

Lecture 12: Fast Reinforcement Learning ¹

Emma Brunskill

CS234 Reinforcement Learning

Winter 2019

¹With some slides derived from David Silver

Class Structure

- Last time: Fast Learning (Bandits and regret)
- **This time: Fast Learning (Bayesian bandits to MDPs)**
- Next time: Fast Learning & *Exploration*

Settings, Frameworks & Approaches

- Over next couple lectures will consider 2 settings, multiple frameworks, and approaches
- Settings: Bandits (single decisions), MDPs
- Frameworks: evaluation criteria for formally assessing the quality of a RL algorithm
- Approaches: Classes of algorithms for achieving particular evaluation criteria in a certain set
- Note: We will see that some approaches can achieve multiple frameworks in multiple settings

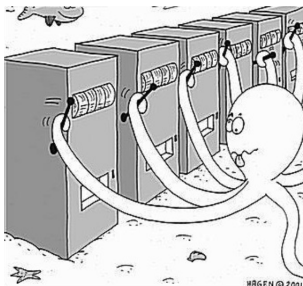
Table of Contents

- 1 Recall: Multi-armed Bandit framework
- 2 Optimism Under Uncertainty for Bandits
- 3 Bayesian Bandits and Bayesian Regret Framework
- 4 Probability Matching
- 5 Framework: Probably Approximately Correct for Bandits
- 6 MDPs

Recall: Multiarmed Bandits

- Multi-armed bandit is a tuple of $(\mathcal{A}, \mathcal{R})$
- \mathcal{A} : known set of m actions (arms)
- $\mathcal{R}^a(r) = \mathbb{P}[r \mid a]$ is an unknown probability distribution over rewards
- At each step t the agent selects an action $a_t \in \mathcal{A}$
- The environment generates a reward $r_t \sim \mathcal{R}^{a_t}$
- Goal: Maximize cumulative reward $\sum_{\tau=1}^t r_{\tau}$

this differs than supervised learning because you only see the reward for the action you take. So the data is "sensored"



Regret

- **Action-value** is the mean reward for action a

$$Q(a) = \mathbb{E}[r \mid a]$$

- **Optimal value** V^*

$$V^* = Q(a^*) = \max_{a \in \mathcal{A}} Q(a)$$

- **Regret** is the opportunity loss for one step

$$l_t = \mathbb{E}[V^* - Q(a_t)]$$

- **Total Regret** is the total opportunity loss

$$L_t = \mathbb{E}\left[\sum_{\tau=1}^t V^* - Q(a_\tau)\right]$$

- Maximize cumulative reward \iff minimize total regret

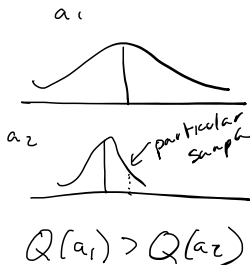


Table of Contents

- 1 Recall: Multi-armed Bandit framework
- 2 Optimism Under Uncertainty for Bandits**
- 3 Bayesian Bandits and Bayesian Regret Framework
- 4 Probability Matching
- 5 Framework: Probably Approximately Correct for Bandits
- 6 MDPs

Approach: Optimism Under Uncertainty

estimate an UCB on the potential expected reward for each of the arms. So for each of the arms, we ask, what do we think is an upper bound for their expected value. When we act, we pick whichever arm has the highest UCB

1993 Kaelbling (MIT)

- Estimate an upper confidence $U_t(a)$ for each action value, such that $Q(a) \leq U_t(a)$ with high probability
- This depends on the number of times $N_t(a)$ action a has been selected
- Select action maximizing Upper Confidence Bound (UCB)

$$a_t = \arg \max_{a \in A} [U_t(a)]$$

Hoeffding
inequality

2 things could happen

- either $a_t = a^*$
- or $a_t \neq a^*$

regret of O
 $U_t(a_t)$ decrease

UCB Bandit Regret

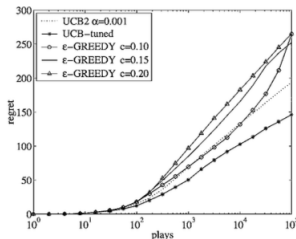
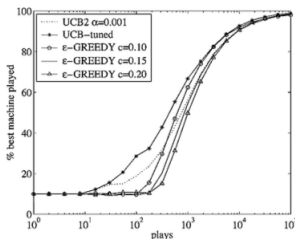
- UCB

$$a_t = \arg \max_{a \in \mathcal{A}} \hat{Q}(a) + \sqrt{\frac{2 \log t}{N_t(a)}}$$

- Theorem: The UCB algorithm achieves logarithmic asymptotic total regret

$$\lim_{t \rightarrow \infty} L_t \leq 8 \log t \sum_{a | \Delta_a > 0} \Delta_a$$

times we act (pointing to the sum)
gaps (pointing to Δ_a)
 $\Delta_a = Q(a^*) - Q(a)$
related but diff to bound from last time
problem-dep bound (pointing to the sum)



Toy Example: Ways to Treat Broken Toes, Optimism¹

- True (unknown) parameters for each arm (action) are
 - surgery: $Q(a^1) = \theta_1 = .95$
 - buddy taping: $Q(a^2) = \theta_2 = .9$
 - doing nothing: $Q(a^3) = \theta_3 = .1$
- Optimism under uncertainty, UCB1 (Auer, Cesa-Bianchi, Fischer 2002)
 - ① Sample each arm once

¹Note: This is a made up example. This is not the actual expected efficacies of the various treatment options for a broken toe

Toy Example: Ways to Treat Broken Toes, Optimism¹

- True (unknown) parameters for each arm (action) are
 - surgery: $Q(a^1) = \theta_1 = .95$
 - buddy taping: $Q(a^2) = \theta_2 = .9$
 - doing nothing: $Q(a^3) = \theta_3 = .1$
- UCB1 (Auer, Cesa-Bianchi, Fischer 2002)
 - ① Sample each arm once
 - Take action a^1 ($r \sim \text{Bernoulli}(0.95)$), get +1, $\hat{Q}(a^1) = 1$
 - Take action a^2 ($r \sim \text{Bernoulli}(0.90)$), get +1, $\hat{Q}(a^2) = 1$
 - Take action a^3 ($r \sim \text{Bernoulli}(0.1)$), get 0, $\hat{Q}(a^3) = 0$

empirical estimate

¹Note: This is a made up example. This is not the actual expected efficacies of the various treatment options for a broken toe

Toy Example: Ways to Treat Broken Toes, Optimism¹

- True (unknown) parameters for each arm (action) are

- surgery: $Q(a^1) = \theta_1 = .95$
- buddy taping: $Q(a^2) = \theta_2 = .9$
- doing nothing: $Q(a^3) = \theta_3 = .1$

- UCB1 (Auer, Cesa-Bianchi, Fischer 2002)

- 1 Sample each arm once

- Take action a^1 ($r \sim \text{Bernoulli}(0.95)$), get +1, $\hat{Q}(a^1) = 1$ ✓
- Take action a^2 ($r \sim \text{Bernoulli}(0.90)$), get +1, $\hat{Q}(a^2) = 1$
- Take action a^3 ($r \sim \text{Bernoulli}(0.1)$), get 0, $\hat{Q}(a^3) = 0$

- 2 Set $t = 3$, Compute upper confidence bound on each action

$$UCB(a) = \hat{Q}(a) + \sqrt{\frac{2 \log t}{N_t(a)}}$$

total arm pulls (circled in yellow, pointing to $2 \log t$)
particular arm (pointing to $N_t(a)$)

$$UCB(a_3) = \sqrt{\frac{2 \log 3}{1}}$$
$$UCB(a_1) = 1 + \sqrt{\frac{2 \log 3}{1}} = UCB(a_2)$$

¹Note: This is a made up example. This is not the actual expected efficacies of the various treatment options for a broken toe

Toy Example: Ways to Treat Broken Toes, Optimism¹

- True (unknown) parameters for each arm (action) are
 - surgery: $Q(a^1) = \theta_1 = .95$
 - buddy taping: $Q(a^2) = \theta_2 = .9$
 - doing nothing: $Q(a^3) = \theta_3 = .1$
- UCB1 (Auer, Cesa-Bianchi, Fischer 2002)
 - 1 Sample each arm once
 - Take action a^1 ($r \sim \text{Bernoulli}(0.95)$), get +1, $\hat{Q}(a^1) = 1$
 - Take action a^2 ($r \sim \text{Bernoulli}(0.90)$), get +1, $\hat{Q}(a^2) = 1$
 - Take action a^3 ($r \sim \text{Bernoulli}(0.1)$), get 0, $\hat{Q}(a^3) = 0$
 - 2 Set $t = 3$, Compute upper confidence bound on each action

$$UCB(a) = \hat{Q}(a) + \sqrt{\frac{2 \log t}{N_t(a)}}$$

- 3 $t = 3$, Select action $a_t = \arg \max_a UCB(a)$,
- 4 Observe reward 1
- 5 Compute upper confidence bound on each action

Check Your Understanding

- True (unknown) parameters for each arm (action) are

- surgery: $Q(a^1) = \theta_1 = .95$
- buddy taping: $Q(a^2) = \theta_2 = .9$
- doing nothing: $Q(a^3) = \theta_3 = .1$

- UCB1 (Auer, Cesa-Bianchi, Fischer 2002)

- 1 Sample each arm once

- Take action a^1 ($r \sim \text{Bernoulli}(0.95)$), get +1, $\hat{Q}(a^1) = 1$
- Take action a^2 ($r \sim \text{Bernoulli}(0.90)$), get +1, $\hat{Q}(a^2) = 1$
- Take action a^3 ($r \sim \text{Bernoulli}(0.1)$), get 0, $\hat{Q}(a^3) = 0$

- 2 Set $t = 3$, Compute upper confidence bound on each action

$$UCB(a) = \hat{Q}(a) + \sqrt{\frac{2 \log t}{N_t(a)}}$$

$$UCB(a_1) = UCB(a_2) = 1 + \sqrt{\frac{2 \log 3}{1}}$$
$$UCB(a_3) = \sqrt{\frac{2 \log 3}{1}}$$

- 3 $t = t + 1$, Select action $a_t = 1$, Observe reward 1

- 4 Compute upper confidence bound on each action α_i

- 5 Assume ties are evenly split. Prob of each arm if using ϵ -greedy (with $\epsilon=0.1$)? If using UCB? ϵ $1/|A|$

Toy Example: Ways to Treat Broken Toes, Optimism, Assessing Regret

pull a_1 got a^1

$$UCB(a_1) = 1 + \sqrt{\frac{2 \log 4}{2}} \quad UCB(a_2) = 1 + \sqrt{\frac{2 \log 4}{1}}$$

- True (unknown) parameters for each arm (action) are

- surgery: $Q(a^1) = \theta_1 = .95$
- buddy taping: $Q(a^2) = \theta_2 = .9$
- doing nothing: $Q(a^3) = \theta_3 = .1$

$$UCB(a_3) = \sqrt{\frac{2 \log 4}{1}}$$

- UCB1 (Auer, Cesa-Bianchi, Fischer 2002)

Action	Optimal Action	Regret
a^1	a^1	0
a^2	a^1	.05 = .95 - .9
a^3	a^1	.85 = .95 - .1
a^1	a^1	0
a^2	a^1	.05

again, we only know this in a simulated world. In reality we don't know this. If we did know this, then we'd just pick the known optimal action every time.

we select a^2 because its more uncertain than a^1 and has the same expected reward (Q-value)

Check Your Understanding



- An alternative would be to always select the arm with the highest lower bound
- Why can this yield linear regret?
- Consider a two arm case for simplicity

Greedy Bandit Algorithms and Optimistic Initialization

another option would be to just initialize the arms to a high value. Then average that in with the actual pulls. The challenge is determining how optimistic to be when doing the fake pull.

- Simple optimism under uncertainty approach
 - Pretend already observed one pull of each arm, and saw some optimistic reward
 - Average these fake pulls and rewards in when computing average empirical reward
- Comparing regret results:
- **Greedy**: Linear total regret
- **Constant ϵ -greedy**: Linear total regret
- **Decaying ϵ -greedy**: Sublinear regret if can use right schedule for decaying ϵ , but that requires knowledge of gaps, which are unknown
- **Optimistic initialization**: Sublinear regret if initialize values sufficiently optimistically, else linear regret

Table of Contents

- 1 Recall: Multi-armed Bandit framework
- 2 Optimism Under Uncertainty for Bandits
- 3 Bayesian Bandits and Bayesian Regret Framework**
- 4 Probability Matching
- 5 Framework: Probably Approximately Correct for Bandits
- 6 MDPs

Bayesian Bandits


an alternative approach is to assume we have some information about the parametric distribution of the rewards, and to exploit that

- So far we have made no assumptions about the reward distribution \mathcal{R}
 - Except bounds on rewards $\mathcal{R} = [0, 1]$
- **Bayesian bandits** exploit prior knowledge of rewards, $p[\mathcal{R}]$
- They compute posterior distribution of rewards $p[\mathcal{R} \mid h_t]$, where $h_t = (a_1, r_1, \dots, a_{t-1}, r_{t-1})$
- Use posterior to guide exploration
 - Upper confidence bounds (Bayesian UCB)
 - Probability matching (Thompson Sampling)
- Better performance if prior knowledge is accurate


given the history (i.e. arms we've pulled and their rewards)

Short Refresher / Review on Bayesian Inference

in our case, the parameters are what define the distribution of rewards for each arm



- In Bayesian view, we start with a prior over the unknown parameters
 - Here the unknown distribution over the rewards for each arm
- Given observations / data about that parameter, update our uncertainty over the unknown parameters using Bayes Rule



when we pull an arm, we get a reward. We then use that reward to update our understanding of the unknown parameters

Short Refresher / Review on Bayesian Inference

- In Bayesian view, we start with a prior over the unknown parameters
 - Here the unknown distribution over the rewards for each arm
- Given observations / data about that parameter, update our uncertainty over the unknown parameters using Bayes Rule
- For example, let the reward of arm i be a probability distribution that depends on parameter ϕ_i (unknown)
- Our initial prior over ϕ_i is $p(\phi_i)$
- We pull arm i and observe reward r_{i1}
- Then we can use this to update our estimate over ϕ_i as

Bayes rule

$$p(\phi_i | r_{i1}) = \frac{\overset{\text{data evidence}}{p(r_{i1} | \phi_i)} \overset{\text{prior}}{p(\phi_i)}}{p(r_{i1})} = \frac{p(r_{i1} | \phi_i) p(\phi_i)}{\int_{\phi_i} p(r_{i1} | \phi_i) p(\phi_i) d\phi_i}$$

posterior

probability of observing that reward,
regardless of the parameters

Short Refresher / Review on Bayesian Inference II

- In Bayesian view, we start with a prior over the unknown parameters
- Given observations / data about that parameter, update our uncertainty over the unknown parameters using Bayes Rule

$$p(\phi_i | r_{i1}) = \frac{\overset{\text{data likelihood}}{p(r_{i1} | \phi_i)} \overset{\text{prior}}{p(\phi_i)}}{\int_{\phi_i} p(r_{i1} | \phi_i) p(\phi_i) d\phi_i}$$

- In general computing this update may be tricky to do exactly with no additional structure on the form of the prior and data likelihood

Short Refresher / Review on Bayesian Inference: Conjugate

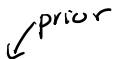
- In Bayesian view, we start with a prior over the unknown parameters
- Given observations / data about that parameter, update our uncertainty over the unknown parameters using Bayes Rule

Gaussian \rightarrow $p(\phi_i | r_{i1}) = \frac{p(r_{i1} | \phi_i) p(\phi_i)}{\int_{\phi_i} p(r_{i1} | \phi_i) p(\phi_i) d\phi_i}$ \leftarrow *conjugate* \leftarrow *Gaussian*

- In general computing this update may be tricky
- But sometimes can be done analytically
- If the parametric representation of the prior and posterior is the same, the prior and model are called **conjugate**.
- For example, exponential families have conjugate priors

Short Refresher / Review on Bayesian Inference: Bernoulli

- Consider a bandit problem where the reward of an arm is a binary outcome $\{0, 1\}$ sampled from a Bernoulli with parameter θ
 - E.g. Advertisement click through rate, patient treatment succeeds/fails, ...
- The Beta distribution $Beta(\alpha, \beta)$ is conjugate for the Bernoulli distribution

prior

$$p(\theta|\alpha, \beta) = \theta^{\alpha-1}(1-\theta)^{\beta-1} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}$$

where $\Gamma(x)$ is the Gamma family.

Short Refresher / Review on Bayesian Inference: Bernoulli

arm with mean = θ

- Consider a bandit problem where the reward of an arm is a binary outcome $\{0, 1\}$ sampled from a Bernoulli with parameter θ
 - E.g. Advertisement click through rate, patient treatment succeeds/fails, ...
- The Beta distribution $Beta(\alpha, \beta)$ is conjugate for the Bernoulli distribution

$$\overbrace{p(\theta|\alpha, \beta)}^{\text{prior}} = \theta^{\alpha-1}(1-\theta)^{\beta-1} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}$$

where $\Gamma(x)$ is the Gamma family.

- Assume the prior over θ is a $Beta(\alpha, \beta)$ as above
- Then after observed a reward $r \in \{0, 1\}$ then updated posterior over θ is $Beta(r + \alpha, 1 - r + \beta)$
*observe $r=1$ $Beta(\alpha+1, \beta)$
 $r=0$ $Beta(\alpha, \beta+1)$*

Framework: Regret and Bayesian Regret

- How do we evaluate performance in the Bayesian setting?
- Frequentist regret assumes a true (unknown) set of parameters

$$\text{Regret}(\mathcal{A}, T; \theta) = \sum_{t=1}^T \mathbb{E}[Q(a^*) - Q(a_t)]$$

- Bayesian regret assumes there is a prior over parameters

$$\text{BayesRegret}(\mathcal{A}, T; \theta) = \mathbb{E}_{\theta \sim p_\theta} \left[\sum_{t=1}^T \mathbb{E}[Q(a^*) - Q(a_t) | \theta] \right]$$

Table of Contents

- 1 Recall: Multi-armed Bandit framework
- 2 Optimism Under Uncertainty for Bandits
- 3 Bayesian Bandits and Bayesian Regret Framework
- 4 Probability Matching**
- 5 Framework: Probably Approximately Correct for Bandits
- 6 MDPs

Approach: Probability Matching

✓1929

- Assume we have a parametric distribution over rewards for each arm
- Probability matching** selects action a according to probability that a is the optimal action

prior pulls & reward outcomes

$$\pi(a | h_t) = \mathbb{P}[Q(a) > Q(a'), \forall a' \neq a | h_t]$$

probability that arm is better than all the other arms

- Probability matching is optimistic in the face of uncertainty
 - Uncertain actions have higher probability of being max
- Can be difficult to compute probability that an action is optimal analytically from posterior
- Somewhat incredibly, a simple approach implements probability matching

Thompson Sampling

$$\theta, \text{ sample } .9$$
$$Q(a_i) = \mathbb{E}[\theta_i] = .9$$

$$\text{Bernoulli } p(\theta_i) \quad i=1:|A|$$
$$p(\theta_i) = \text{Beta}(1, 1)$$

we pick a parametric family to represent our prior distribution of our reward distribution for each arm

- 1: Initialize prior over each arm a , $p(\mathcal{R}_a)$
- 2: **loop**
- 3: For each arm a **sample** a reward distribution \mathcal{R}_a from posterior
- 4: Compute action-value function $Q(a) = \mathbb{E}[\mathcal{R}_a]$
- 5: $a_t = \arg \max_{a \in \mathcal{A}} Q(a) \leftarrow$
- 6: Observe reward r
- 7: Update posterior $p(\mathcal{R}_a | r)$ using Bayes law
- 8: **end loop**

this is the probability over the parameters that determine the rewards distribution

pick a theta

the expected value for a bernoulli parameter is just the theta. For a gaussian, it would be the mean

we pretend that we know the reward distribution and thus $Q(a)$.

Thompson Sampling Implements Probability Matching

↙ prob matching

$$\begin{aligned}\pi(a \mid h_t) &= \mathbb{P}[Q(a) > Q(a'), \forall a' \neq a \mid h_t] \\ &= \mathbb{E}_{\mathcal{R} \mid h_t} \left[\mathbb{1}(a = \arg \max_{a \in \mathcal{A}} Q(a)) \right]\end{aligned}$$

Toy Example: Ways to Treat Broken Toes, Thompson Sampling

instead of using an uninformed prior, you can also use an informed prior. If you have a good prior, it will act as if you already took samples. The downside is that if your informed prior is bad, then you'll be misled for a while. Using an uninformed prior means that you don't get a benefit from prior knowledge, but you also don't get a disadvantage

- True (unknown) Bernoulli parameters for each arm/action
- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- Thompson sampling:
- Place a prior over each arm's parameter. Here choose Beta(1,1)
(Uniform) $p(\theta_i) = \text{Beta}(1, 1)$

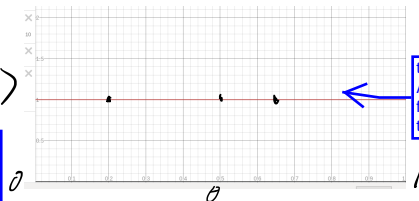
- 1 Sample a Bernoulli parameter given current prior over each arm
Beta(1,1), Beta(1,1), Beta(1,1):

0.3

0.5

0.6

$p(\theta)$



Beta(1,1)

this is our distribution over parameters. And we are going to sample a theta from this distribution. There is one of these distributions for each ith arm

we sampled these parameters. And now we assume it's true! So it makes sense for us to select the third arm (even in reality it's the worst!)

Toy Example: Ways to Treat Broken Toes, Thompson Sampling¹

- True (unknown) Bernoulli parameters for each arm/action
- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- Thompson sampling:
- Place a prior over each arm's parameter. Here choose Beta(1,1)
 - ① Sample a Bernoulli parameter given current prior over each arm
Beta(1,1), Beta(1,1), Beta(1,1): 0.3 0.5 0.6
 - ② Select $a = \arg \max_{a \in A} Q(a) = \arg \max_{a \in A} \theta(a) = 3$

¹Note: This is a made up example. This is not the actual expected efficacies of the various treatment options for a broken toe

Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- True (unknown) Bernoulli parameters for each arm/action
- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- Thompson sampling:
- Place a prior over each arm's parameter. Here choose $\theta_i \sim \text{Beta}(1,1)$
 - 1 Per arm, sample a Bernoulli θ given prior: 0.3 0.5 0.6
 - 2 Select $a_t = \arg \max_{a \in A} Q(a) = \arg \max_{a \in A} \theta(a) = 3$
 - 3 Observe the patient outcome's outcome: 0
 - 4 Update the posterior over the $Q(a_t) = Q(a^3)$ value for the arm pulled

we take arm three

$p(\theta_3 / r=0)$

now we update our posterior for arm 3

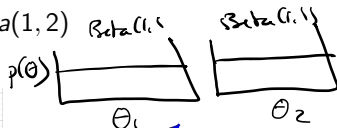
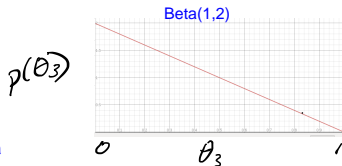
Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- True (unknown) Bernoulli parameters for each arm/action
- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- Thompson sampling:
- Place a prior over each arm's parameter. Here choose $\theta_i \sim \text{Beta}(1,1)$
 - 1 Sample a Bernoulli parameter given current prior over each arm
 $\text{Beta}(1,1), \text{Beta}(1,1), \text{Beta}(1,1): 0.3 \ 0.5 \ 0.6$
 - 2 Select $a_t = \arg \max_{a \in A} Q(a) = \arg \max_{a \in A} \theta(a) = 3$
 - 3 Observe the patient outcome's outcome: 0
 - 4 Update the posterior over the $Q(a_t) = Q(a^3)$ value for the arm pulled
 - $\text{Beta}(c_1, c_2)$ is the conjugate distribution for Bernoulli
 - If observe 1, $c_1 + 1$ else if observe 0 $c_2 + 1$
 - 5 New posterior over Q value for arm pulled is:
 - 6 New posterior $p(Q(a^3)) = p(\theta(a^3)) = \text{Beta}(1, 2)$

Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- True (unknown) Bernoulli parameters for each arm/action
- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- Thompson sampling:
 - 1 Sample a Bernoulli parameter given current prior over each arm
Beta(1,1), Beta(1,1), Beta(1,1): 0.3 0.5 0.6
 - 2 Select $a_t = \arg \max_{a \in A} Q(a) = \arg \max_{a \in A} \theta(a) = 3$
 - 3 Observe the patient outcome's outcome: 0
 - 4 New posterior $p(Q(a_t^*)) = p(\theta(a_t)) = \text{Beta}(1, 2)$

now, if you sampled from this Beta(1,2) distribution, you have a greater chance of sampling a lower value than a larger value



the posterior distribution for the other arms remains unchanged because we didn't pull their arm

Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- True (unknown) Bernoulli parameters for each arm/action
- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- Thompson sampling:
- Place a prior over each arm's parameter. Here choose $\theta_i \sim \text{Beta}(1,1)$
 - ① Sample a Bernoulli parameter given current prior over each arm
Beta(1,1), Beta(1,1), Beta(1,2): 0.7, 0.5, 0.3 $\max \rightarrow a_1$

Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- True (unknown) Bernoulli parameters for each arm/action
- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- Thompson sampling:
- Place a prior over each arm's parameter. Here choose $\theta_i \sim \text{Beta}(1,1)$
 - 1 Sample a Bernoulli parameter given current prior over each arm
 $\text{Beta}(1,1)$, $\text{Beta}(1,1)$, $\text{Beta}(1,2)$: 0.7, 0.5, 0.3
 - 2 Select $a_t = \arg \max_{a \in A} Q(a) = \arg \max_{a \in A} \theta(a) = 1$
 - 3 Observe the patient outcome's outcome: 1
 - 4 New posterior $p(Q(a^1)) = p(\theta(a^1)) = \text{Beta}(2, 1)$

$\#r = 0_S + 1$

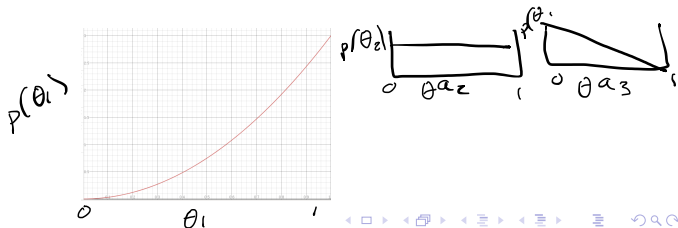
$\#r = 1_S + 1$



$\text{Beta}(2, 1)$
 $\text{Beta}(1, 1)$
 $\text{Beta}(1, 2)$

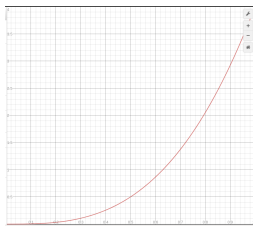
Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- True (unknown) Bernoulli parameters for each arm/action
- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- Thompson sampling:
- Place a prior over each arm's parameter. Here choose $\theta_i \sim \text{Beta}(1,1)$
 - 1 Sample a Bernoulli parameter given current prior over each arm
 $\text{Beta}(2,1)$, $\text{Beta}(1,1)$, $\text{Beta}(1,2)$: 0.71, 0.65, 0.1
 - 2 Select $a_t = \arg \max_{a \in A} Q(a) = \arg \max_{a \in A} \theta(a) = 1$
 - 3 Observe the patient outcome's outcome: 1
 - 4 New posterior $p(Q(a^1)) = p(\theta(a^1)) = \text{Beta}(3,1)$



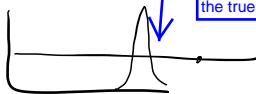
Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- True (unknown) Bernoulli parameters for each arm/action
- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- Thompson sampling:
- Place a prior over each arm's parameter. Here choose $\theta_i \sim \text{Beta}(1,1)$
 - 1 Sample a Bernoulli parameter given current prior over each arm
Beta(2,1), Beta(1,1), Beta(1,2): 0.75, 0.45, 0.4
 - 2 Select $a_t = \arg \max_{a \in A} Q(a) = \arg \max_{a \in A} \theta(a) = 1$
 - 3 Observe the patient outcome's outcome: 1
 - 4 New posterior $p(Q(a^1)) = p(\theta(a^1)) = \text{Beta}(4, 1)$



Toy Example: Ways to Treat Broken Toes, Thompson Sampling vs Optimism

1



if you keep sampling θ_1 , the distribution will eventually collapse to the true value

- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- How does the sequence of arm pulls compare in this example so far?

Optimism	TS	Optimal
a^1	a^3	a^1
a^2	a^1	a^1
a^3	a^1	a^1
a^1	a^1	a^1
a^2	a^1	a^1

in TS you're exploiting faster than UCB.
Notice how often we're taking a^1 compared to UCB

if we didn't have binary rewards, we wouldn't use a Beta and bernoulli. If we had real-valued rewards we could use a Gaussian and have a Gaussian prior. It depends on what your reward distribution looks like, and then you want to find a conjugate for that distribution. There's a lot of different families for parametric distributions that you can use and update.

Toy Example: Ways to Treat Broken Toes, Thompson Sampling vs Optimism

- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- Incurred (frequentist) regret?

Optimism	TS	Optimal	Regret Optimism	Regret TS
a^1	a^3	a^1	0	0 .85
a^2	a^1	a^1	0.05	. 0
a^3	a^1	a^1	0.85	0
a^1	a^1	a^1	0	0
a^2	a^1	a^1	0.05	0

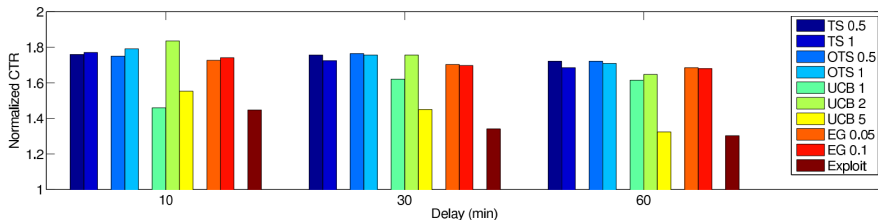
Thompson sampling implements probability matching

- Thompson sampling(1929) achieves Lai and Robbins lower bound
- Bounds for optimism are tighter than for Thompson sampling
- But empirically Thompson sampling can be extremely effective

Thompson Sampling for News Article Recommendation (Chapelle and Li, 2010)

- Contextual bandit: input context which impacts reward of each arm, context sampled iid each step
- Arms = articles
- Reward = click (+1) on article ($Q(a)$ =click through rate)

TS is stochastic where as UCB is deterministic. So if rewards are delayed (i.e. you don't know for a while if you are healed) then in UCB you will do the same thing for other patients until we get some results back. TS we will try different things



Bayesian Regret Bounds for Thompson Sampling

- Regret(UCB,T)

$$BayesRegret(TS, T) = E_{\theta \sim p_\theta} \left[\sum_{t=1}^T Q(a^*) - Q(a_t) | \theta \right]$$

- Posterior sampling has the same (ignoring constants) regret bounds as UCB

Table of Contents

- 1 Recall: Multi-armed Bandit framework
- 2 Optimism Under Uncertainty for Bandits
- 3 Bayesian Bandits and Bayesian Regret Framework
- 4 Probability Matching
- 5 Framework: Probably Approximately Correct for Bandits**
- 6 MDPs

Framework: Probably Approximately Correct

are you giving everyone a headache,
or a few patients very bad side effects

the action we select is within epsilon of
the optimal value

regret is cumulative, so we can't
distinguish between the two

- Theoretical regret bounds specify how regret grows with T
- Could be making lots of little mistakes or infrequent large ones
- May care about bounding the number of non-small errors
- More formally, probably approximately correct (PAC) results state that the algorithm will choose an action a whose value is ϵ -optimal ($Q(a) \geq Q(a^*) - \epsilon$) with probability at least $1 - \delta$ on all but a polynomial number of steps
- Polynomial in the problem parameters (# actions, ϵ , δ , etc)
- Most PAC algorithms based on optimism or Thompson sampling

"approximately
correct"

"probably"

the number of mistakes bigger than
epsilon (i.e. patients with big side
effects) is no more than a polynomial
function. Can compute this in advance

PAC for bandits is less
common than PAC for MDPs

Toy Example: Probably Approximately Correct and Regret

- Surgery: $\theta_1 = \underline{.95}$ / Taping: $\theta_2 = \underline{.9}$ / Nothing: $\theta_3 = \underline{.1}$
- Let $\epsilon = 0.05$.
- O = Optimism, TS = Thompson Sampling: W/in $\epsilon = I(Q(a_t) \geq Q(a^*) - \epsilon)$

in PAC we would count the mistakes

counting mistakes

O	TS	Optimal	O Regret	O W/in ϵ	TS Regret	TS W/in ϵ
a^1	a^3	a^1	0	\checkmark	0.85	N
a^2	a^1	a^1	0.05	\checkmark	0	\checkmark
a^3	a^1	a^1	0.85	N	0	\checkmark
a^1	a^1	a^1	0	\checkmark	0	\checkmark
a^2	a^1	a^1	0.05	\checkmark	0	\checkmark

allows us to take either a^1 or a^2 because they are within epsilon of each other

Toy Example: Probably Approximately Correct and Regret

- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- Let $\epsilon = 0.05$.
- O = Optimism, TS = Thompson Sampling: W/in $\epsilon = I(Q(a_t) \geq Q(a^*) - \epsilon)$

O	TS	Optimal	O Regret	O W/in ϵ	TS Regret	TS W/in ϵ
a^1	a^3	a^1	0	Y	0.85	N
a^2	a^1	a^1	0.05	Y	0	Y
a^3	a^1	a^1	0.85	N	0	Y
a^1	a^1	a^1	0	Y	0	Y
a^2	a^1	a^1	0.05	Y	0	Y

Table of Contents

We've covered two approaches:
Optimism, and TS

- 1 Recall: Multi-armed Bandit framework
- 2 Optimism Under Uncertainty for Bandits
- 3 Bayesian Bandits and Bayesian Regret Framework
- 4 Probability Matching
- 5 Framework: Probably Approximately Correct for Bandits
- 6 MDPs**

tabular MDPs

- Very similar set of frameworks and approaches are relevant for fast learning in reinforcement learning
- Frameworks
 - Regret
 - Bayesian regret
 - Probably approximately correct (PAC)
- Approaches
 - Optimism under uncertainty
 - Probability matching / Thompson sampling
- Framework: Probably approximately correct

Fast RL in Markov Decision Processes

- Very similar set of frameworks and approaches are relevant for fast learning in reinforcement learning
- Frameworks
 - Regret
 - Bayesian regret
 - Probably approximately correct (PAC)
- Approaches
 - **Optimism under uncertainty**
 - Probability matching / Thompson sampling
- Framework: Probably approximately correct

Optimistic Initialization: Model-Free RL

highest reward you can
see in any (s,a) pair



- Initialize action-value function $Q(s,a)$ optimistically (for ex. $\frac{r_{max}}{1-\gamma}$)
 - where $r_{max} = \max_a \max_s R(s, a)$
 - Check your understanding: why is that value guaranteed to be optimistic?
- Run favorite model-free RL algorithm
 - Monte-Carlo control
 - Sarsa
 - Q-learning ...
- Encourages systematic exploration of states and actions

Optimistic Initialization: Model-Free RL

- Initialize action-value function $Q(s,a)$ optimistically (for ex. $\frac{r_{max}}{1-\gamma}$)
 - where $r_{max} = \max_a \max_s R(s, a)$
- Run model-free RL algorithm: MC control, Sarsa, Q-learning ...
- In general the above have no guarantees on performance, but may work better than greedy or ϵ -greedy approaches
- Even-Dar and Mansour (NeurIPS 2002) proved that
 - If run Q-learning with learning rates α_i on time step i ,
 - If initialize $V(s) = \frac{r_{max}}{(1-\gamma) \prod_{i=1}^T \alpha_i}$ where α_i is the learning rate on step i and T is the number of samples need to learn a near optimal Q
 - Then greedy-only Q-learning is PAC
- Recent work (Jin, Allen-Zhu, Bubeck, Jordan NeurIPS 2018) proved that (much less) optimistically initialized Q-learning has good (though not tightest) regret bounds

$$\alpha = \frac{1}{i^\tau} \approx 10^{-\tau}$$

Approaches to Model-based Optimism for Provably Efficient RL

- ① Be very optimistic until confident that empirical estimates are close to true (dynamics/reward) parameters (Brafman & Tennenholtz JMLR 2002)
- ② Be optimistic given the information have
 - Compute confidence sets on dynamics and reward models, or
 - Add reward bonuses that depend on experience / data
- We will focus on the last class of approaches

Model-Based Interval Estimation with Exploration Bonus (MBIE-EB)

initialize counts that we've seen for (s,a) and (s,a,s')

(Strehl and Littman, J of Computer & Sciences 2008)

$\gamma_{max} = 1$

$r \in (0, 1)$ bounded

- 1: Given ϵ, δ, γ
- 2: $n_{sa}(s,a) = 0 \quad \forall s, \forall a \quad n(s,a,s') = 0 \quad \forall s, \forall a, \forall s' \quad rc(s,a) = 0 \quad \forall s, \forall a$
- 3: $\beta = \frac{1}{1-\gamma} \sqrt{2 \log(15/(\delta \epsilon))}$
- 4: $t \leftarrow 0, s_t = \text{initial state}$
- 5: $\tilde{Q}_t(s,a) = \gamma/(1-\gamma) \quad \forall s, \forall a$
- 6: **loop**
- 7: $a_t = \arg \max_{a \in A} \tilde{Q}(s_t, a)$
- 8: Observe reward r_t and state s_{t+1}
- 9: $n_{sa}(s,a)++ \quad n_{s's'}(s,a,s')++$
- 10: $\hat{r}(s,a) = rc(s,a) / n_{sa}(s,a) \quad \hat{T}(s'|s,a) = n(s,a,s') / n(s,a) \quad \forall s,a$
- 11: $\tilde{Q}_t(s,a) = \hat{r}(s,a) + \gamma \sum_{s'} \hat{T}(s'|s,a) \max_{a'} \tilde{Q}(s', a') + \beta / \sqrt{n_{sa}(s,a)}$
- 12: **while not converged do**
- 13: $\tilde{Q}(s,a) = \hat{r}(s,a) + \gamma \sum_{s'} \hat{T}(s'|s,a) \max_{a'} \tilde{Q}(s', a') + \beta / \sqrt{n_{sa}(s,a)}$
- 14: **end while**
- 15: **end loop**

beta is a parameter we use to define our reward bonuses

keep track of total sum of rewards for (s,a)

update counts

empirical reward model

empirical transition model

$\beta / \sqrt{n_{sa}(s,a)}$
reward bonus

Model-Based Interval Estimation with Exploration Bonus (MBIE-EB)

(Strehl and Littman, J of Computer & Sciences 2008)

-
- 1: Given ϵ, δ, m
 - 2: $\beta = \frac{1}{1-\gamma} \sqrt{0.5 \ln(2|S||A|m/\delta)}$
 - 3: $n_{sas}(s, a, s') = 0 \quad s \in S, a \in A, s' \in S$
 - 4: $rc(s, a) = 0, n_{sa}(s, a) = 0, \tilde{Q}(s, a) = 1/(1-\gamma) \quad \forall s \in S, a \in A$
 - 5: $t = 0, s_t = s_{init}$
 - 6: **loop**
 - 7: $a_t = \arg \max_{a \in A} Q(s_t, a)$
 - 8: Observe reward r_t and state s_{t+1}
 - 9: $n_{sa}(s_t, a_t) = n(s_t, a_t) + 1, n_{sas}(s_t, a_t, s_{t+1}) = n_{sas}(s_t, a_t, s_{t+1}) + 1$
 - 10: $rc(s_t, a_t) = \frac{rc(s_t, a_t)n_{sa}(s_t, a_t) + r_t}{(n_{sa}(s_t, a_t) + 1)}$
 - 11: $\hat{R}(s, a) = \frac{rc(s_t, a_t)}{n(s_t, a_t)}$ and $\hat{T}(s'|s, a) = \frac{n_{sas}(s_t, a_t, s')}{n_{sa}(s_t, a_t)} \quad \forall s' \in S$
 - 12: **while** not converged **do**
 - 13: $\tilde{Q}(s, a) = \hat{R}(s, a) + \gamma \sum_{s'} \hat{T}(s'|s, a) \max_{a'} \tilde{Q}(s', a') + \underbrace{\frac{\beta}{\sqrt{n_{sa}(s, a)}}}_{\text{reward bonus}} \quad \forall s \in S, a \in A$
 - 14: **end while**
 - 15: **end loop**

Framework: PAC for MDPs

- For a given ϵ and δ , A RL algorithm \mathcal{A} is PAC if on all but N steps, the action selected by algorithm \mathcal{A} on time step t , a_t , is ϵ -close to the optimal action, where N is a polynomial function of $(\underbrace{|S|, |A|, \gamma, \epsilon, \delta})$
- Is this true for all algorithms? $\mathcal{N}^?$

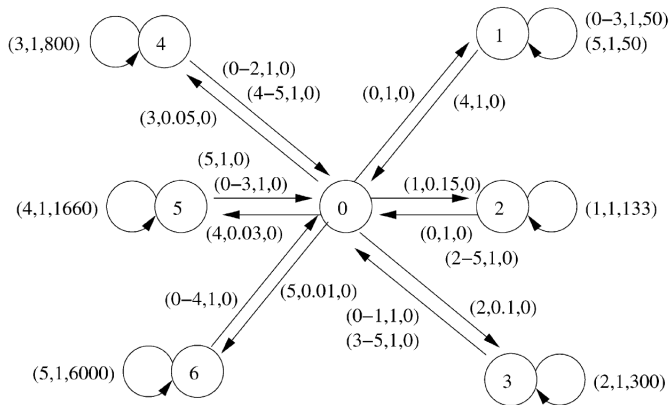
MBIE-EB is a PAC RL Algorithm

Theorem 2. Suppose that ϵ and δ are two real numbers between 0 and 1 and $M = \langle S, A, T, \mathcal{R}, \gamma \rangle$ is any MDP. There exists an input $m = m(\frac{1}{\epsilon}, \frac{1}{\delta})$, satisfying $m(\frac{1}{\epsilon}, \frac{1}{\delta}) = O(\frac{|S|}{\epsilon^2(1-\gamma)^4} + \frac{1}{\epsilon^2(1-\gamma)^4} \ln \frac{|S||A|}{\epsilon(1-\gamma)^\delta})$, and $\beta = (1/(1-\gamma))\sqrt{\ln(2|S||A|m/\delta)}/2$ such that if MBIE-EB is executed on MDP M , then the following holds. Let \mathcal{A}_t denote MBIE-EB's policy at time t and s_t denote the state at time t . With probability at least $1 - \delta$, $V_M^{\mathcal{A}_t}(s_t) \geq V_M^*(s_t) - \epsilon$ is true for all but $O(\underbrace{\frac{|S||A|}{\epsilon^3(1-\gamma)^6}(|S| + \ln \frac{|S||A|}{\epsilon(1-\gamma)^\delta}) \ln \frac{1}{\delta} \ln \frac{1}{\epsilon(1-\gamma)}}_{\text{timesteps}}) t$.

A Sufficient Set of Conditions to Make a RL Algorithm PAC

- Strehl, A. L., Li, L., Littman, M. L. (2006). Incremental model-based learners with formal learning-time guarantees. In Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence (pp. 485-493)

MBIE-EB Empirically: 6 Arms Simulation



MBIE-EB Empirically: 6 Arms Results

