Lecture 11: Fast Reinforcement Learning

Emma Brunskill

CS234 Reinforcement Learning

- Slides from or derived from David Silver, Examples new.

L11N1 Refresh Your Knowledge.

- Importance sampling leverages the Markov assumption to improve accuracy
  1. True
  2. False.
  3. Not sure
- We can use the performance difference lemma / relative policy performance to: (Select all that are true)
  1. Bound the difference in value between two policies using the advantage function of one policy, and samples from the other policy
  2. Approximately bound the difference in value between two policies using the advantage function of policy 1, importance weights between the two policies, and samples from policy 1
  3. The approximation error in the relative policy performance bounds is bounded by the KL divergence between the states visited under one policy, vs the other
  4. These ideas are used in PPO
  5. Not sure

L11N1 Refresh Your Knowledge. Answers

- Importance sampling leverages the Markov assumption to improve accuracy
  1. True
  2. False.
  3. Not sure
  4. False.
- We can use the performance difference lemma / relative policy performance to: (Select all that are true)
  1. Bound the difference in value between two policies using the advantage function of one policy, and samples from the other policy
  2. Approximately bound the difference in value between two policies using the advantage function of policy 1, importance weights between the two policies, and samples from policy 1
  3. The approximation error in the relative policy performance bounds is bounded by the KL divergence between the states visited under one policy, vs the other
  4. These ideas are used in PPO

Answer: Importance sampling does not use the Markov assumption. For the second question, 1, 2 and 4 are true. The approximation error is bounded by the average (over the states visited by one policy) of KL divergence between the two policies.

Class Structure

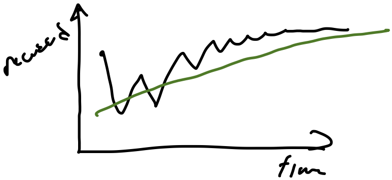
- Last time: Learning from past data
- **This time: Data Efficient Reinforcement Learning – Bandits**
- Next time: Data Efficient Reinforcement Learning

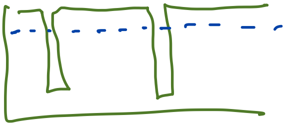
Computational Efficiency and Sample Efficiency

Computational Efficiency	Sample Efficiency (data)
Atari mujoco	mobile phones for health interventions consumer marketing educational tech environmental policies

Evaluation Criteria

- How do we evaluate how "good" an algorithm is?
- If converges? deadly triad not guaranteed
- If converges to optimal policy?
- How quickly reaches optimal policy? how much data
- Mistakes make along the way?
- Will introduce different measures to evaluate RL algorithms





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

Today

- Setting: Introduction to multi-armed bandits & Approach: greedy methods
- Framework: Regret
- Approach: ϵ-greedy methods
- Approach: Optimism under uncertainty
- Framework: Bayesian regret
- Approach: Probability matching / Thompson sampling

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 $\sum_{\tau=1}^{t} r_\tau$

Toy Example: Ways to Treat Broken Toes

- 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 / reward is binary variable: whether the toe has healed (+1) or not healed (0) after 6 weeks, as assessed by x-ray

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

L11N2 Check Your Understanding: Bandit Toes

- Consider deciding how to best treat patients with broken toes
- Imagine have 3 common options: (1) surgery (2) buddy taping the broken toe with another toe (3) doing nothing
- 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
- Select all that are true
  1. Pulling an arm / taking an action corresponds to whether the toe has healed or not
  2. A multi-armed bandit is a better fit to this problem than a MDP because treating each patient involves multiple decisions
  3. After treating a patient, if $\theta_i \neq 0$ and $\theta_i \neq 1$ $\forall i$ sometimes a patient's toe will heal and sometimes it may not
  4. Not sure

L11N2 Check Your Understanding: Bandit Toes Solution

- Consider deciding how to best treat patients with broken toes
- Imagine have 3 common options: (1) surgery (2) buddy taping the broken toe with another toe (3) doing nothing
- 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
- Select all that are true
  1. Pulling an arm / taking an action corresponds to whether the toe has healed or not
  2. A multi-armed bandit is a better fit to this problem than a MDP because treating each patient involves multiple decisions
  3. After treating a patient, if $\theta_i \neq 0$ and $\theta_i \neq 1$ $\forall i$ sometimes a patient's toe will heal and sometimes it may not
  4. Not sure

  3 is true. Pulling an arm corresponds to treating a patient. A MAB is a better fit than a MDP, because actions correspond to treating a patient, and the treatment of one patient does not influence that next patient that comes to be treated.

Greedy Algorithm

- We consider algorithms that estimate $\hat{Q}_t(a) \approx Q(a) = \mathbb{E}[R(a)]$
- Estimate the value of each action by Monte-Carlo evaluation

$$\hat{Q}_t(a) = \frac{1}{N_t(a)} \sum_{i=1}^{t-1} r_i \mathbb{1}(a_i = a)$$

- The **greedy** algorithm selects the action with highest value

$$a_t^* = \arg\max_{a \in \mathcal{A}} \hat{Q}_t(a)$$

Toy Example: Ways to Treat Broken Toes

- Imagine true (unknown) Bernoulli reward parameters for each arm (action) are
  - surgery: $Q(a^1) = \theta_1 = .95$
  - buddy taping: $Q(a^2) = \theta_2 = .9$
  - doing nothing: $Q(a^3) = \theta_3 = .1$

Toy Example: Ways to Treat Broken Toes, Greedy

- Imagine true (unknown) Bernoulli reward parameters for each arm (action) are
  - surgery: $Q(a^1) = \theta_1 = .95$
  - buddy taping: $Q(a^2) = \theta_2 = .9$
  - doing nothing: $Q(a^3) = \theta_3 = .1$
- Greedy
  1. Sample each arm once
     - Take action a^1 ($r \sim$Bernoulli(0.95)), get 0, $\hat{Q}(a^1) = 0$
     - Take action a^2 ($r \sim$Bernoulli(0.90)), get $+1$, $\hat{Q}(a^2) = 1$
     - Take action a^3 ($r \sim$Bernoulli(0.1)), get 0, $\hat{Q}(a^3) = 0$
  2. What is the probability of greedy selecting each arm next? Assume ties are split uniformly.

     $p(a_2) = 1$

Toy Example: Ways to Treat Broken Toes, Greedy

- Imagine true (unknown) Bernoulli reward parameters for each arm (action) are
  - surgery: $Q(a^1) = \theta_1 = .95$
  - buddy taping: $Q(a^2) = \theta_2 = .9$
  - doing nothing: $Q(a^3) = \theta_3 = .1$
- Greedy
  1. Sample each arm once
     - Take action a^1 ($r \sim$Bernoulli(0.95)), get 0, $\hat{Q}(a^1) = 0$
     - Take action a^2 ($r \sim$Bernoulli(0.90)), get $+1$, $\hat{Q}(a^2) = 1$
     - Take action a^3 ($r \sim$Bernoulli(0.1)), get 0, $\hat{Q}(a^3) = 0$
  2. Will the greedy algorithm ever find the best arm in this case? no

Greedy Algorithm

- We consider algorithms that estimate $\hat{Q}_t(a) \approx Q(a) = \mathbb{E}[R(a)]$
- Estimate the value of each action by Monte-Carlo evaluation

$$\hat{Q}_t(a) = \frac{1}{N_t(a)} \sum_{t=1}^{T} r_t \mathbb{1}(a_t = a)$$

- The **greedy** algorithm selects the action with highest value

$$a_t^* = \arg\max_{a \in \mathcal{A}} \hat{Q}_t(a)$$

- **Greedy can lock onto suboptimal action, forever**

Today

- Setting: Introduction to multi-armed bandits & Approach: greedy methods
- **Framework: Regret**
- Approach: ϵ-greedy methods
- Approach: Optimism under uncertainty
- Framework: Bayesian regret
- Approach: Probability matching / Thompson sampling

Assessing the Performance of Algorithms

- How do we evaluate the quality of a RL (or bandit) algorithm?
- So far: computational complexity, convergence, convergence to a fixed point, & empirical performance performance
- Today: introduce a formal measure of how well a RL/bandit algorithm will do in any environment, compared to optimal

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

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

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 $\iff$ minimize total regret

Evaluating Regret

- **Count** $N_t(a)$ is number of times action a has been selected at time step t
- **Gap** Δ_a is the difference in value between action a and optimal action a^*, $\Delta_i = V^* - Q(a_i)$ advantage of a^* over a
- Regret is a function of gaps and counts

$$
\begin{aligned}
L_t &= \mathbb{E}\left[\sum_{\tau=1}^{t} V^* - Q(a_\tau)\right] \\
&= \sum_{a \in \mathcal{A}} \mathbb{E}[N_t(a)](V^* - Q(a)) \\
&= \sum_{a \in \mathcal{A}} \mathbb{E}[N_t(a)]\Delta_a
\end{aligned}
$$

- A good algorithm ensures small counts for large gap,s but gaps are not known

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

- True (unknown) Bernoulli reward parameters for each arm (action) are
  - surgery: $Q(a^1) = \theta_1 = .95$
  - buddy taping: $Q(a^2) = \theta_2 = .9$
  - doing nothing: $Q(a^3) = \theta_3 = .1$
- Greedy

t	Action	Optimal Action	Observed Reward	Regret
1	a^1	a^1	0	0
2	a^2	a^1	1	.95-.9=.05
3	a^3	a^1	0	.95-.1=.85
4	a^2	a^1	1	
5	a^2	a^1	0	

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

- True (unknown) Bernoulli reward parameters for each arm (action) are
  - surgery: $Q(a^1) = \theta_1 = .95$
  - buddy taping: $Q(a^2) = \theta_2 = .9$
  - doing nothing: $Q(a^3) = \theta_3 = .1$
- Greedy

Action	Optimal Action	Observed Reward	Regret
a^1	a^1	0	0
a^2	a^1	1	0.05
a^3	a^1	0	0.85
a^2	a^1	1	0.05
a^2	a^1	0	0.05

- Regret for greedy methods can be **linear** in the number of decisions made (timestep)

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

- Greedy

Action	Optimal Action	Observed Reward	Regret
a^1	a^1	0	0
a^2	a^1	1	0.05
a^3	a^1	0	0.85
a^2	a^1	1	0.05
a^2	a^1	0	0.05

- **Note: in real settings we cannot evaluate the regret because it requires knowledge of the expected reward of the true best action.**
- Instead we can prove an upper bound on the potential regret of an algorithm in **any bandit** problem

Today

- Setting: Introduction to multi-armed bandits & Approach: greedy methods
- Framework: Regret
- **Approach: ϵ-greedy methods**
- Approach: Optimism under uncertainty
- Framework: Bayesian regret
- Approach: Probability matching / Thompson sampling

ϵ-Greedy Algorithm

- The ϵ-**greedy** algorithm proceeds as follows:
  - With probability $1 - \epsilon$ select $a_t = \arg\max_{a \in \mathcal{A}} \hat{Q}_t(a)$
  - With probability ϵ select a random action
- Always will be making a sub-optimal decision ϵ fraction of the time
- Already used this in prior homeworks

Toy Example: Ways to Treat Broken Toes, ϵ-**Greedy**

- Imagine true (unknown) Bernoulli reward parameters for each arm (action) are
  - surgery: $Q(a^1) = \theta_1 = .95$
  - buddy taping: $Q(a^2) = \theta_2 = .9$
  - doing nothing: $Q(a^3) = \theta_3 = .1$
- ϵ-greedy
  1. Sample each arm once
     - Take action a^1 ($r \sim$Bernoulli(0.95)), get $+1$, $\hat{Q}(a^1) = 1$
     - Take action a^2 ($r \sim$Bernoulli(0.90)), get $+1$, $\hat{Q}(a^2) = 1$
     - Take action a^3 ($r \sim$Bernoulli(0.1)), get 0, $\hat{Q}(a^3) = 0$
  2. Let $\epsilon = 0.1$
  3. What is the probability ϵ-greedy will pull each arm next? Assume ties are split uniformly. 90% prob greedy a_1 & a_2 each 45%

     10% 3.3% a_1, a_2, a_3

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

- True (unknown) Bernoulli reward 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	

- Will ϵ-greedy ever select a^3 again? If ϵ is fixed, how many times will each arm be selected?

Recall: Bandit Regret

- **Count** $N_t(a)$ is expected number of selections for action a
- **Gap** Δ_a is the difference in value between action a and optimal action a^*, $\Delta_i = V^* - Q(a_i)$
- Regret is a function of gaps and counts

$$\begin{aligned} L_t &= \mathbb{E}\left[\sum_{\tau=1}^{t} V^* - Q(a_\tau)\right] \\ &= \sum_{a \in \mathcal{A}} \mathbb{E}[N_t(a)](V^* - Q(a)) \\ &= \sum_{a \in \mathcal{A}} \mathbb{E}[N_t(a)]\Delta_a \end{aligned}$$

- A good algorithm ensures small counts for large gap, but gaps are not known

L11N3 Check Your Understanding: ϵ-greedy Bandit Regret

- **Count** $N_t(a)$ is expected number of selections for action a
- **Gap** Δ_a is the difference in value between action a and optimal action a^*, $\Delta_i = V^* - Q(a_i)$
- Regret is a function of gaps and counts $= \sum_a \frac{\epsilon}{|A|} T \Delta_a + \dots$

$$L_t = \sum_{a \in \mathcal{A}} \mathbb{E}[N_t(a)] \Delta_a$$

- Informally an algorithm has linear regret if it takes a non-optimal action a constant fraction of the time
- Assume $\exists a \; s.t. \; \Delta_a > 0$
- Select all
  1. $\epsilon = 0.1$ ϵ-greedy can have linear regret
  2. $\epsilon = 0$ ϵ-greedy can have linear regret
  3. Not sure

both are true

L11N3 Check Your Understanding: ϵ-greedy Bandit Regret Answer

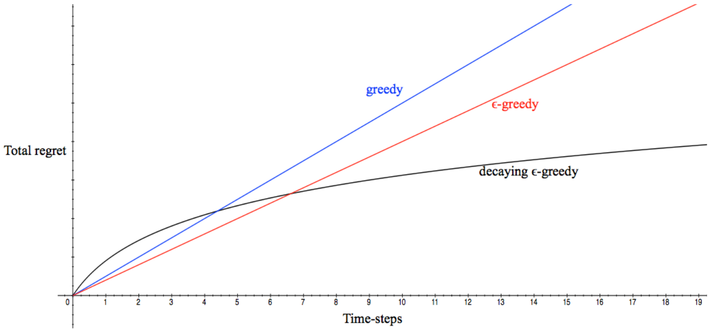
- **Count** $N_t(a)$ is expected number of selections for action a
- **Gap** Δ_a is the difference in value between action a and optimal action a^*, $\Delta_i = V^* - Q(a_i)$
- Regret is a function of gaps and counts

$$L_t = \sum_{a \in \mathcal{A}} \mathbb{E}[N_t(a)]\Delta_a$$

- Informally an algorithm has linear regret if it takes a non-optimal action a constant fraction of the time
- Assume $\exists a \; s.t. \; \Delta_a > 0$
- Select all
  1. $\epsilon = 0.1$ ϵ-greedy can have linear regret
  2. $\epsilon = 0$ ϵ-greedy can have linear regret
  3. Not sure

Both can have linear regret.

"Good": Sublinear or below regret



- **Explore forever**: have linear total regret
- **Explore never**: have linear total regret
- Is it possible to achieve sublinear (in the time steps/number of decisions made) regret?

Types of Regret bounds

- **Problem independent**: Bound how regret grows as a function of T, the total number of time steps the algorithm operates for
- **Problem dependent**: Bound regret as a function of the number of times we pull each arm and the gap between the reward for the pulled arm and a^*

Lower Bound

- Use lower bound to determine how hard this problem is
- The performance of any algorithm is determined by similarity between optimal arm and other arms
- Hard problems have similar looking arms with different means
- This is described formally by the gap Δ_a and the similarity in distributions $D_{KL}(\mathcal{R}^a \| \mathcal{R}^{a^*})$
- Theorem (Lai and Robbins): Asymptotic total regret is at least logarithmic in number of steps

$$\lim_{t \to \infty} L_t \geq \log t \sum_{a | \Delta_a > 0} \frac{\Delta_a}{D_{KL}(\mathcal{R}^a \| \mathcal{R}^{a^*})}$$

- Promising in that lower bound is sublinear

Today

- Setting: Introduction to multi-armed bandits & Approach: greedy methods
- Framework: Regret
- Approach: ϵ-greedy methods
- **Approach: Optimism under uncertainty**
- Framework: Bayesian regret
- Approach: Probability matching / Thompson sampling

Approach: Optimism in the Face of Uncertainty

- Choose actions that ~~that~~ might have a high value
- Why?
- Two outcomes:

  – get high reward
  – learn something

Approach: Optimism in the Face of Uncertainty

uncertainty

- Choose actions that that might have a high value
- Why?
- Two outcomes:
  - Getting high reward: if the arm really has a high mean reward
  - Learn something: if the arm really has a lower mean reward, pulling it will (in expectation) reduce its average reward and the uncertainty over its value

Upper Confidence Bounds

- 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)

algorithms

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

Hoeffding's Inequality

- Theorem (Hoeffding's Inequality): Let $X_1, \ldots, X_n$ be i.i.d. random variables in $[0, 1]$, and let $\bar{X}_n = \frac{1}{n}\sum_{\tau=1}^{n} X_\tau$ be the sample mean. Then

$$\mathbb{P}\left[\mathbb{E}\left[X\right] > \bar{X}_n + u\right] \leq \exp(-2nu^2)$$

want CI to hold with $1-\delta$ prob

$$P(|E(X) - \bar{X}_n| > u) \leq 2\exp(-2nu^2) = \delta$$

$$\exp(-2nu^2) = \delta/2$$

$$u^2 = 1/n \log 2/\delta$$

$$u = \sqrt{\frac{\log 2/\delta}{n}}$$

$$\bar{X}_n - u \leq E[X] \leq \bar{X}_n + u$$

w/prob $\geq 1-\delta$

UCB Bandit Regret

Auer 2002?

- This leads to the UCB1 algorithm

empirical avg

$$a_t = \arg\max_{a \in \mathcal{A}} \left[\hat{Q}(a) + \sqrt{\frac{2 \log \frac{1}{\delta}}{N_t(a)}} \right]$$

# of samples of a after t time steps

Toy Example: Ways to Treat Broken Toes, Thompson Sampling[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: $Q(a^3) = \theta_3 = .1$
- Optimism under uncertainty, UCB1 (Auer, Cesa-Bianchi, Fischer 2002)
  1. Sample each arm once

[1] 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[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: $Q(a^3) = \theta_3 = .1$
- UCB1 (Auer, Cesa-Bianchi, Fischer 2002)
  1. Sample each arm once
     - Take action a^1 ($r \sim$Bernoulli(0.95)), get $+1$, $\hat{Q}(a^1) = 1$
     - Take action a^2 ($r \sim$Bernoulli(0.90)), get $+1$, $\hat{Q}(a^2) = 1$
     - Take action a^3 ($r \sim$Bernoulli(0.1)), get 0, $\hat{Q}(a^3) = 0$

[1]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[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: $Q(a^3) = \theta_3 = .1$
- UCB1 (Auer, Cesa-Bianchi, Fischer 2002)
  1. Sample each arm once
     - Take action a^1 ($r \sim$Bernoulli(0.95)), get $+1$, $\hat{Q}(a^1) = 1$
     - Take action a^2 ($r \sim$Bernoulli(0.90)), get $+1$, $\hat{Q}(a^2) = 1$
     - Take action a^3 ($r \sim$Bernoulli(0.1)), get 0, $\hat{Q}(a^3) = 0$
  2. Set $t = 3$, Compute upper confidence bound on each action

$$UCB(a) = \hat{Q}(a) + \sqrt{\frac{2 \log \frac{1}{\delta}}{N_t(a)}}$$

$UCB(a_1)$ $1 + \sqrt{\frac{2 \log 1/\delta}{1}}$

a_2 ''

a_3 $0 + \sqrt{\frac{2 \log 1/\delta}{1}}$

[1] 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[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: $Q(a^3) = \theta_3 = .1$
- UCB1 (Auer, Cesa-Bianchi, Fischer 2002)
  1. Sample each arm once
     - Take action a^1 ($r \sim$Bernoulli(0.95)), get $+1$, $\hat{Q}(a^1) = 1$
     - Take action a^2 ($r \sim$Bernoulli(0.90)), get $+1$, $\hat{Q}(a^2) = 1$
     - Take action a^3 ($r \sim$Bernoulli(0.1)), get 0, $\hat{Q}(a^3) = 0$
  2. Set $t = 3$, Compute upper confidence bound on each action

$$UCB(a) = \hat{Q}(a) + \sqrt{\frac{2 \log \frac{1}{\delta}}{N_t(a)}}$$

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

$a = 1$

$UCB(a_1) = 1 + \sqrt{\frac{2 \log 1/\delta}{2}}$

$UCB(a_2) = 1 + \sqrt{\frac{2 \log 1/\delta}{1}}$

$UCB(a_3) = 0 + \sqrt{\frac{2 \log 1/\delta}{1}}$

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

Toy Example: Ways to Treat Broken Toes, Optimism[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: $Q(a^3) = \theta_3 = .1$
- UCB1 (Auer, Cesa-Bianchi, Fischer 2002)
  1. Sample each arm once
     - Take action a^1 ($r \sim$Bernoulli(0.95)), get $+1$, $\hat{Q}(a^1) = 1$
     - Take action a^2 ($r \sim$Bernoulli(0.90)), get $+1$, $\hat{Q}(a^2) = 1$
     - Take action a^3 ($r \sim$Bernoulli(0.1)), get 0, $\hat{Q}(a^3) = 0$
  2. Set $t = 3$, Compute upper confidence bound on each action

$$UCB(a) = \hat{Q}(a) + \sqrt{\frac{2 \log \frac{1}{\delta}}{N_t(a)}}$$

  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

[1] 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	

Confidence Level δ

$\log 1/\delta$

$\log \frac{T|A|}{\delta}$

- Subtle
- If there are a fixed number of time steps T for the problem setting, can set $\delta = \frac{\delta}{T} |A|$
  - Union bound: $P(\cup E_i) \leq \sum_i P(E_i)$
- Often want to do this in other settings

Regret Bound for UCB Multi-armed Bandit Sketch

- Any sub-optimal arm $a \neq a^*$ is pulled by UCB at most $\mathbb{E}N_T(a) \leq C' \frac{\log \frac{1}{\delta}}{\Delta_a^2} + \frac{\pi^2}{3} + 1$.

  So the regret of UCB is bounded by $\sum_a \Delta_a \mathbb{E}N_T(a) \leq \sum_a C' \frac{\log T}{\Delta_a} + |A|(\frac{\pi^2}{3} + 1)$.

  (Arm means $\in [0, 1]$)

Bandit Algorithms for Lattimore Csaba Szepesvari chp 7

true value   empirical   UCB   (loose with the δs)

$$P\left(|Q(a) - \hat{Q}_t(a)| \geq \sqrt{\frac{C\log\frac{1}{\delta}}{N_t(a)}}\right) \leq \frac{\delta}{T} \qquad (1)$$

times we pull $a \neq a^*$ and $\Delta a \neq 0$

if (1) holds $\quad Q(a) - \sqrt{\frac{C\log 1/\delta}{N_t(a)}} \leq \hat{Q}_t(a) \leq Q(a) + \sqrt{\frac{C\log 1/\delta}{N_t(a)}} \quad (2)$

if (1) holds

under UCB algorithm $\quad UCB(a) > UCB(a^*)$

$$\hat{Q}_t(a) + \sqrt{\frac{C\log 1/\delta}{N_t(a)}} > \hat{Q}_t(a^*) + \sqrt{\frac{C\log 1/\delta}{N_t(a^*)}} \quad (3)$$

$$> Q(a^*)$$

substitute in from (2)

$$Q(a) + \sqrt{\frac{C\log 1/\delta}{N_t(a)}} \cdot 2 > Q(a^*)$$

$$2\sqrt{\frac{C\log 1/\delta}{N_t(a)}} > Q(a^*) - Q(a) = \Delta a$$

Regret Bound for UCB Multi-armed Bandit Sketch

- Any sub-optimal arm $a \neq a^*$ is pulled by UCB at most $\mathbb{E}N_T(a) \leq C' \frac{\log \frac{1}{\delta}}{\Delta_a^2} + \frac{\pi^2}{3} + 1$. So the regret of UCB is bounded by $\sum_a \Delta_a \mathbb{E}N_T(a) \leq \sum_a C' \frac{\log T}{\Delta_a} + |A|(\frac{\pi^2}{3} + 1)$. (Arm means $\in [0, 1]$)

$$Q(a) - \sqrt{\frac{C \log \frac{1}{\delta}}{N_t(a)}} \leq \hat{Q}_t(a) \leq Q(a) + \sqrt{\frac{C \log \frac{1}{\delta}}{N_t(a)}} \tag{2}$$

$$\hat{Q}_t(a) + \sqrt{\frac{C \log \frac{1}{\delta}}{N_t(a)}} \geq \hat{Q}_t(a^*) + \sqrt{\frac{C \log \frac{1}{\delta}}{N_t(a^*)}} \geq Q(a^*) \tag{3}$$

$$Q(a) + 2\sqrt{\frac{C \log \frac{1}{\delta}}{N_t(a)}} \geq Q(a^*) \tag{4}$$

$$2\sqrt{\frac{C \log \frac{1}{\delta}}{N_t(a)}} \geq Q(a^*) - Q(a) = \Delta_a \tag{5}$$

$$4 \frac{C \log 1/\delta}{N_t(a)} \geq \Delta_a^2$$

$$N_t(a) \leq \frac{4 C \log 1/\delta}{\Delta_a^2}$$

$$N_t(a) \leq \frac{4C \log \frac{1}{\delta}}{\Delta_a^2} \tag{6}$$

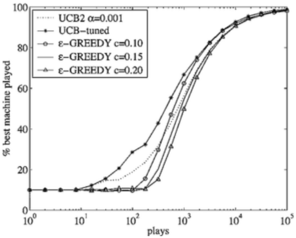
UCB Bandit Regret

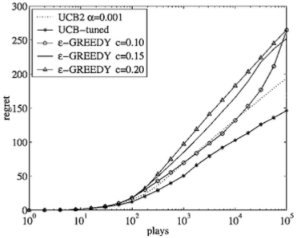
- This leads to the UCB1 algorithm

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

- Theorem: The UCB algorithm achieves logarithmic asymptotic total regret

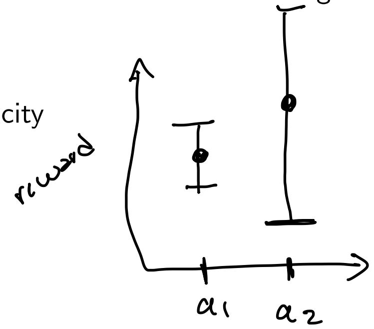
$$\lim_{t \to \infty} L_t \leq 8 \log t \sum_{a | \Delta_a > 0} \frac{1}{\Delta_a}$$





Optional 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



Today

- Setting: Introduction to multi-armed bandits & Approach: greedy methods
- Framework: Regret
- Approach: ϵ-greedy methods
- Approach: Optimism under uncertainty
- Note: bandits are a simpler place to see these ideas, but these ideas will extend to MDPs
- Next time: more fast learning

Lecture 12: Fast Reinforcement Learning

Emma Brunskill

CS234 Reinforcement Learning

- With some slides from or derived from David Silver, Examples new

Refresh Your Understanding: Multi-armed Bandits

- Select all that are true:
  1. Algorithms that minimize regret also maximize reward
  2. Up to variations in constants, ignoring δ, UCB selects the arm with $\arg\max_a \hat{Q}_t(a) + \sqrt{\frac{1}{N_t(a)} \log(f(\delta))}$
  3. Over an infinite trajectory, UCB will sample all arms an infinite number of times
  4. UCB still would likely learn to pull the optimal arm more than other arms if we instead used $\arg\max_a \hat{Q}_t(a) + \sqrt{\frac{1}{\sqrt{N_t(a)}} \log(t/\delta)}$
  5. UCB uses $\arg\max_a \hat{Q}_t(a) + b$ where b is a bonus term. Consider $b = 5$. This will make the algorithm optimistic with respect to the empirical rewards but it may still cause such an algorithm to suffer linear regret.
  6. A k-armed multi-armed bandit is like a single state MDP with k actions
  7. Not Sure

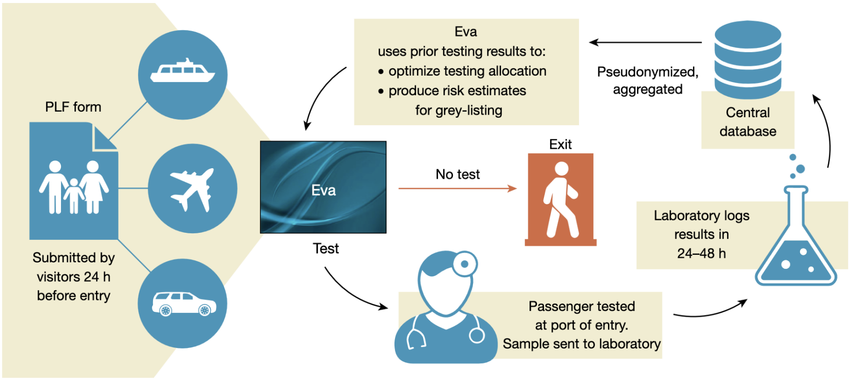
Refresh Your Understanding: Multi-armed Bandits Solution

- Select all that are true:
  1. Algorithms that minimize regret also maximize reward T
  2. Up to variations in constants, ignoring δ, UCB selects the arm with T
     $\arg\max_a \hat{Q}_t(a) + \sqrt{\frac{1}{N_t(a)} \log(f(/\delta))}$
  3. Over an infinite trajectory, UCB will sample all arms an infinite number of times T
  4. UCB still would likely learn to pull the optimal arm more than other T
     arms if we instead used $\arg\max_a \hat{Q}_t(a) + \sqrt{\frac{1}{\sqrt{N_t(a)}} \log(t/\delta)}$
  5. UCB uses $\arg\max_a \hat{Q}_t(a) + b$ where b is a bonus term. Consider $b = 5$. This will make the algorithm optimistic with respect to the empirical F rewards but it may still cause such an algorithm to suffer linear regret.
  6. A k-armed multi-armed bandit is like a single state MDP with k actions
  7. Not Sure T

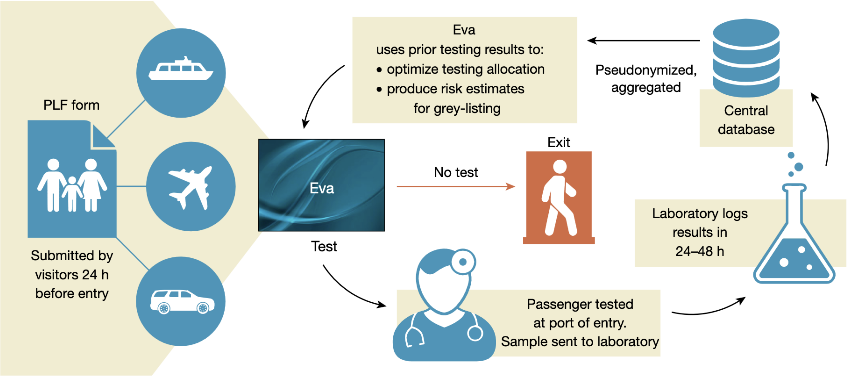
Where We are

- Last time: Bandits and regret and UCB (fast learning)
- This time: Bayesian bandits (fast learning)
- Next time: MDPs (fast learning)

Deciding Who To Test for Covid. Bastani et al. Nature 2021



Deciding Who To Test for Covid. Bastani et al. Nature 2021



- *A nonstationary, contextual, batched bandit problem with delayed feedback and constraints*

Today

- Bandits and Probably Approximately Correct
- Bayesian bandits
- Thompson sampling
- Bayesian Regret

Multiarmed Bandits Notation Recap

- 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

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

- Maximize cumulative reward $\iff$ minimize total regret

Simpler Optimism

- Last time saw UCB, an optimism under uncertainty approach, which has sublinear regret bounds
- Do we need to formally model uncertainty to get the right form of optimism?

Optimistic Initialization with Greedy Bandit Algorithms

- Simple and practical idea: initialize $\hat{Q}(s, a)$ to high value
- Update action value by incremental Monte-Carlo evaluation
- Starting with $N(a) > 0$

$$\hat{Q}_t(a_t) = \hat{Q}_{t-1} + \frac{1}{N_t(a_t)}(r_t - \hat{Q}_{t-1})$$

Optimistic Initialization with Greedy Bandit Algorithms

- Simple and practical idea: initialize $\hat{Q}(s, a)$ to high value
- Update action value by incremental Monte-Carlo evaluation
- Starting with $N(a) > 0$

$$\hat{Q}_t(a_t) = \hat{Q}_{t-1} + \frac{1}{N_t(a_t)}(r_t - \hat{Q}_{t-1})$$

- Encourages systematic exploration early on
- But can still lock onto suboptimal action
- Depends on how high initialize Q
- Check your understanding: What is the downside to initializing Q too high?
- Check your understanding: Is this trivial to do with function approximation? Why or why not?

Optimistic Initialization with Greedy Bandit Algorithms

- Simple and practical idea: initialize Q(a) to high value
- Update action value by incremental Monte-Carlo evaluation
- Starting with $N(a) > 0$

$$\hat{Q}_t(a_t) = \hat{Q}_{t-1} + \frac{1}{N_t(a_t)}(r_t - \hat{Q}_{t-1})$$

- Will turn out that if carefully choose the initialization value, can get good performance
- Under a new measure for evaluating algorithms

Framework: Regret

- 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

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) algorithms
    - on each time step, 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 time steps
- Polynomial in the problem parameters (#actions, ϵ, δ, etc)

Probably Approximately Correct Algorithms

- 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) algorithms
    - on each time step, 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 time steps
- Polynomial in the problem parameters (#actions, ϵ, δ, etc)
- Most PAC algorithms based on optimism or Thompson sampling
- Some PAC algorithms using optimism simply initialize all values to a (specific to the problem) high value

Toy Example: Probably Approximately Correct and Regret

- Surgery: $\phi_1 = .95$ / Taping: $\phi_2 = .9$ / Nothing: $\phi_3 = .1$
- Let $\epsilon = 0.05$
- O = Optimism, TS = Thompson Sampling: W/in $\epsilon = \mathbb{I}(Q(a_t) \geq Q(a^*) - \epsilon)$

O	Optimal	O Regret	O W/in ϵ
a^1	a^1	0	
a^2	a^1	0.05	E optimal
a^3	a^1	0.85	
a^1	a^1	0	
a^2	a^1	0.05	

Greedy Bandit Algorithms vs Optimistic Initialization

- **Greedy**: Linear total regret
- **Constant ϵ-greedy**: Linear total regret
- **Decaying ϵ-greedy**: Sublinear regret but schedule for decaying ϵ requires knowledge of gaps, which are unknown
- **Optimistic initialization**: Sublinear regret if initialize values sufficiently optimistically, else linear regret

Today

- Bandits and Probably Approximately Correct
- **Bayesian Bandits**
- Thompson Sampling
- Bayesian Regret

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, $p[\mathcal{R}]$

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 Bays 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 Bay~~s~~es 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
- Give 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
- Give 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 success/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 success/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 $Beta(\alpha, \beta)$ 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

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 often 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

Initialize prior over each arm a, $p(\mathcal{R}_a)$
for iteration$=1, 2, \ldots$ **do**
    For each arm a **sample** a reward distribution $\mathcal{R}_a$ from posterior
    Compute action-value function $Q(a) = \mathbb{E}[\mathcal{R}_a]$
    $a_t = \arg\max_{a \in \mathcal{A}} Q(a)$
    Observe reward r
    Update posterior $p(\mathcal{R}_a)$ using Bayes Rule
end for

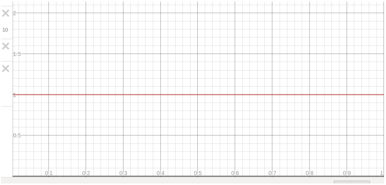
Thompson sampling implements probability matching

- Thompson sampling:

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

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[1]

- 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 $a = \arg\max_{a \in A} Q(a) = \arg\max_{ainA} \theta(a) =$ Do nothing a3

[1] 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 $\theta_i \sim$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 $\theta_i \sim$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

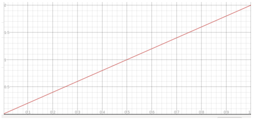
- True (unknown) Bernoulli parameters for each arm/action
- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$
- Thompson sampling:
- Place a prior over each arm's parameter. Here choose $\theta_i \sim$Beta(1,1)
  1. Sample a Bernoulli parameter given current prior over each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

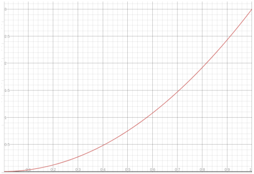
Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- 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 $\theta_i \sim$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 $p(Q(a^1)) = p(\theta(a_1) = Beta(2,1)$



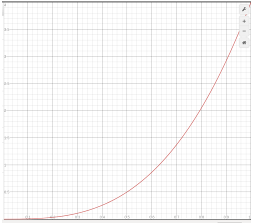
Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- 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 $\theta_i \sim$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 $\theta_i \sim$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
a^1	a^3
a^2	a^1
a^3	a^1
a^1	a^1
a^2	a^1

On to General Setting for Thompson Sampling

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

Today

- Bandits and Probably Approximately Correct
- Bayesian Bandits
- Thompson Sampling
- Bayesian Regret

Framework: Regret and Bayesian Regret

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

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

where $\mathbb{E}_{\tau}$ denotes an expectation with respect to the history of actions taken and rewards observed given an algorithm $\mathcal{A}$.

- Bayesian regret assumes there is a prior over parameters

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

Bounding Regret Using Optimism

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

$$Regret(\mathcal{A}, T; \theta) = \mathbb{E}_\tau \left[\sum_{t=1}^{T} Q(a^*) - Q(a_t) | \theta \right] \leq \mathbb{E}_\tau \left[\sum_{t=1}^{T} U_t(a_t) - Q(a_t) | \theta \right]$$

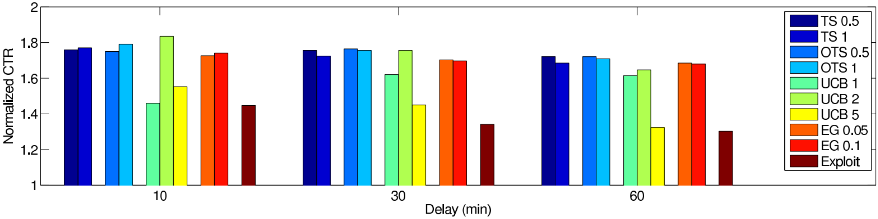
where $\mathbb{E}_\tau$ denotes an expectation with respect to the history of actions taken and rewards observed given an algorithm $\mathcal{A}$ (under event that U_t is an upper bound).

Thompson sampling implements probability matching

- Frequentist bounds for standard* Thompson sampling do not* (last checked) match best bounds for frequentist algorithms
- Empirically Thompson sampling can be effective, especially in contextual multi-armed bandits

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)



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) T
  2. Optimism algorithms would be better than TS here, because they have stronger regret bounds for this setting F
  3. Thompson sampling could cause much worse performance than optimism if the initial prior is very misleading. T
  4. Not sure

Check Your Understanding: Thompson Sampling and Optimism **Solutions**

- 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 for this setting
  3. Thompson sampling could cause much worse performance than optimism if the initial prior is very misleading.
  4. Not sure

  Solution: (1) T (2) F (3) T. Consider prior Beta(100,1) for a Bernoulli arm with parameter 0.1. Then the prior puts large weight on high values of theta for a long time.

Optimal Policy for Bayesian Bandits?

- Thompson Sampling often works well, but is it optimal?
- Given prior, and known horizon, could compute decision policy that would maximize expected rewards given the available horizon
- Computational challenge: naively this would create a decision policy that is a function of the history to the next arm to pull

Gittins Index for Bayesian Bandits

- Thompson Sampling often works well, but is it optimal?
- Given prior, and known horizon, could compute decision policy that would maximize expected rewards given the available horizon
- Computational challenge: naively this would create a decision policy that is a function of the history to the next arm to pull
- **Index policy**: a decision policy that computes a "real-valued index for each arm and plays the arm with the largest index," using statistics only from that arm and the horizon (definition from Lattimore and Svespari 2019 Bandit Algorithms)
- **Gittins index**: optimal policy for maximizing expected discounted reward in a Bayesian multi-armed bandit

Today

- Bandits and Probably Approximately Correct
- Bayesian Bandits
- Thompson Sampling
- Bayesian Regret

What You Should Understand

- Understand how multi-armed bandits relate to MDPs
- Be able to define regret and PAC
- Be able to prove why UCB bandit algorithm has sublinear regret
- Understand (be able to give an example) why e-greedy and greedy and pessimism can result in linear regret
- Be able to implement Thompson Sampling for bernoulli
- Be able to implement UCB bandit algorithm