

Lecture 12: Fast Reinforcement Learning ¹

Emma Brunskill

CS234 Reinforcement Learning

Winter 2020

¹With some slides derived from David Silver

Refresh Your Understanding: Multi-armed Bandits

- Select all that are true:

- ① UCB selects the arm with $\arg \max_a \hat{Q}_t(a) + \sqrt{\frac{1}{N_t(a)} \log(1/\delta)}$
- ② Over an infinite trajectory, UCB will sample all arms an infinite number of times
- ③ UCB still would learn to pull the optimal arm the most if we instead used $\arg \max_a \hat{Q}_t(a) + \sqrt{\frac{1}{N_t(a)} \log(t/\delta)}$
- ④ UCB uses a bonus on top of average empirical reward. If the bonus was $\epsilon > 0$ but small, the resulting algorithm might still suffer linear regret
- ⑤ Algorithms that minimize regret also maximize reward
- ⑥ Not sure

Class Structure

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

Recall Motivation

- Fast learning is important when our decisions impact the real world

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. So far seen empirical evaluations, asymptotic convergence, regret
- Approaches: Classes of algorithms for achieving particular evaluation criteria in a certain set. So far for exploration seen: greedy, ϵ -greedy, optimism

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 $\sum_{\tau=1}^t r_\tau$
- **Regret** is the opportunity loss for one step

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

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

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

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 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)

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

- Theorem: The UCB algorithm achieves logarithmic asymptotic total regret

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

Simpler Optimism?

- Do we need to formally model uncertainty to get the "right" level of optimism?

Greedy Bandit Algorithms and Optimistic Initialization

- 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

Greedy Bandit Algorithms and Optimistic Initialization

- 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

- So far we have made no assumptions about the reward distribution \mathcal{R}
 - Except bounds on rewards
- **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

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

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
- Initial prior over ϕ_i is $p(\phi_i)$
- Pull arm i and observe reward r_{i1}
- Use Bayes rule to update estimate over ϕ_i :

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
- Initial prior over ϕ_i is $p(\phi_i)$
- Pull arm i and observe reward r_{i1}
- Use Bayes rule to update estimate over ϕ_i :

$$p(\phi_i | r_{i1}) = \frac{p(r_{i1} | \phi_i) p(\phi_i)}{p(r_{i1})} = \frac{p(r_{i1} | \phi_i) p(\phi_i)}{\int_{\phi_i} p(r_{i1} | \phi_i) p(\phi_i) d\phi_i}$$

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

$$p(\phi_i | r_{i1}) = \frac{p(r_{i1} | \phi_i) p(\phi_i)}{\int_{\phi_i} p(r_{i1} | \phi_i) p(\phi_i) d\phi_i}$$

- 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

$$p(\phi_i | r_{i1}) = \frac{p(r_{i1} | \phi_i) p(\phi_i)}{\int_{\phi_i} p(r_{i1} | \phi_i) p(\phi_i) d\phi_i}$$

- 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

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

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

Short Refresher / Review on Bayesian Inference: Bernoulli

- 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

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

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

- 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 $Beta(r + \alpha, 1 - r + \beta)$

Bayesian Inference for Decision Making

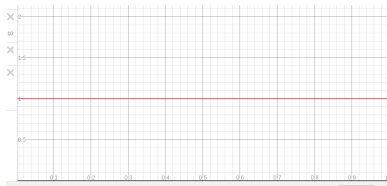
- Maintain distribution over reward parameters
- Use this to inform action selection

Thompson Sampling

-
- 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: $a_t = \arg \max_{a \in \mathcal{A}} Q(a)$
 - 6: Observe reward r
 - 7: Update posterior $p(\mathcal{R}_a|r)$ using Bayes law
 - 8: **end loop**
-

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



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

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(1,1): 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
 - 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:
- 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(1,1): 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 $p(Q(a^3)) = p(\theta(a_3) = \text{Beta}(1, 2)$

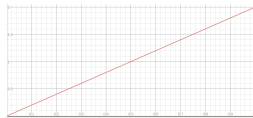


Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- True (unknown) Bernoulli parameters for each arm/action
- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- Thompson sampling:
- 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

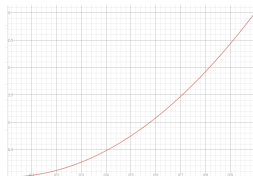
Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- 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 $p(Q(a^1)) = p(\theta(a^1)) = \text{Beta}(2, 1)$



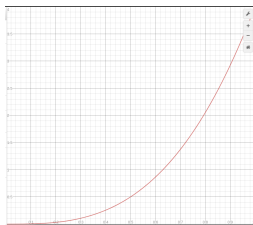
Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- 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): 0.71, 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

- 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^2	a^1	
a^3	a^1	
a^1	a^1	
a^2	a^1	

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
a^2	a^1	a^1	0.05	
a^3	a^1	a^1	0.85	
a^1	a^1	a^1	0	
a^2	a^1	a^1	0.05	

- Now we will see how Thompson sampling works in general, and what it is doing

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

- 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

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

- 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

-
- 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: $a_t = \arg \max_{a \in \mathcal{A}} Q(a)$
 - 6: Observe reward r
 - 7: Update posterior $p(\mathcal{R}_a|r)$ using Bayes law
 - 8: **end loop**
-

Thompson Sampling Implements Probability Matching

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

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

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

Bayesian Regret Bounds for Thompson Sampling

- Regret(UCB,T)

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

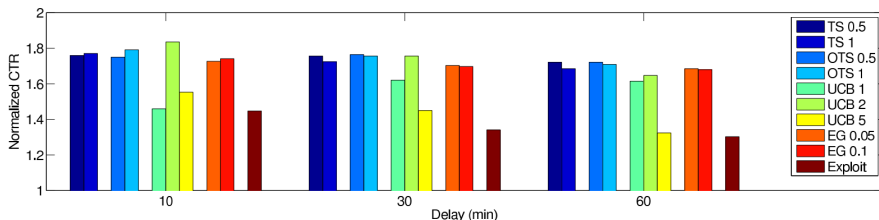
- Posterior sampling has the same (ignoring constants) regret bounds as UCB

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 did extremely well! Lead to a big resurgence of interest in Thompson sampling.



Check Your Understanding: Thompson Sampling and Optimism

- Consider an online news website with thousands of people logging on each second. Frequently a new person will come online before we see whether the last person has clicked (or not). Select all that are true:
 - ① Thompson sampling would be better than optimism here, because optimism algorithms are deterministic and would select the same action until we get feedback (click or not).
 - ② Optimism algorithms would be better than TS here, because they have stronger regret bounds
 - ③ Thompson sampling could cause much worse performance than optimism if the initial prior is very misleading.
 - ④ Not sure

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

- 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

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 $\epsilon = \mathbb{I}(Q(a_t) \geq Q(a^*) - \epsilon)$

O	TS	Optimal	O Regret	O W/in ϵ	TS Regret	TS W/in ϵ
a^1	a^3	a^1	0		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	

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 $\epsilon = \mathbb{I}(Q(a_t) \geq Q(a^*) - \epsilon)$

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

- 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**

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

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

- 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 a_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

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)

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

```
1: Given  $\epsilon, \delta, m$ 
2:
3:
4:
5:
6: loop
7:    $a_t = \arg \max_{a \in \mathcal{A}} Q(s_t, a)$ 
8:   Observe reward  $r_t$  and state  $s_{t+1}$ 
9:
10:
11:
12:   while not converged do
13:      $\tilde{Q}(s, a) = \hat{R}(s, a) + \gamma \sum_{s'} \hat{T}(s'|s, a) \max_{a'} \tilde{Q}(s', a') +$ 
14:   end while
15: end loop
```

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: $n_{sas}(s, a, s') = 0 \ s \in S, a \in A, s' \in S$
 - 4: $rc(s, a) = 0, n_{sa}(s, a) = 0, \tilde{Q}(s, a) = 1/(1-\gamma) \ \forall s \in S, a \in A$
 - 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 $\hat{T}(s'|s, a) = \frac{n_{sas}(s_t, a_t, s')}{n_{sa}(s_t, a_t)} \ \forall s' \in S$
 - 12: **while** not converged **do**
 - 13: $\tilde{Q}(s, a) = \hat{R}(s, a) + \gamma \sum_{s'} \hat{T}(s'|s, a) \max_{a'} \tilde{Q}(s', a') + \frac{\beta}{\sqrt{n_{sa}(s, a)}} \ \forall s \in S, a \in A$
 - 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 $(|S|, |A|, \gamma, \epsilon, \delta)$
- Is this true for all algorithms?

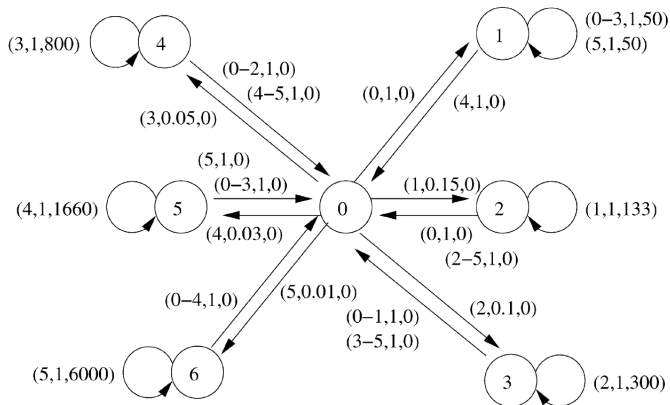
MBIE-EB is a PAC RL Algorithm

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 $O(\frac{|S||A|}{\epsilon^3(1-\gamma)^6} (|S| + \ln \frac{|S||A|}{\epsilon(1-\gamma)^\delta}) \ln \frac{1}{\delta} \ln \frac{1}{\epsilon(1-\gamma)})$ timesteps t .

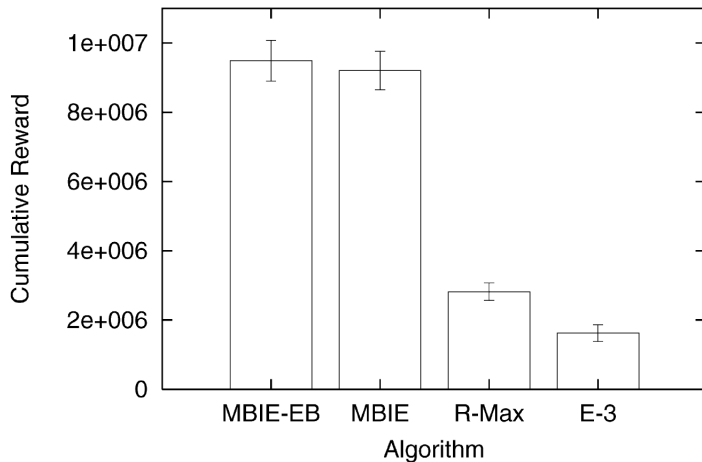
A Sufficient Set of Conditions to Make a RL Algorithm PAC

- 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



Summary so Far: Settings, Frameworks & Approaches

- Over 3 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. So far seen empirical evaluations, asymptotic convergence, regret, probably approximately correct (PAC)
- Approaches: Classes of algorithms for achieving particular evaluation criteria in a certain set. So far for exploration seen: greedy, ϵ -greedy, optimism, Thompson sampling