian bandit
- Thompson sampling

Feedback from feedback forms

Feedback from feedback forms

1. Thank you to everyone who filled out the forms!

Feedback from feedback forms

1. Thank you to everyone who filled out the forms!

Today

- ✓ Feedback from last lecture
- Recap
- UCB regret analysis
- Regret lower-bound
- Bayesian bandit
- Thompson sampling

Recap

Recap

- Pure greedy and pure exploration achieve linear regret

Recap

- Pure greedy and pure exploration achieve linear regret
- Explore-then-commit (ETC) and ε-greedy:

Recap

- Pure greedy and pure exploration achieve linear regret
- Explore-then-commit (ETC) and ε-greedy:
  - balance exploration with exploitation

Recap

- Pure greedy and pure exploration achieve linear regret
- Explore-then-commit (ETC) and ε-greedy:
  - balance exploration with exploitation
  - Achieve sublinear regret of $\tilde{O}(T^{2/3})$

Recap

- Pure greedy and pure exploration achieve linear regret
- Explore-then-commit (ETC) and ε-greedy:
    - balance exploration with exploitation
    - Achieve sublinear regret of $\tilde{O}(T^{2/3})$
    - Exploration is non-adaptive

Recap

- Pure greedy and pure exploration achieve linear regret
- Explore-then-commit (ETC) and ε-greedy:
    - balance exploration with exploitation
    - Achieve sublinear regret of $\tilde{O}(T^{2/3})$
    - Exploration is non-adaptive
- Today: UCB does better than a rate of $T^{2/3}$

Recap of UCB

Recap of UCB

Notation: $N_t^{(k)} = \sum_{\tau=0}^{t-1} 1_{\{a_\tau = k\}}$ and $\hat{\mu}_t^{(k)} = \frac{1}{N_t^{(k)}} \sum_{\tau=0}^{t-1} 1_{\{a_\tau = k\}} r_\tau$

Recap of UCB

Notation: $N_t^{(k)} = \sum_{\tau=0}^{t-1} 1_{\{a_\tau = k\}}$ and $\hat{\mu}_t^{(k)} = \frac{1}{N_t^{(k)}} \sum_{\tau=0}^{t-1} 1_{\{a_\tau = k\}} r_\tau$

Any-algorithm time-and-arm-uniform error bounds for arm mean estimates:

$$\mathbb{P}\left(\forall k \leq K, t < T, \, |\hat{\mu}_t^{(k)} - \mu^{(k)}| \leq \sqrt{\ln(2TK/\delta)/2N_t^{(k)}} \right) \geq 1 - \delta$$

Recap of UCB

Notation: $N_t^{(k)} = \sum_{\tau=0}^{t-1} 1_{\{a_\tau = k\}}$ and $\hat{\mu}_t^{(k)} = \frac{1}{N_t^{(k)}} \sum_{\tau=0}^{t-1} 1_{\{a_\tau = k\}} r_\tau$

Any-algorithm time-and-arm-uniform error bounds for arm mean estimates:

$$\mathbb{P}\left(\forall k \leq K, t < T, \, |\hat{\mu}_t^{(k)} - \mu^{(k)}| \leq \sqrt{\ln(2TK/\delta)/2N_t^{(k)}} \right) \geq 1 - \delta$$

UCB Algorithm:

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

Choose the arm with the highest upper confidence bound, i.e.,

$$a_t = \arg \max_{k \in \{1, \ldots, K\}} \hat{\mu}_t^{(k)} + \sqrt{\ln(2TK/\delta)/2N_t^{(k)}}$$

Today

- Feedback from last lecture
- Recap
- UCB regret analysis
- Regret lower-bound
- Bayesian bandit
- Thompson sampling

UCB Regret Analysis Strategy

UCB Regret Analysis Strategy

1. Bound regret at each time step

UCB Regret Analysis Strategy

1. Bound regret at each time step
2. Bound the sum of those bounds over time steps

UCB regret at each time step

Recall $k^\star$ is optimal arm, so $\mu^{(k^\star)}$ is true best arm mean. Thus time step t regret is:

UCB regret at each time step

Recall $k^\star$ is optimal arm, so $\mu^{(k^\star)}$ is true best arm mean. Thus time step t regret is:

$\mu^{(k^\star)} - \mu^{(a_t)} \leq \hat{\mu}_t^{(k^\star)} + \sqrt{\ln(2KT/\delta)/2N_t^{(k^\star)}} - \mu^{(a_t)}$ (CI coverage on arm $k^\star$)

UCB regret at each time step

Recall $k^\star$ is optimal arm, so $\mu^{(k^\star)}$ is true best arm mean. Thus time step t regret is:

$\mu^{(k^\star)} - \mu^{(a_t)} \leq \hat{\mu}_t^{(k^\star)} + \sqrt{\ln(2KT/\delta)/2N_t^{(k^\star)}} - \mu^{(a_t)}$ (CI coverage on arm $k^\star$)   Next step?

UCB regret at each time step

Recall $k^\star$ is optimal arm, so $\mu^{(k^\star)}$ is true best arm mean. Thus time step t regret is:

$$\mu^{(k^\star)} - \mu^{(a_t)} \leq \hat{\mu}_t^{(k^\star)} + \sqrt{\ln(2KT/\delta)/2N_t^{(k^\star)}} - \mu^{(a_t)}$$ (CI coverage on arm $k^\star$)

Next step?

$$\leq \hat{\mu}_t^{(a_t)} + \sqrt{\ln(2KT/\delta)/2N_t^{(a_t)}} - \mu^{(a_t)}$$ (a_t maximizes UCB by definition)

UCB regret at each time step

Recall $k^\star$ is optimal arm, so $\mu^{(k^\star)}$ is true best arm mean. Thus time step t regret is:

$\mu^{(k^\star)} - \mu^{(a_t)} \leq \hat{\mu}_t^{(k^\star)} + \sqrt{\ln(2KT/\delta)/2N_t^{(k^\star)}} - \mu^{(a_t)}$ (CI coverage on arm $k^\star$) Next step?

$\leq \hat{\mu}_t^{(a_t)} + \sqrt{\ln(2KT/\delta)/2N_t^{(a_t)}} - \mu^{(a_t)}$ (a_t maximizes UCB by definition)

$\leq \sqrt{\ln(2KT/\delta)/2N_t^{(a_t)}} + \sqrt{\ln(2KT/\delta)/2N_t^{(a_t)}}$ (CI coverage on arm a_t)

UCB regret at each time step

Recall $k^\star$ is optimal arm, so $\mu^{(k^\star)}$ is true best arm mean. Thus time step t regret is:

$\mu^{(k^\star)} - \mu^{(a_t)} \leq \hat{\mu}_t^{(k^\star)} + \sqrt{\ln(2KT/\delta)/2N_t^{(k^\star)}} - \mu^{(a_t)}$ (CI coverage on arm $k^\star$) Next step?

$\leq \hat{\mu}_t^{(a_t)} + \sqrt{\ln(2KT/\delta)/2N_t^{(a_t)}} - \mu^{(a_t)}$ (a_t maximizes UCB by definition)

$\leq \sqrt{\ln(2KT/\delta)/2N_t^{(a_t)}} + \sqrt{\ln(2KT/\delta)/2N_t^{(a_t)}}$ (CI coverage on arm a_t)

$= \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}}$

UCB regret at each time step

Recall $k^\star$ is optimal arm, so $\mu^{(k^\star)}$ is true best arm mean. Thus time step t regret is:

$$\mu^{(k^\star)} - \mu^{(a_t)} \leq \hat{\mu}_t^{(k^\star)} + \sqrt{\ln(2KT/\delta)/2N_t^{(k^\star)}} - \mu^{(a_t)}$$ (CI coverage on arm $k^\star$)

Next step?

$$\leq \hat{\mu}_t^{(a_t)} + \sqrt{\ln(2KT/\delta)/2N_t^{(a_t)}} - \mu^{(a_t)}$$ (a_t maximizes UCB by definition)

$$\leq \sqrt{\ln(2KT/\delta)/2N_t^{(a_t)}} + \sqrt{\ln(2KT/\delta)/2N_t^{(a_t)}}$$ (CI coverage on arm a_t)

$$= \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}}$$

all lines above hold simultaneously for all t w/p $1-\delta$ because of *uniform* Hoeffding

Sum of UCB per-time-step regrets

1. per-time-step regret bound $\mu^{(k^\star)} - \mu^{(a_t)} \leq \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}}$ w/p $1-\delta$

2.

Sum of UCB per-time-step regrets

1. per-time-step regret bound $\mu^{(k^\star)} - \mu^{(a_t)} \leq \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}}$ w/p $1-\delta$

2. $$\text{Regret}_T \leq \sum_{t=0}^{T-1} \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}} = \sqrt{2\ln(2KT/\delta)} \sum_{t=0}^{T-1} \sqrt{\frac{1}{N_t^{(a_t)}}} \quad \text{w/p } 1-\delta$$

Sum of UCB per-time-step regrets

1. per-time-step regret bound $\mu^{(k^\star)} - \mu^{(a_t)} \leq \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}}$ w/p $1-\delta$

2. $$\text{Regret}_T \leq \sum_{t=0}^{T-1} \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}} = \sqrt{2\ln(2KT/\delta)} \sum_{t=0}^{T-1} \sqrt{\frac{1}{N_t^{(a_t)}}} \quad \text{w/p } 1-\delta$$

$$\sum_{t=0}^{T-1} \sqrt{\frac{1}{N_t^{(a_t)}}} = \sum_{t=0}^{T-1} \sum_{k=1}^{K} 1_{\{a_t=k\}} \sqrt{\frac{1}{N_t^{(k)}}}$$

Sum of UCB per-time-step regrets

1. per-time-step regret bound $\mu^{(k^\star)} - \mu^{(a_t)} \leq \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}}$ w/p $1-\delta$

2. $\text{Regret}_T \leq \sum_{t=0}^{T-1} \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}} = \sqrt{2\ln(2KT/\delta)} \sum_{t=0}^{T-1} \sqrt{\frac{1}{N_t^{(a_t)}}}$ w/p $1-\delta$

$$\sum_{t=0}^{T-1} \sqrt{\frac{1}{N_t^{(a_t)}}} = \sum_{t=0}^{T-1} \sum_{k=1}^{K} 1_{\{a_t=k\}} \sqrt{\frac{1}{N_t^{(k)}}} = \sum_{k=1}^{K} \sum_{n=1}^{N_T^{(k)}-1} \sqrt{\frac{1}{n}}$$

Sum of UCB per-time-step regrets

1. per-time-step regret bound $\mu^{(k^\star)} - \mu^{(a_t)} \leq \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}}$ w/p $1 - \delta$

2. $$\text{Regret}_T \leq \sum_{t=0}^{T-1} \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}} = \sqrt{2\ln(2KT/\delta)} \sum_{t=0}^{T-1} \sqrt{\frac{1}{N_t^{(a_t)}}}$$ w/p $1 - \delta$

$$\sum_{t=0}^{T-1} \sqrt{\frac{1}{N_t^{(a_t)}}} = \sum_{t=0}^{T-1} \sum_{k=1}^{K} 1_{\{a_t = k\}} \sqrt{\frac{1}{N_t^{(k)}}} = \sum_{k=1}^{K} \sum_{n=1}^{N_T^{(k)}} \sqrt{\frac{1}{n}} \leq K \sum_{n=1}^{T} \sqrt{\frac{1}{n}}$$

Sum of UCB per-time-step regrets

1. per-time-step regret bound $\mu^{(k^\star)} - \mu^{(a_t)} \leq \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}}$ w/p $1-\delta$

2. $$\text{Regret}_T \leq \sum_{t=0}^{T-1} \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}} = \sqrt{2\ln(2KT/\delta)} \sum_{t=0}^{T-1} \sqrt{\frac{1}{N_t^{(a_t)}}}$$ w/p $1-\delta$

$$\sum_{t=0}^{T-1} \sqrt{\frac{1}{N_t^{(a_t)}}} = \sum_{t=0}^{T-1} \sum_{k=1}^{K} 1_{\{a_t=k\}} \sqrt{\frac{1}{N_t^{(k)}}} = \sum_{k=1}^{K} \sum_{n=1}^{N_T^{(k)}} \sqrt{\frac{1}{n}} \leq K \sum_{n=1}^{T} \sqrt{\frac{1}{n}} \leq 2K\sqrt{T}$$

Sum of UCB per-time-step regrets

time steps $0, \ldots, K-1$, regret bndd by K

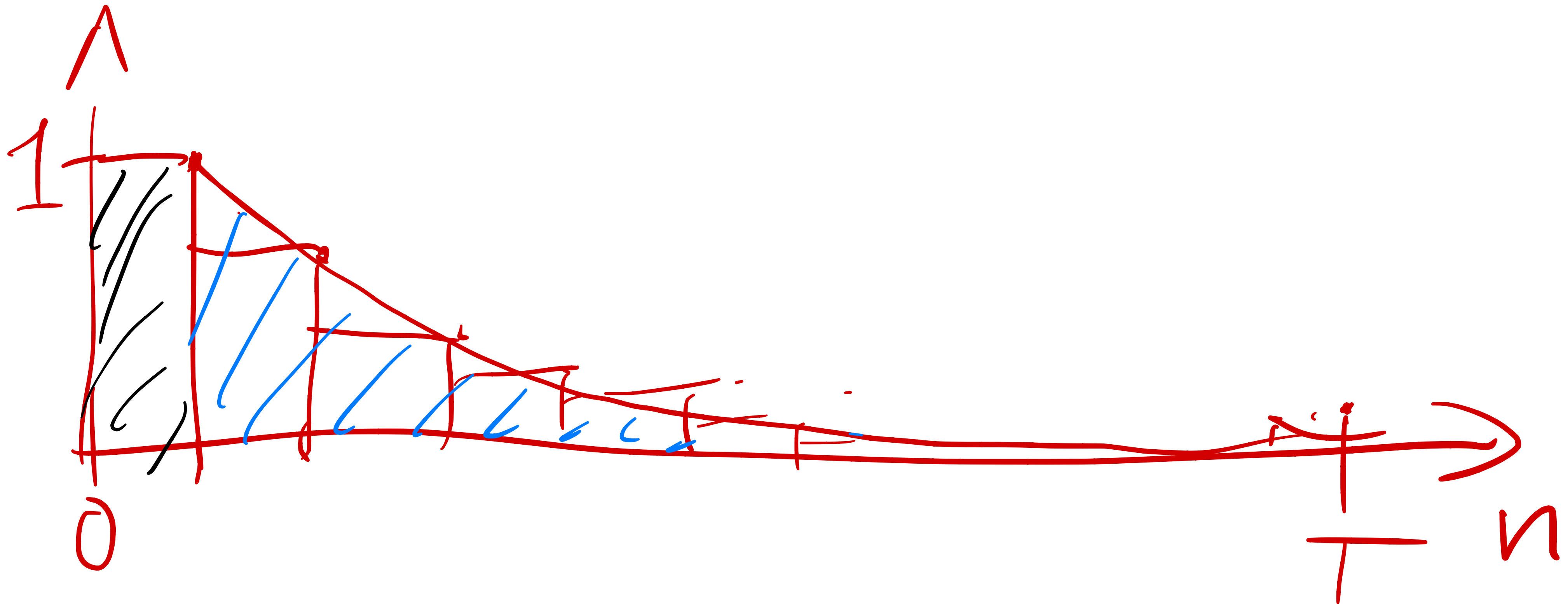
rest of this bounds regret for alg's times $K, \ldots, T-1$

1. per-time-step regret bound $\mu^{(k^\star)} - \mu^{(a_t)} \le \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}}$ w/p $1-\delta$

2. $$\text{Regret}_T \le \sum_{t=0}^{T-1} \sqrt{2\ln(2KT/\delta)/N_t^{(a_t)}} = \sqrt{2\ln(2KT/\delta)} \sum_{t=0}^{T-1} \sqrt{\frac{1}{N_t^{(a_t)}}} \quad \text{w/p } 1-\delta$$

$$\sum_{t=0}^{T-1} \sqrt{\frac{1}{N_t^{(a_t)}}} = \sum_{t=0}^{T-1} \sum_{k=1}^{K} 1_{\{a_t=k\}} \sqrt{\frac{1}{N_t^{(k)}}} = \sum_{k=1}^{K} \sum_{n=1}^{N_T^{(k)}} \sqrt{\frac{1}{n}} \le K \sum_{n=1}^{T} \sqrt{\frac{1}{n}} \le 2K\sqrt{T}$$

$$\sum_{n=1}^{T} \frac{1}{\sqrt{n}} \le 1 + \int_1^T \frac{1}{\sqrt{x}}\,dx = 1 + 2\sqrt{x}\,\Big|_{x=1}^{x=T} = 2\sqrt{T}$$



UCB total regret

UCB total regret

Finally, putting it all together, we get:

$$\text{Regret}_T \leq 2K\sqrt{T}\sqrt{2\ln(KT/\delta)} \qquad \text{w/p } 1 - \delta$$

UCB total regret

Finally, putting it all together, we get:

$$\text{Regret}_T \leq 2K\sqrt{T}\sqrt{2\ln(KT/\delta)} \quad \text{w/p } 1-\delta$$

$$= \tilde{O}(\sqrt{T}) \quad \text{w/p } 1-\delta$$

UCB total regret

Finally, putting it all together, we get:

$$\text{Regret}_T \leq 2K\sqrt{T}\sqrt{2\ln(KT/\delta)} \quad \text{w/p } 1-\delta$$

$$= \tilde{O}(\sqrt{T}) \quad \text{w/p } 1-\delta$$

In fact, a more sophisticated analysis can get: $\text{Regret}_T = \tilde{O}(\sqrt{KT}) \quad \text{w/p } 1-\delta$

Today

- Feedback from last lecture
- Recap
- UCB regret analysis
- Regret lower-bound
- Bayesian bandit
- Thompson sampling

Can we do better than $\Omega(\sqrt{T})$ regret?

Can we do better than $\Omega(\sqrt{T})$ regret?

Short answer: no

Can we do better than $\Omega(\sqrt{T})$ regret?

Short answer: no

But how can we know that?

Can we do better than $\Omega(\sqrt{T})$ regret?

Short answer: no

But how can we know that?

A *lower bound* on the achievable regret

Can we do better than $\Omega(\sqrt{T})$ regret?

Short answer: no

But how can we know that?

A *lower bound* on the achievable regret

So far we our theoretical analysis has always considered a fixed algorithm and analyzed it (by deriving a regret upper bound with high probability)

Can we do better than $\Omega(\sqrt{T})$ regret?

Short answer: no

But how can we know that?

A *lower bound* on the achievable regret

So far we our theoretical analysis has always considered a fixed algorithm and analyzed it (by deriving a regret upper bound with high probability)

To get a lower bound, we would need to consider what regret could be achieved by *any* algorithm, and show it can't be better than some rate

Intuition for lower bound

Intuition for lower bound

1. CLT tells us that with T i.i.d. samples from a distribution ν, we can only learn ν's mean μ to within $\Omega(1/\sqrt{T})$

Intuition for lower bound

1. CLT tells us that with T i.i.d. samples from a distribution ν, we can only learn ν's mean μ to within $\Omega(1/\sqrt{T})$
2. Then since in a bandit, we get at most T samples total, certainly we can't learn any of the arm means better than to within $\Omega(1/\sqrt{T})$

Intuition for lower bound

1. CLT tells us that with T i.i.d. samples from a distribution ν, we can only learn ν's mean μ to within $\Omega(1/\sqrt{T})$
2. Then since in a bandit, we get at most T samples total, certainly we can't learn any of the arm means better than to within $\Omega(1/\sqrt{T})$
3. This means that if an arm $\tilde{k}$ is about $1/\sqrt{T}$ away from the best arm $k^\star$, then at no point during the bandit can we confidently tell them apart

Intuition for lower bound

1. CLT tells us that with T i.i.d. samples from a distribution ν, we can only learn ν's mean μ to within $\Omega(1/\sqrt{T})$
2. Then since in a bandit, we get at most T samples total, certainly we can't learn any of the arm means better than to within $\Omega(1/\sqrt{T})$
3. This means that if an arm $\tilde{k}$ is about $1/\sqrt{T}$ away from the best arm $k^\star$, then at no point during the bandit can we confidently tell them apart
4. Thus, we should expect to sample $\tilde{k}$ roughly as often as $k^\star$, which is at best roughly $T/2$ times (if we ignore any other arms)

Intuition for lower bound

1. CLT tells us that with T i.i.d. samples from a distribution ν, we can only learn ν's mean μ to within $\Omega(1/\sqrt{T})$
2. Then since in a bandit, we get at most T samples total, certainly we can't learn any of the arm means better than to within $\Omega(1/\sqrt{T})$
3. This means that if an arm $\tilde{k}$ is about $1/\sqrt{T}$ away from the best arm $k^\star$, then at no point during the bandit can we confidently tell them apart
4. Thus, we should expect to sample $\tilde{k}$ roughly as often as $k^\star$, which is at best roughly $T/2$ times (if we ignore any other arms)
5. Finally, since the regret incurred each time we pull arm $\tilde{k}$ is $1/\sqrt{T}$, and we pull it $T/2$ times, we get a regret lower bound of $(1/\sqrt{T}) \times T/2 = \Omega(\sqrt{T})$

Today

- ✓ Feedback from last lecture
- ✓ Recap
- ✓ UCB regret analysis
- ✓ Regret lower-bound
- Bayesian bandit
- Thompson sampling

Bayesian bandit

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: $\pi(\nu^{(1)}, \ldots, \nu^{(K)})$

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 $\mu^{(k)}$

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.,
$(\mu^{(1)}, \ldots, \mu^{(K)}) =: \boldsymbol{\mu} \sim \text{Uniform}([0,1]^K)$

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: $\pi(\nu^{(1)}, \dots, \nu^{(K)})$

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., $(\mu^{(1)}, \dots, \mu^{(K)}) =: \boldsymbol{\mu} \sim \text{Uniform}([0,1]^K)$

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

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., $(\mu^{(1)}, \ldots, \mu^{(K)}) =: \boldsymbol{\mu} \sim \text{Uniform}([0,1]^K)$

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

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.

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

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 $t = 0$, how can we characterize our uncertainty about $\boldsymbol{\mu}$?

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 $t = 0$, how can we characterize our uncertainty about $\boldsymbol{\mu}$?
   We have no data, and the distribution of the reward distributions is simply given by the prior on the reward parameters $\boldsymbol{\mu}$:
   $$\mathbb{P}(\boldsymbol{\mu}) = \pi(\boldsymbol{\mu})$$

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 $t = 0$, how can we characterize our uncertainty about $\boldsymbol{\mu}$?
   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 $\boldsymbol{\mu} \sim \text{Uniform}([0,1]^K)$, we would write $\mathbb{P}(\boldsymbol{\mu}) = 1_{\{0 \le \mu^{(k)} \le 1 \ \forall k\}}$)

Bayesian Bernoulli bandit (cont'd)

1. At $t = 0$, $\mathbb{P}(\boldsymbol{\mu}) = \pi(\boldsymbol{\mu})$

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:

$$\mathbb{P}(\boldsymbol{\mu} \mid r_0, a_0) = \frac{\mathbb{P}(r_0, a_0 \mid \boldsymbol{\mu})\mathbb{P}(\boldsymbol{\mu})}{\mathbb{P}(r_0, a_0)}$$

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:

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

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:

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

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

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:

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

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

Can you see any way to simplify?

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:

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

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

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

Can you see any way to simplify?

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:

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

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

Can you see any way to simplify?

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

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:

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

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:

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

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:

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

If prior is Uniform$([0,1]^K)$, i.e., $\pi(\boldsymbol{\mu}) = 1 \;\; \forall \boldsymbol{\mu}$:

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

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:

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

If prior is Uniform($[0,1]^K$), i.e., $\pi(\boldsymbol{\mu}) = 1 \ \ \forall \boldsymbol{\mu}$:

$$= \frac{(\mu^{(a_0)})^{r_0}(1 - \mu^{(a_0)})^{1-r_0}}{\int_{\tilde{\boldsymbol{\mu}} \in [0,1]^K} (\tilde{\mu}^{(a_0)})^{r_0}(1 - \tilde{\mu}^{(a_0)})^{1-r_0} d\tilde{\boldsymbol{\mu}}} = \frac{(\mu^{(a_0)})^{r_0}(1 - \mu^{(a_0)})^{1-r_0}}{\int_0^1 (\tilde{\mu}^{(a_0)})^{r_0}(1 - \tilde{\mu}^{(a_0)})^{1-r_0} d\tilde{\mu}^{(a_0)}}$$

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 $r_0 \sim \text{Bernoulli}(\mu^{(a_0)})$, and the distribution of $\boldsymbol{\mu}$ gets updated via Bayes rule:

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

If prior is Uniform($[0,1]^K$), i.e., $\pi(\boldsymbol{\mu}) = 1 \;\; \forall \boldsymbol{\mu}$:

$$= \frac{(\mu^{(a_0)})^{r_0}(1 - \mu^{(a_0)})^{1-r_0}}{\int_{\tilde{\boldsymbol{\mu}} \in [0,1]^K} (\tilde{\mu}^{(a_0)})^{r_0}(1 - \tilde{\mu}^{(a_0)})^{1-r_0}d\tilde{\boldsymbol{\mu}}} = \frac{(\mu^{(a_0)})^{r_0}(1 - \mu^{(a_0)})^{1-r_0}}{\int_0^1 (\tilde{\mu}^{(a_0)})^{r_0}(1 - \tilde{\mu}^{(a_0)})^{1-r_0}d\tilde{\mu}^{(a_0)}} = 2(\mu^{(a_0)})^{r_0}(1 - \mu^{(a_0)})^{1-r_0}$$

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:

$$\mathbb{P}(\boldsymbol{\mu} \mid r_0, a_0) = 2(\mu^{(a_0)})^{r_0}(1 - \mu^{(a_0)})^{1-r_0}$$

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

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

$\vdots$

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 $\boldsymbol{\mu}$.

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!

Bayesian Bernoulli bandit (cont'd)

Bayesian Bernoulli bandit with uniform prior on $\boldsymbol{\mu}$ gives a running posterior on the mean of each arm k that is $\text{Beta}(1 + \#\{\text{arm } k \text{ successes}\}, 1 + \#\{\text{arm } k \text{ failures}\})$

Bayesian Bernoulli bandit (cont'd)

Bayesian Bernoulli bandit with uniform prior on $\boldsymbol{\mu}$ gives a running posterior on the mean of each arm k that is $\text{Beta}(1 + \#\{\text{arm } k \text{ successes}\}, 1 + \#\{\text{arm } k \text{ failures}\})$

(derived by Bayes rule and some algebra, see HW2)

Bayesian Bernoulli bandit (cont'd)

Bayesian Bernoulli bandit with uniform prior on $\boldsymbol{\mu}$ gives a running posterior on the mean of each arm k that is Beta$(1 + \#\{\text{arm } k \text{ successes}\}, 1 + \#\{\text{arm } k \text{ failures}\})$

(derived by Bayes rule and some algebra, see HW2)

Beta(α_k, β_k) has mean (posterior mean = what we *expect* $\mu^{(k)}$ to be):

$$\frac{\alpha_k}{\alpha_k + \beta_k} = \frac{1 + \#\{\text{arm } k \text{ successes}\}}{2 + \#\{\text{arm } k \text{ pulls}\}}$$

which starts at 1/2 and approaches the sample mean of arm k with more pulls.

Bayesian Bernoulli bandit (cont'd)

Bayesian Bernoulli bandit with uniform prior on $\boldsymbol{\mu}$ gives a running posterior on the mean of each arm k that is Beta$(1 + \#\{\text{arm } k \text{ successes}\}, 1 + \#\{\text{arm } k \text{ failures}\})$

(derived by Bayes rule and some algebra, see HW2)

Beta(α_k, β_k) has mean (posterior mean = what we *expect* $\mu^{(k)}$ to be):

$$\frac{\alpha_k}{\alpha_k + \beta_k} = \frac{1 + \#\{\text{arm } k \text{ successes}\}}{2 + \#\{\text{arm } k \text{ pulls}\}}$$

which starts at 1/2 and approaches the sample mean of arm k with more pulls.

Beta(α_k, β_k) has variance (posterior variance ≈ how uncertain we are about $\mu^{(k)}$):

$$\frac{\alpha_k}{\alpha_k + \beta_k} \times \frac{\beta_k}{\alpha_k + \beta_k} \times \frac{1}{\alpha_k + \beta_k + 1}$$

which decreases at a rate of roughly $1/\#\{\text{arm } k \text{ pulls}\}$

Bayesian bandit summary

A Bayesian bandit augments the bandit environment we've been working in so far with a prior distribution on the unknown reward distributions; for Bernoulli bandits, the reward distributions are entirely characterized by $\boldsymbol{\mu}$, so prior is: $\pi(\boldsymbol{\mu})$

Bayesian bandit summary

A Bayesian bandit augments the bandit environment we've been working in so far with a prior distribution on the unknown reward distributions; for Bernoulli bandits, the reward distributions are entirely characterized by $\boldsymbol{\mu}$, so prior is: $\pi(\boldsymbol{\mu})$

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 $\boldsymbol{\mu}$.

Bayesian bandit summary

A Bayesian bandit augments the bandit environment we've been working in so far with a prior distribution on the unknown reward distributions; for Bernoulli bandits, the reward distributions are entirely characterized by $\boldsymbol{\mu}$, so prior is: $\pi(\boldsymbol{\mu})$

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 $\boldsymbol{\mu}$.

Note that although we are now treating $\boldsymbol{\mu}$ as random, we still assume its value is only drawn once (from the prior) and then stays the same throughout t

Bayesian bandit summary

A Bayesian bandit augments the bandit environment we've been working in so far with a prior distribution on the unknown reward distributions; for Bernoulli bandits, the reward distributions are entirely characterized by $\boldsymbol{\mu}$, so prior is: $\pi(\boldsymbol{\mu})$

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 $\boldsymbol{\mu}$.

Note that although we are now treating $\boldsymbol{\mu}$ as random, we still assume its value is only drawn once (from the prior) and then stays the same throughout t

What changes with t is our information about $\boldsymbol{\mu}$, i.e., the posterior distribution, as we collect more and more data by pulling arms via a bandit algorithm

Today

- ✓ Feedback from last lecture
- ✓ Recap
- ✓ UCB regret analysis
- ✓ Regret lower-bound
- ✓ 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

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

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

Thompson sampling: sample from this distribution to determine next arm to pull

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

Thompson sampling: sample from this distribution to determine next arm to pull

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

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

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

How can we sample from this distribution?

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

How can we sample from this distribution? 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 $k^\star \mid r_0, a_0, \ldots, r_{t-1}, a_{t-1}$

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

How can we sample from this distribution? 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

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

How can we sample from this distribution? 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 intuition

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

Thompson sampling intuition

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?

Thompson sampling intuition

<u>Thompson sampling</u>: $a_t \sim$ 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:

Thompson sampling intuition

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)

Thompson sampling intuition

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

Thompson sampling intuition

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

Thompson sampling intuition

<u>Thompson sampling</u>: $a_t \sim$ 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

Thompson sampling intuition

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"

Thompson sampling intuition

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

Thompson sampling intuition

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 vs other algorithms

Thompson sampling samples arms proportionally to how promising they are

Thompson sampling vs other algorithms

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”

Thompson sampling vs other algorithms

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

Thompson sampling vs other algorithms

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 vs other algorithms

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”

Thompson sampling vs other algorithms

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!

Thompson sampling in practice

Thompson sampling in practice

Thompson sampling has excellent performance in practice, but is still just a heuristic

Thompson sampling in practice

Thompson sampling has excellent performance in practice, but is still just a heuristic

However, asymptotically, i.e., as $T \to \infty$, it actually is optimal in a certain sense

Thompson sampling in practice

Thompson sampling has excellent performance in practice, but is still just a heuristic

However, asymptotically, i.e., as $T \to \infty$, it actually is optimal in a certain sense

There is an *instance-dependent* lower-bound result that says that for any bandit algorithm:

$$\liminf_{T \to \infty} \frac{\mathbb{E}[N_T^{(k)}]}{\ln(T)} \geq \frac{1}{d(\nu^{(k^\star)}, \nu^{(k)})},$$

where d is a distance between distributions called the Kullback—Leibler divergence

Thompson sampling in practice

Thompson sampling has excellent performance in practice, but is still just a heuristic

However, asymptotically, i.e., as $T \to \infty$, it actually is optimal in a certain sense

There is an *instance-dependent* lower-bound result that says that for any bandit algorithm:

$$\liminf_{T\to\infty} \frac{\mathbb{E}[N_T^{(k)}]}{\ln(T)} \geq \frac{1}{d(\nu^{(k^\star)}, \nu^{(k)})},$$

where d is a distance between distributions called the Kullback—Leibler divergence

It turns out that Thompson sampling satisfies this lower-bound with equality!

Thompson sampling in practice

Thompson sampling has excellent performance in practice, but is still just a heuristic

However, asymptotically, i.e., as $T \to \infty$, it actually is optimal in a certain sense

There is an *instance-dependent* lower-bound result that says that for any bandit algorithm:

$$\liminf_{T\to\infty} \frac{\mathbb{E}[N_T^{(k)}]}{\ln(T)} \geq \frac{1}{d(\nu^{(k^\star)}, \nu^{(k)})},$$

where d is a distance between distributions called the Kullback—Leibler divergence

It turns out that Thompson sampling satisfies this lower-bound with equality!

So it is asymptotically optimal, not just in its rate, but its constant too!

Thompson sampling in practice

Thompson sampling has excellent performance in practice, but is still just a heuristic

However, asymptotically, i.e., as $T \to \infty$, it actually is optimal in a certain sense

There is an *instance-dependent* lower-bound result that says that for any bandit algorithm:

$$\liminf_{T \to \infty} \frac{\mathbb{E}[N_T^{(k)}]}{\ln(T)} \geq \frac{1}{d(\nu^{(k^\star)}, \nu^{(k)})},$$

where d is a distance between distributions called the Kullback—Leibler divergence

It turns out that Thompson sampling satisfies this lower-bound with equality!

So it is asymptotically optimal, not just in its rate, but its constant too!

(UCB is not, but there are more complicated versions of it that are)

Thompson sampling in practice (cont'd)

So Thompson sampling is basically exactly optimal for large T

Thompson sampling in practice (cont'd)

So Thompson sampling is basically exactly optimal for large T

What could go wrong for smaller T? Suppose $K = 2$ and $T = 3$, and:

Thompson sampling in practice (cont'd)

So Thompson sampling is basically exactly optimal for large T

What could go wrong for smaller T? Suppose $K = 2$ and $T = 3$, and:

- $t = 0$: $a_0 = 1$, $r_0 = 1$

Thompson sampling in practice (cont'd)

So Thompson sampling is basically exactly optimal for large T

What could go wrong for smaller T? Suppose $K = 2$ and $T = 3$, and:

- $t = 0$: $a_0 = 1$, $r_0 = 1$
- $t = 1$: $a_1 = 2$, $r_1 = 0$

Thompson sampling in practice (cont'd)

So Thompson sampling is basically exactly optimal for large T

What could go wrong for smaller T? Suppose $K = 2$ and $T = 3$, and:

- $t = 0$: $a_0 = 1, r_0 = 1$
- $t = 1$: $a_1 = 2, r_1 = 0$
- $t = 2$ (last time step, with $\hat{\mu}_2^{(1)} = 1$ and $\hat{\mu}_2^{(2)} = 0$): $a_2 =$?

Thompson sampling in practice (cont'd)

So Thompson sampling is basically exactly optimal for large T

What could go wrong for smaller T? Suppose $K = 2$ and $T = 3$, and:

- $t = 0$: $a_0 = 1$, $r_0 = 1$
- $t = 1$: $a_1 = 2$, $r_1 = 0$
- $t = 2$ (last time step, with $\hat{\mu}_2^{(1)} = 1$ and $\hat{\mu}_2^{(2)} = 0$): $a_2 =$?

Thompson sampling has a decent probability of choosing $a_2 = 2$, since with just one sample from each arm, Thompson sampling isn't sure which arm is best.

Thompson sampling in practice (cont'd)

So Thompson sampling is basically exactly optimal for large T

What could go wrong for smaller T? Suppose $K = 2$ and $T = 3$, and:

- $t = 0$: $a_0 = 1$, $r_0 = 1$
- $t = 1$: $a_1 = 2$, $r_1 = 0$
- $t = 2$ (last time step, with $\hat{\mu}_2^{(1)} = 1$ and $\hat{\mu}_2^{(2)} = 0$): $a_2 =$?

Thompson sampling has a decent probability of choosing $a_2 = 2$, since with just one sample from each arm, Thompson sampling isn't sure which arm is best.

But $a_2 = 1$ is clear right choice here: there is no future value to learning more, i.e., no reason to explore rather than exploit.

Thompson sampling in practice (cont'd)

So Thompson sampling is basically exactly optimal for large T

What could go wrong for smaller T? Suppose $K = 2$ and $T = 3$, and:

- $t = 0$: $a_0 = 1$, $r_0 = 1$
- $t = 1$: $a_1 = 2$, $r_1 = 0$
- $t = 2$ (last time step, with $\hat{\mu}_2^{(1)} = 1$ and $\hat{\mu}_2^{(2)} = 0$): $a_2 = ?$

Thompson sampling has a decent probability of choosing $a_2 = 2$, since with just one sample from each arm, Thompson sampling isn't sure which arm is best.

But $a_2 = 1$ is clear right choice here: there is no future value to learning more, i.e., no reason to explore rather than exploit.

Thompson sampling doesn't know this, and neither does UCB (although UCB wouldn't happen to make the same mistake in this case).

Thompson sampling in practice (cont'd)

For small T, Thompson sampling is not greedy enough

Thompson sampling in practice (cont'd)

For small T, Thompson sampling is not greedy enough

Fix: add a tuning parameter to make it more greedy. Some possibilities:

Thompson sampling in practice (cont'd)

For small T, Thompson sampling is not greedy enough

Fix: add a tuning parameter to make it more greedy. Some possibilities:

- Update the Beta parameters by $1+\epsilon$ instead of just 1 each time

Thompson sampling in practice (cont'd)

For small T, Thompson sampling is not greedy enough

Fix: add a tuning parameter to make it more greedy. Some possibilities:

- Update the Beta parameters by $1+\epsilon$ instead of just 1 each time
- Instead of just taking one sample of $\boldsymbol{\mu}$ and computing the greedy action with respect to it, take n samples, compute the greedy action with respect to each, and pick the *mode* of those greedy actions

Thompson sampling in practice (cont'd)

For small T, Thompson sampling is not greedy enough

Fix: add a tuning parameter to make it more greedy. Some possibilities:

- Update the Beta parameters by $1+\epsilon$ instead of just 1 each time
- Instead of just taking one sample of $\boldsymbol{\mu}$ and computing the greedy action with respect to it, take n samples, compute the greedy action with respect to each, and pick the *mode* of those greedy actions

All of these favor arms that the algorithm has more confidence are good (i.e., arms that have worked well so far), as opposed to arms that *may* be good

Thompson sampling in practice (cont'd)

For small T, Thompson sampling is not greedy enough

Fix: add a tuning parameter to make it more greedy. Some possibilities:

- Update the Beta parameters by $1+\epsilon$ instead of just 1 each time
- Instead of just taking one sample of $\boldsymbol{\mu}$ and computing the greedy action with respect to it, take n samples, compute the greedy action with respect to each, and pick the *mode* of those greedy actions

All of these favor arms that the algorithm has more confidence are good (i.e., arms that have worked well so far), as opposed to arms that *may* be good

Such tuning can improve Thompson sampling's performance even for reasonably large T (the asymptotic optimality of vanilla TS is *very* asymptotic)

Today

- ✓ Feedback from last lecture
- ✓ Recap
- ✓ UCB regret analysis
- ✓ Regret lower-bound
- ✓ Bayesian bandit
- ✓ Thompson sampling

Summary:

- UCB achieves regret of $\tilde{O}(\sqrt{TK})$
- A regret lower-bound exists that says one can't do better than $\Omega(\sqrt{T})$ regret
- Bayesian bandit prior + Bayes rule gives exact running uncertainty quantification
- Thompson sampling samples optimal arm from its (posterior) distribution
- Thompson sampling achieves excellent performance in practice

Attendance:

bit.ly/3RcTC9T

Feedback:

bit.ly/3RHtlxy