Lecture 12: Fast Reinforcement Learning Part II [2]

Emma Brunskill

CS234 Reinforcement Learning.

Winter 2018

[2] With many slides from or derived from David Silver, Worked Examples New

Class Structure

- Last time: Fast Learning, Exploration/Exploitation Part 1
- **This Time: Fast Learning Part II**
- Next time: Batch RL

Table of Contents

Performance Criteria of RL Algorithms

- Empirical performance
- Convergence (to something ...)
- Asymptotic convergence to optimal policy
- Finite sample guarantees: probably approximately correct
- Regret (with respect to optimal decisions)
- Optimal decisions given information have available
- PAC uniform

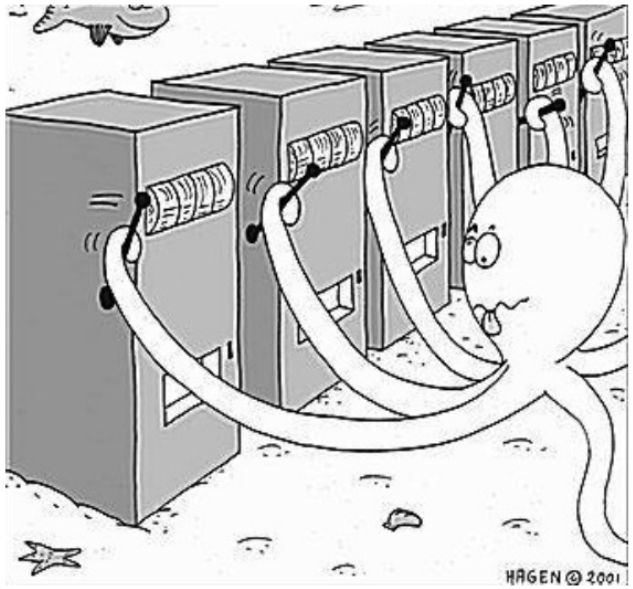
Table of Contents

Principles

- Naive Exploration (last time)
- Optimistic Initialization (last time)
- Optimism in the Face of Uncertainty (last time + this time)
- Probability Matching (last time + this time)
- Information State Search (this time)

Multiarmed Bandits

- Multi-armed bandit is a tuple of $(\mathcal{A}, \mathcal{R})$
- $\mathcal{A}$: known set of m actions
- $\mathcal{R}^a(r) = \mathbb{P}[r \mid a]$ 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

- **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

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

- Maximize cumulative reward $\Longleftrightarrow$ minimize total regret

Optimism Under Uncertainty: Upper Confidence Bounds

- Estimate an upper confidence $\hat{U}_t(a)$ for each action value, such that $Q(a) \leq \hat{Q}_t(a) + \hat{U}_t(a)$ with high probability
- This depends on the number of times $N(a)$ has been selected
  - Small $N_t(a) \rightarrow$ large $\hat{U}_t(a)$ (estimate value is uncertain)
  - Large $N_t(a) \rightarrow$ small $\hat{U}_t(a)$ (estimate value is accurate)
- Select action maximizing Upper Confidence Bound (UCB)

$$a_t = \arg\max a \in \mathcal{A} \hat{Q}_t(a) + \hat{U}_t(a)$$

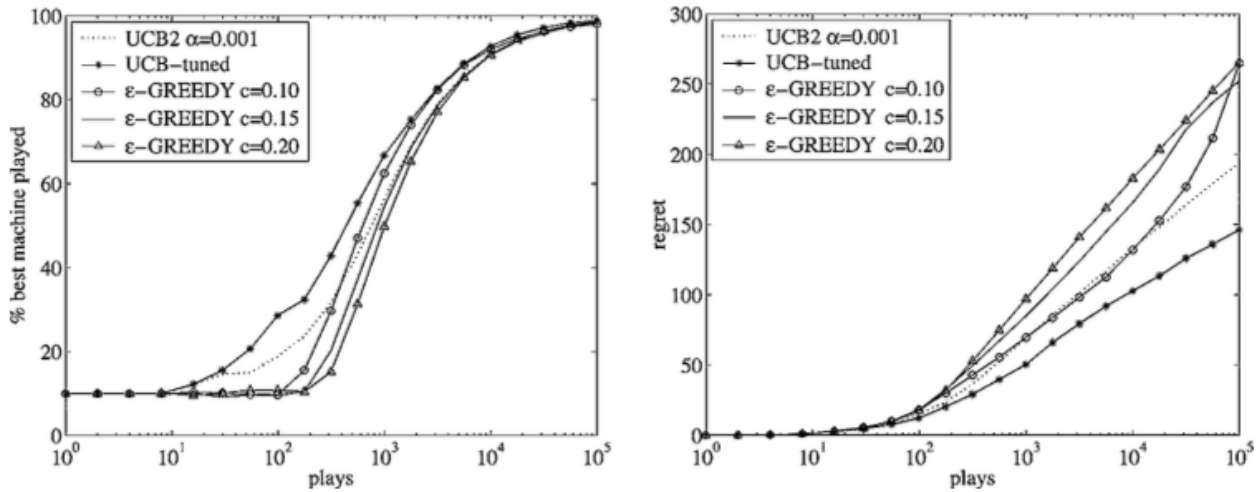
UCB1

- This leads to the UCB1 algorithm

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

- Theorem: The UCB algorithm achieves logarithmic asymptotic total regret

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



Toy Example: Ways to Treat Broken Toes[13]

- Consider deciding how to best treat patients with broken toes
- Imagine have 3 possible options: (1) surgery (2) buddy taping the broken toe with another toe, (3) do nothing
- Outcome measure is binary variable: whether the toe has healed (+1) or not healed (0) after 6 weeks, as assessed by x-ray

[13] Note: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[15]

- Consider deciding how to best treat patients with broken toes
- Imagine have 3 common options: (1) surgery (2) surgical boot (3) buddy taping the broken toe with another toe
- Outcome measure is binary variable: whether the toe has healed (+1) or not (0) after 6 weeks, as assessed by x-ray
- Model as a multi-armed bandit with 3 arms, where each arm is a Bernoulli variable with an unknown parameter θ_i
- Check your understanding: what does a pull of an arm / taking an action correspond to? Why is it reasonable to model this as a multi-armed bandit instead of a Markov decision process?

[15] Note: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[17]

- Imagine 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: $Q(a^3) = \theta_3 = .1$

[17] Note: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[19]

- 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

[19] Note: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, Optimism[21]

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

[21] Note: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, Optimism[23]

- 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 ($r \sim$Bernoulli(0.95), get $+1$, $Q(a^1) = 1$
     - Take action a^2 ($r \sim$Bernoulli(0.90), get $+1$, $Q(a^2) = 1$
     - Take action a^3 ($r \sim$Bernoulli(0.1), get 0, $Q(a^3) = 0$
  2. Set $t = 3$, Compute upper confidence bound on each action

$$ucb(a) = Q(a) + \sqrt{\frac{2lnt}{N_t(a)}}$$

[23] Note: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, Optimism[25]

- 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 ($r \sim$Bernoulli(0.95), get $+1$, $Q(a^1) = 1$
     - Take action a^2 ($r \sim$Bernoulli(0.90), get $+1$, $Q(a^2) = 1$
     - Take action a^3 ($r \sim$Bernoulli(0.1), get 0, $Q(a^3) = 0$
  2. Set $t = 3$, Compute upper confidence bound on each action

$$ucb(a) = Q(a) + \sqrt{\frac{2lnt}{N_t(a)}}$$

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

[25] Note: This is a made up example. This is not the actual expected efficacies of the

Toy Example: Ways to Treat Broken Toes, Optimism[27]

- 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 ($r \sim$Bernoulli(0.95), get $+1$, $Q(a^1) = 1$
     - Take action a^2 ($r \sim$Bernoulli(0.90), get $+1$, $Q(a^2) = 1$
     - Take action a^3 ($r \sim$Bernoulli(0.1), get 0, $Q(a^3) = 0$
  2. Set $t = 3$, Compute upper confidence bound on each action

$$ucb(a) = Q(a) + \sqrt{\frac{2lnt}{N_t(a)}}$$

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

[27] Note: This is a made up example. This is not the actual expected efficacies of the

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

- 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)

Action	Optimal Action	Regret
a^1	a^1	
a^2	a^1	
a^3	a^1	
a^1	a^1	
a^2	a^1	

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

Table of Contents

Probability Matching

- Assume 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 analytically from posterior

Thompson sampling implements probability matching

- Thompson sampling:

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

Thompson sampling implements probability matching

- Thompson sampling:

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

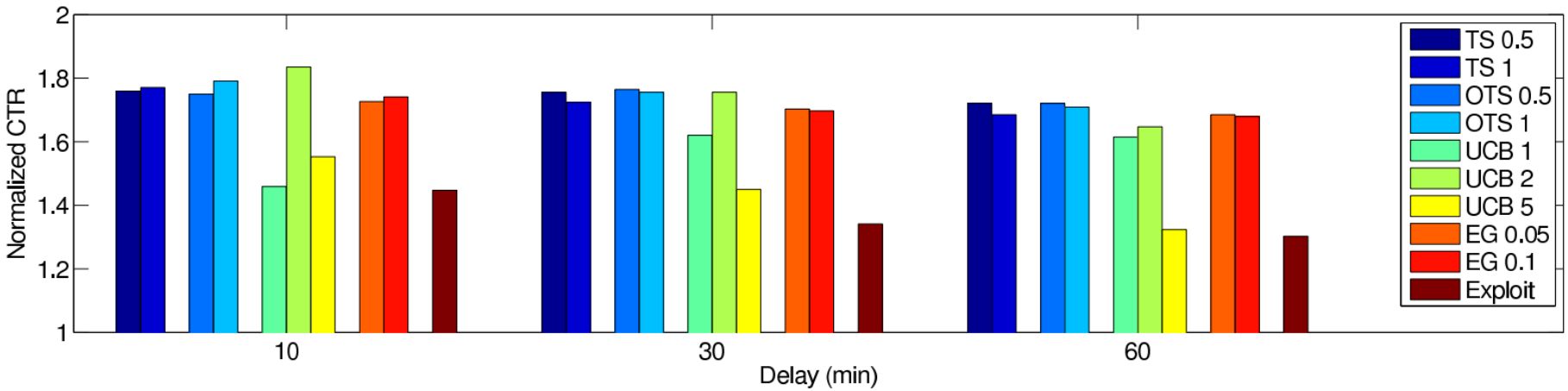
- Use Bayes law to compute posterior distribution $p[\mathcal{R} \mid h_t]$
- **Sample** a reward distribution $\mathcal{R}$ from posterior
- Compute action-value function $Q(a) = \mathbb{E}[\mathcal{R}_a]$
- Select action maximizing value on sample, $a_t = \arg\max_{a \in \mathcal{A}} Q(a)$

Thompson sampling implements probability matching

- Thompson sampling achieves Lai and Robbins lower bound
- Last checked: bounds for optimism are tighter than for Thomspon 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)



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 Sampling[38]

- 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 $a = \arg\max_{a \in A} Q(a) = \arg\max_{ainA} \theta(a) =$

[38] Note: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 Beta(1,1)
  1. Per arm, sample a Bernoulli θ given prior: 0.3 0.5 0.6
  2. Select $a_t = \arg\max_{a \in A} Q(a) = \arg\max_{ainA} \theta(a) = 3$
  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 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 $a_t = \arg\max_{a \in A} Q(a) = \arg\max_{ainA} \theta(a) = 3$
  3. Observe the patient outcome's outcome: 0
  4. Update the posterior over the $Q(a_t) = Q(a^1)$ 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 $p(Q(a^3)) = p(\theta(a_3) = Beta(1, 2)$

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 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 $a_t = \arg\max_{a \in A} Q(a) = \arg\max_{ainA} \theta(a) = 1$
  3. Observe the patient outcome's outcome: 0
  4. New posterior $p(Q(a^1)) = p(\theta(a_1) = Beta(1, 2)$

Toy Example: Ways to Treat Broken Toes, Thompson Sampling

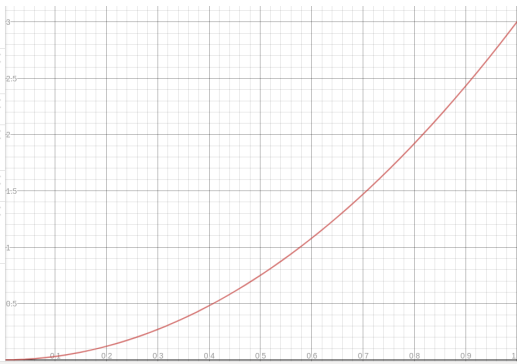
- True (unknown) Bernoulli parameters for each arm/action
- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- Thompson sampling:
- Place a prior over each arm's parameter. Here choose Beta(1,1)
  1. Sample a Bernoulli parameter given current prior over each arm
     Beta(1,1), 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 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
  2. Select $a_t = \arg\max_{a \in A} Q(a) = \arg\max_{ainA} \theta(a) = 1$
  3. Observe the patient outcome's outcome: 1
  4. New posterior $p(Q(a^1)) = p(\theta(a_1) = Beta(2, 1)$

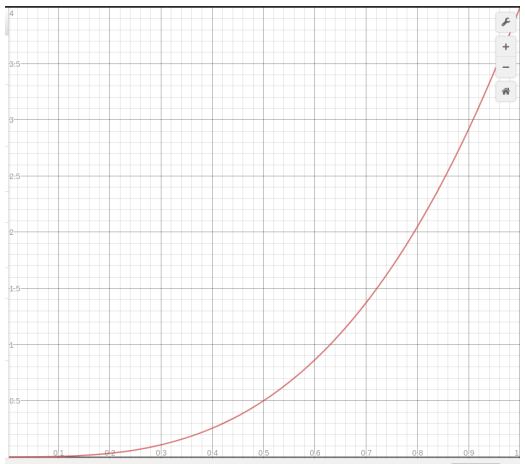
Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- 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 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 $a_t = \arg\max_{a \in A} Q(a) = \arg\max_{ainA} \theta(a) = 1$
  3. Observe the patient outcome's outcome: 1
  4. New posterior $p(Q(a^1)) = p(\theta(a_1) = Beta(3, 1)$



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 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 $a_t = \arg\max_{a \in A} Q(a) = \arg\max_{ainA} \theta(a) = 1$
  3. Observe the patient outcome's outcome: 1
  4. New posterior $p(Q(a^1)) = p(\theta(a_1) = Beta(4, 1)$



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	Regret Optimism	Regret TS
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 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	

Alternate Metric: 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)
- Exist 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 $\epsilon = I(Q(a_t) \geq Q(a^*) - \epsilon)$

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

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

Relevant Background: Value of Information

- Exploration is useful because it gains information
- Can we quantify the **value of information (VOI)**?
  - How much reward a decision-maker would be prepared to pay in order to have that information, prior to making a decision
  - Long-term reward after getting information - immediate reward

Relevant Background: Value of Information Example

- Consider bandit where only get to make a **single** decision
- Oil company considering buying rights to drill in 1 of 5 locations
- 1 of locations contains \$10 million worth of oil, others 0
- Cost of buying rights to drill is \$2 million
- Seismologist says for a fee will survey one of 5 locations and report back definitively whether that location does or does not contain oil
- What should one consider paying seismologist?

Relevant Background: Value of Information Example

- 1 of locations contains \$10 million worth of oil, others 0
- Cost of buying rights to drill is \$2 million
- Seismologist says for a fee will survey one of 5 locations and report back definitively whether that location does or does not contain oil
- Value of information: expected profit if ask seismologist minus expected profit if don't ask
- Expected profit if don't ask:
  - Guess at random

$$= \frac{1}{5}(10-2) + \frac{4}{5}(0-2) = 0 \tag{1}$$

Relevant Background: Value of Information Example

- 1 of locations contains \$10 million worth of oil, others 0
- Cost of buying rights to drill is \$2 million
- Seismologist says for a fee will survey one of 5 locations and report back definitively whether that location does or does not contain oil
- Value of information: expected profit if ask seismologist minus expected profit if don't ask
- Expected profit if don't ask:
  - Guess at random

$$= \frac{1}{5}(10-2) + \frac{4}{5}(0-2) = 0 \tag{2}$$

- Expected profit if ask:
  - If one surveyed has oil, expected profit is: $10-2=8$
  - If one surveyed doesn't have oil, expected profit: (guess at random from other locations) $\frac{1}{4}(10-2) - \frac{3}{4}(-2) = 0.5$
  - Weigh by probability will survey location with oil: $= \frac{1}{5}8 + \frac{4}{5}0.5 = 2$
- VOI: $2-0=2$

Relevant Background: Value of Information

- Back to making a sequence of decisions under uncertainty
- Information gain is higher in uncertain situations
- But need to consider value of that information
  - Would it change our decisions?
  - Expected utility benefit

Information State Space

- So far viewed bandits as a simple fully observable Markov decision process (where actions don't impact next state)
- Beautiful idea: frame bandits as a partially observable Markov decision process where the hidden state is the mean reward of each arm

Information State Space

- So far viewed bandits as a simple fully observable Markov decision process (where actions don't impact next state)
- Beautiful idea: frame bandits as a partially observable Markov decision process where the hidden state is the mean reward of each arm
- (Hidden) State is static
- Actions are same as before, pulling an arm
- Observations: Sample from reward model given hidden state
- POMDP planning = Optimal Bandit learning

Information State Space

- POMDP belief state / information state $\tilde{s}$ is posterior over hidden parameters (e.g. mean reward of each arm)
- $\tilde{s}$ is a statistic of the history, $\tilde{s} = f(h_t)$
- Each action a causes a transition to a new information state $\tilde{s}'$ (by adding information), with probability $\tilde{\mathcal{P}}^a_{\tilde{s},\tilde{s}'}$
- Equivalent to a POMDP
- Or a MDP $\tilde{\mathcal{M}} = (\tilde{\mathcal{S}}, \mathcal{A}, \tilde{\mathcal{P}}, \mathcal{R}, \gamma)$ in augmented information state space

Bernoulli Bandits

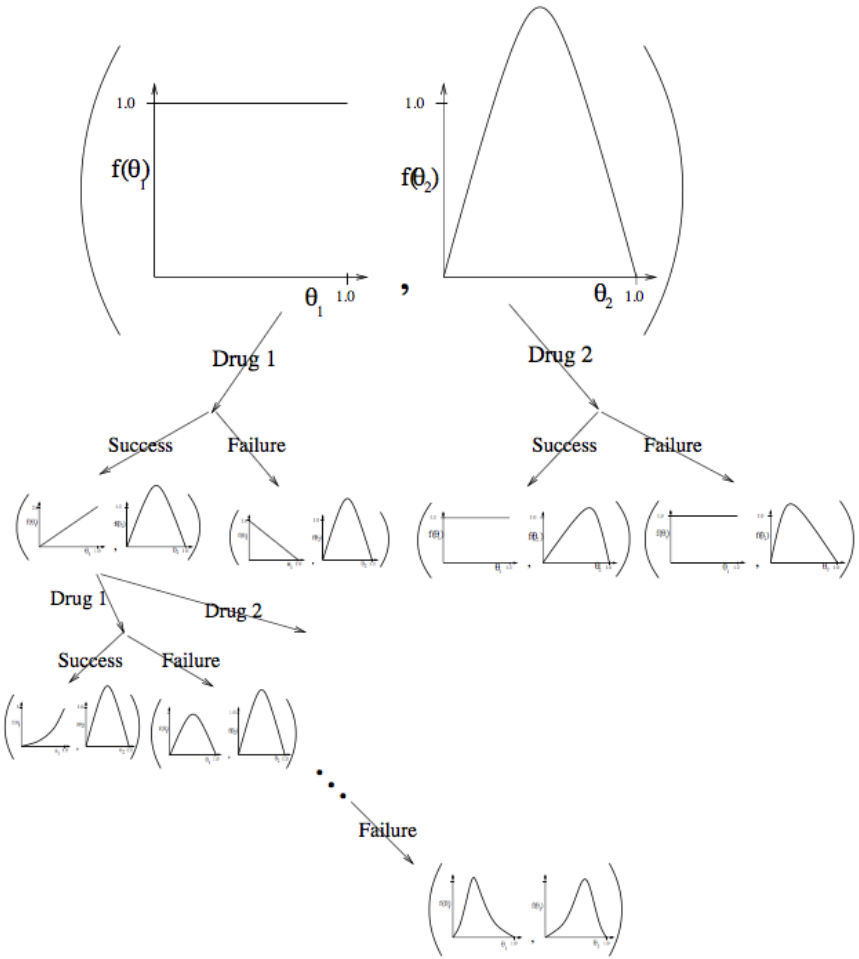
- Consider a Bernoulli bandit such that $\mathcal{R}^a = \mathcal{B}(\mu_a)$
- e.g. Win or lose a game with probability μ_a
- Want to find which arm has the highest μ_a
- The information state is $\tilde{s} = (\alpha, \beta)$
  - α_a counts the pulls of arm a where the reward was 0
  - β_a counts the pulls of arm a where the reward was 1

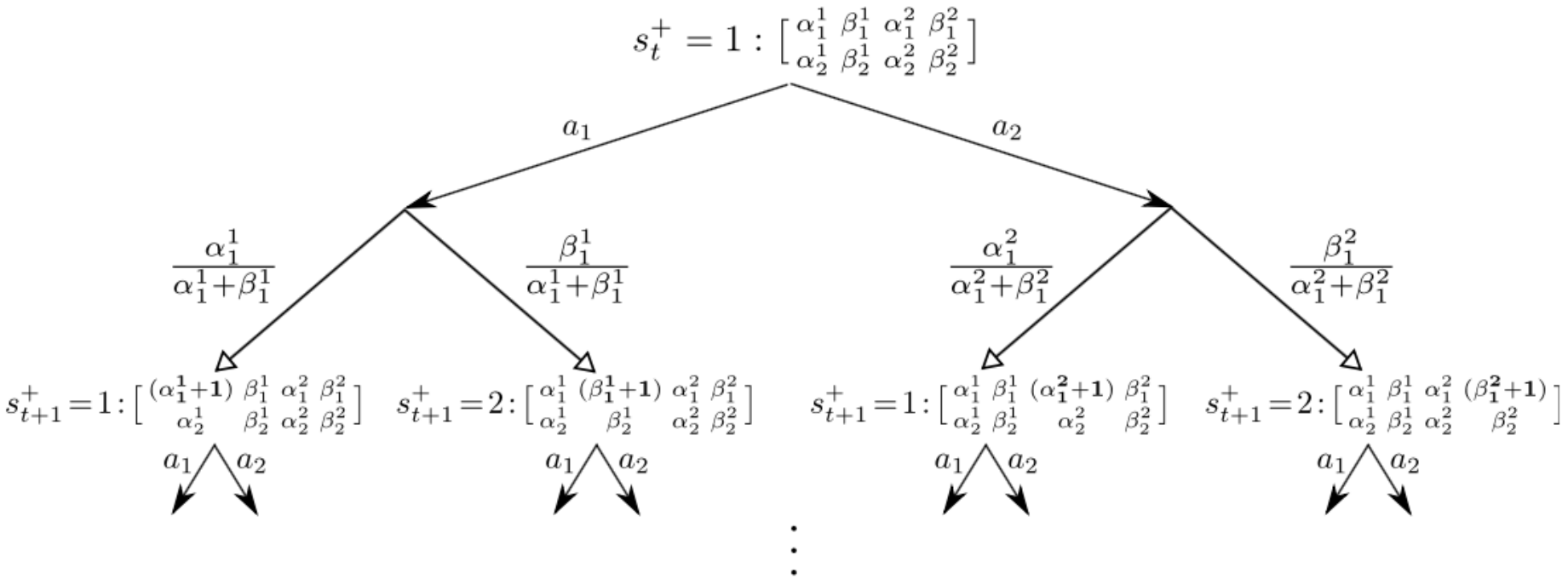
Solving Information State Space Bandits

- We now have an infinite MDP over information states
- This MDP can be solved by reinforcement learning
- Model-free reinforcement learning (e.g. Q-learning)
- Bayesian model-based RL (e.g. Gittins indices)
- This approach is known as Bayes-adaptive RL: Finds Bayes-optimal exploration/exploitation trade-off with respect to prior distribution
- In other words, selects actions that maximize expected reward given information have so far
- Check your understanding: Can an algorithm that optimally solves an information state bandit have a non-zero regret? Why or why not?

Bayes-Adaptive Bernoulli Bandits

- Start with $\text{Beta}(\alpha_a, \beta_a)$ prior over reward function $\mathcal{R}^a$
- Each time a is selected, update posterior for $\mathcal{R}^a$
  - $\text{Beta}(\alpha_a + 1, \beta_a)$ if $r = 0$
  - $\text{Beta}(\alpha_a, \beta_a + 1)$ if $r = 1$
- This defines transition function $\tilde{\mathcal{P}}$ for the Bayes-adaptive MDP
- Information state (α, β) corresponds to reward model $\text{Beta}(\alpha, \beta)$
- Each state transition corresponds to a Bayesian model update




$s_t^+ = 1 : \begin{bmatrix} \alpha_1^1 & \beta_1^1 & \alpha_1^2 & \beta_1^2 \\ \alpha_2^1 & \beta_2^1 & \alpha_2^2 & \beta_2^2 \end{bmatrix}$
a_1
a_2
$\frac{\alpha_1^1}{\alpha_1^1+\beta_1^1}$
$\frac{\beta_1^1}{\alpha_1^1+\beta_1^1}$
$\frac{\alpha_1^2}{\alpha_1^2+\beta_1^2}$
$\frac{\beta_1^2}{\alpha_1^2+\beta_1^2}$
$s_{t+1}^+ = 1 : \begin{bmatrix} (\alpha_1^1+1) & \beta_1^1 & \alpha_1^2 & \beta_1^2 \\ \alpha_2^1 & \beta_2^1 & \alpha_2^2 & \beta_2^2 \end{bmatrix}$
$s_{t+1}^+ = 2 : \begin{bmatrix} \alpha_1^1 & (\beta_1^1+1) & \alpha_1^2 & \beta_1^2 \\ \alpha_2^1 & \beta_2^1 & \alpha_2^2 & \beta_2^2 \end{bmatrix}$
$s_{t+1}^+ = 1 : \begin{bmatrix} \alpha_1^1 & \beta_1^1 & (\alpha_1^2+1) & \beta_1^2 \\ \alpha_2^1 & \beta_2^1 & \alpha_2^2 & \beta_2^2 \end{bmatrix}$
$s_{t+1}^+ = 2 : \begin{bmatrix} \alpha_1^1 & \beta_1^1 & \alpha_1^2 & (\beta_1^2+1) \\ \alpha_2^1 & \beta_2^1 & \alpha_2^2 & \beta_2^2 \end{bmatrix}$
a_1 a_2
a_1 a_2
a_1 a_2
a_1 a_2

Gittins Indices for Bernoulli Bandits

- Bayes-adaptive MDP can be solved by dynamic programming
- The solution is known as the Gittins index
- Exact solution to Bayes-adaptiev MDP is typically intractable; information state space is too large
- Recent idea: apply simulation-based search (Guez et al. 2012, 2013)
  - Forward search in information state space
  - Using simulations from current information state

Table of Contents

Principles for Strategic Exploration

- The sample principles for exploration/exploitation apply to MDPs
  - Naive Exploration
  - Optimistic Initialization
  - Optimism in the Face of Uncertainty
  - Probability Matching
  - Information State Search

Optimistic Initialization: Model-Free RL

- Initialize action-value function Q(s,a) to $\frac{r_{max}}{1-\gamma}$
- Run favorite model-free RL algorithm
  - Monte-Carlo control
  - Sarsa
  - Q-learning
  - etc.
- Encourages systematic exploration of states and actions

Optimistic Initialization: Model-Based RL

- Construct an **optimistic** model of the MDP
- Initialize transitions to go to terminal state with r_{max} reward
- Solve optimistic MDP by favorite planning algorithm
- Encourages systematic exploration of states and actions
- e.g. RMax algorithm (Brafman and Tennenholtz)

UCB: Model-Based RL

- Maximize UCB on action-value function $Q^{\pi}(s, a)$

$$a_t = \arg\max_{a \in \mathcal{A}} Q(s_t, a) + U(s_t, a)$$

  - Estimate uncertainty in policy evaluation (easy)
  - Ignores uncertainty from policy improvement
- Maximize UCB on optimal action-value function $Q^*(s, a)$

$$a_t = \arg\max_{a \in \mathcal{A}} Q(s_t, a) + U_1(s_t, a) + U_2(s_t, a)$$

  - Estimate uncertainty in policy evaluation (easy)
  - plus uncertainty from policy improvement (hard)

Bayesian Model-Based RL

- Maintain posterior distribution over MDP models
- Estimate both transition and rewards, $p[\mathcal{P}, \mathcal{R} \mid h_t]$, where $h_t = (s_1, a_1, r_1, \ldots, s_t)$ is the history
- Use posterior to guide exploration
  - Upper confidence bounds (Bayesian UCB)
  - Probability matching (Thompson sampling)

Thompson Sampling: Model-Based RL

- Thompson sampling implements probability matching

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

- Use Bayes law to compute posterior distribution $p[\mathcal{P}, \mathcal{R} \mid h_t]$
- **Sample** an MDP $\mathcal{P}, \mathcal{R}$ from posterior
- Solve MDP using favorite planning algorithm to get $Q^*(s, a)$
- Select optimal action for sample MDP, $a_t = \arg\max_{a \in \mathcal{A}} Q^*(s_t, a)$

Information State Search in MDPs

- MDPs can be augmented to include information state
- Now the augmented state is $(s, \tilde{s})$
  - where s is original state within MDP
  - and $\tilde{s}$ is a statistic of the history (accumulated information)
- Each action a causes a transition
  - to a new state s' with probability $\mathcal{P}^a_{s,s'}$
  - to a new information state $\tilde{s}'$
- Defines MDP $\tilde{\mathcal{M}}$ in augmented information state space

Bayes Adaptive MDP

- Posterior distribution over MDP model is an information state

$$\tilde{s}_t = \mathbb{P}[\mathcal{P}, \mathcal{R} \mid h_t]$$

- Augmented MDP over $(s, \tilde{s})$ is called **Bayes-adaptive MDP**
- Solve this MDP to find optimal exploration/exploitation trade-off (with respect to prior)
- However, Bayes-adaptive MDP is typically enormous
- Simulation-based search has proven effective (Guez et al, 2012, 2013)

Table of Contents

Principles

- Naive Exploration
  - Add noise to greedy policy (e.g. ϵ-greedy)
- Optimistic Initialization
  - Assume the best until proven otherwise
- Optimism in the Face of Uncertainty
  - Prefer actions with uncertain values
- Probability Matching
  - Select actions according to probability they are best
- Information State Search
  - Lookahead search incorporating value of information

Generalization and Strategic Exploration

- Active area of ongoing research: combine generalization & strategic exploration
- Many approaches are grounded by principles outlined here
- Some examples:
  - Optimism under uncertainty: Bellemare et al. NIPS 2016; Ostrovski et al. ICML 2017; Tang et al. NIPS 2017
  - Probability matching: Osband et al. NIPS 2016; Mandel et al. IJCAI 2016

Table of Contents

Performance Criteria of RL Algorithms

- Empirical performance
- Convergence (to something ...)
- Asymptotic convergence to optimal policy
- Finite sample guarantees: probably approximately correct
- Regret (with respect to optimal decisions)
- Optimal decisions given information have available
- PAC uniform (Dann, Tor, Brunskill NIPS 2017): stronger criteria, directly provides both PAC and regret bounds

Summary: What You Are Expected to Know

- Define the tension of exploration and exploitation in RL and why this does not arise in supervised or unsupervised learning
- Be able to define and compare different criteria for "good" performance (empirical, convergence, asymptotic, regret, PAC)
- Be able to map algorithms discussed in detail in class to the performance criteria they satisfy

Class Structure

- Last time: Exploration and Exploitation Part I
- **This time: Exploration and Exploitation Part II**
- Next time: Batch RL