Lecture 11: Fast Reinforcement Learning

Emma Brunskill

CS234 Reinforcement Learning

Spring 2024

- Slides from or derived from David Silver, Examples new.
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
**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.
Last time: Learning from past data

This time: Data Efficient Reinforcement Learning – Bandits

Next time: Data Efficient Reinforcement Learning
# Computational Efficiency and Sample Efficiency

<table>
<thead>
<tr>
<th>Computational Efficiency</th>
<th>Sample Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atari, mujoco</td>
<td>(data) mobile phones for health improvement</td>
</tr>
<tr>
<td></td>
<td>consumer marketing educational tech</td>
</tr>
<tr>
<td></td>
<td>environmental policies</td>
</tr>
</tbody>
</table>
Evaluation Criteria

- How do we evaluate how "good" an algorithm is?
  - If converges?
  - If converges to optimal policy?
  - How quickly reaches optimal policy?
  - Mistakes make along the way?
- Will introduce different measures to evaluate RL algorithms
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: $\epsilon$-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)
- \(\mathcal{R}^a(r) = \mathbb{P}[r | a]\) 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 \(\sum_{\tau=1}^{t} r_\tau\)
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
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 \( \theta_i \).

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.
Consider deciding how to best treat patients with broken toes

Imagine three 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 $\theta_i$

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

- 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: \( Q(a^1) = \theta_1 = 0.95 \)
  - buddy taping: \( Q(a^2) = \theta_2 = 0.9 \)
  - doing nothing: \( Q(a^3) = \theta_3 = 0.1 \)
Imagine true (unknown) Bernoulli reward parameters for each arm (action) are

- surgery: $Q(a^1) = \theta_1 = 0.95$
- buddy taping: $Q(a^2) = \theta_2 = 0.9$
- doing nothing: $Q(a^3) = \theta_3 = 0.1$

Greedy

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

2. What is the probability of greedy selecting each arm next? Assume ties are split uniformly.
   $$P(a_2) = \frac{1}{2}$$
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 \( a^1 \) (\( r \sim \text{Bernoulli}(0.95) \)), get 0, \( \hat{Q}(a^1) = 0 \)
   - Take action \( a^2 \) (\( r \sim \text{Bernoulli}(0.90) \)), get +1, \( \hat{Q}(a^2) = 1 \)
   - Take action \( a^3 \) (\( r \sim \text{Bernoulli}(0.1) \)), get 0, \( \hat{Q}(a^3) = 0 \)

2. Will the greedy algorithm ever find the best arm in this case?
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**
Setting: Introduction to multi-armed bandits & Approach: greedy methods

**Framework:** Regret

- Approach: $\epsilon$-greedy methods
- Approach: Optimism under uncertainty
- Framework: Bayesian regret
- Approach: Probability matching / Thompson sampling
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** $V^*$
  \[ V^* = Q(a^*) = \max_{a \in A} Q(a) \]

- **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** $V^*$
  \[ V^* = Q(a^*) = \max_{a \in A} Q(a) \]

- **Regret** is the opportunity loss for one step
  \[ l_t = \mathbb{E}[V^* - Q(a_t)] \]

- **Total Regret** is the total opportunity loss
  \[ L_t = \mathbb{E}\left[\sum_{\tau=1}^{t} V^* - Q(a_\tau)\right] \]

- Maximize cumulative reward $\iff$ minimize total regret
Evaluating Regret

- **Count** $N_t(a)$ is number of times action $a$ has been selected at time step $t$.
- **Gap** $\Delta_a$ 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:

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

$$= \sum_{a \in A} \mathbb{E}[N_t(a)](V^* - Q(a))$$

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

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

<table>
<thead>
<tr>
<th>Action</th>
<th>Optimal Action</th>
<th>Observed Reward</th>
<th>Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a^1)</td>
<td>(a^1)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(a^2)</td>
<td>(a^1)</td>
<td>1</td>
<td>.95 = .9 - .9</td>
</tr>
<tr>
<td>(a^3)</td>
<td>(a^1)</td>
<td>0</td>
<td>.8 = 1 - .95</td>
</tr>
<tr>
<td>(a^2)</td>
<td>(a^1)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(a^2)</td>
<td>(a^1)</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Toy Example: Ways to Treat Broken Toes, Optimism, Assessing Regret of Greedy

- True (unknown) Bernoulli reward parameters for each arm (action) are
  - surgery: $Q(a^1) = \theta_1 = 0.95$
  - buddy taping: $Q(a^2) = \theta_2 = 0.9$
  - doing nothing: $Q(a^3) = \theta_3 = 0.1$

- Greedy

<table>
<thead>
<tr>
<th>Action</th>
<th>Optimal Action</th>
<th>Observed Reward</th>
<th>Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a^1$</td>
<td>$a^1$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>$a^3$</td>
<td>$a^1$</td>
<td>0</td>
<td>0.85</td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td>0</td>
<td>0.05</td>
</tr>
</tbody>
</table>

- 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**

<table>
<thead>
<tr>
<th>Action</th>
<th>Optimal Action</th>
<th>Observed Reward</th>
<th>Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a^1$</td>
<td>$a^1$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>$a^3$</td>
<td>$a^1$</td>
<td>0</td>
<td>0.85</td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td>0</td>
<td>0.05</td>
</tr>
</tbody>
</table>

- **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**: $\varepsilon$-greedy methods
- Approach: Optimism under uncertainty
- Framework: Bayesian regret
- Approach: Probability matching / Thompson sampling
The $\epsilon$-greedy algorithm proceeds as follows:

- With probability $1 - \epsilon$ select $a_t = \arg\max_{a \in A} \hat{Q}_t(a)$
- With probability $\epsilon$ select a random action

Always will be making a sub-optimal decision $\epsilon$ fraction of the time

Already used this in prior homeworks
Imagine true (unknown) Bernoulli reward parameters for each arm (action) are

- surgery: $Q(a^1) = \theta_1 = 0.95$
- buddy taping: $Q(a^2) = \theta_2 = 0.9$
- doing nothing: $Q(a^3) = \theta_3 = 0.1$

$\epsilon$-greedy

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

2. Let $\epsilon = 0.1$

3. What is the probability $\epsilon$-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)

<table>
<thead>
<tr>
<th>Action</th>
<th>Optimal Action</th>
<th>Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a^1$</td>
<td>$a^1$</td>
<td></td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td></td>
</tr>
<tr>
<td>$a^3$</td>
<td>$a^1$</td>
<td></td>
</tr>
<tr>
<td>$a^1$</td>
<td>$a^1$</td>
<td></td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td></td>
</tr>
</tbody>
</table>

- Will $\epsilon$-greedy ever select $a^3$ again? If $\epsilon$ 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** $\Delta_a$ 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

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

- A good algorithm ensures small counts for large gap, but gaps are not known
L11N3 Check Your Understanding: $\epsilon$-greedy Bandit Regret

- **Count** $N_t(a)$ is expected number of selections for action $a$
- **Gap** $\Delta_a$ 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 $L_t = \sum_{a \in A} \mathbb{E}[N_t(a)]\Delta_a$

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. $\epsilon = 0.1$ $\epsilon$-greedy can have linear regret
  2. $\epsilon = 0$ $\epsilon$-greedy can have linear regret
  3. Not sure

\[ \exists a \text{ both are true} \]
L11N3 Check Your Understanding: \( \epsilon \)-greedy Bandit Regret

**Answer**

- **Count** \( N_t(a) \) is expected number of selections for action \( a \)
- **Gap** \( \Delta_a \) 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

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

- 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. \( \epsilon = 0.1 \) \( \epsilon \)-greedy can have linear regret
  2. \( \epsilon = 0 \) \( \epsilon \)-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 $a^*$.
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 $\Delta_a$ 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
Setting: Introduction to multi-armed bandits & Approach: greedy methods
Framework: Regret
Approach: $\epsilon$-greedy methods
**Approach: Optimism under uncertainty**
Framework: Bayesian regret
Approach: Probability matching / Thompson sampling
Approach: Optimism in the Face of Uncertainty

- Choose actions that might have a high value
- Why?
- Two outcomes:
  - get high reward
  - learn something
Choose actions 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)

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

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

$$\Pr(\left|\bar{X}_n - \mathbb{E}[X]\right| > u) \leq 2\exp(-2nu^2) = \frac{8}{\sqrt{n}} \exp(-2nu^2) = \frac{8}{\sqrt{n}}$$

$$u^2 = \frac{1}{n} \log \frac{2/\delta}{n}$$

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

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

with probability $\geq 1 - \delta$. 

$\delta$ want to hold with 1-\delta probability.
This leads to the UCB1 algorithm

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

Sample each arm once

\(^1\text{Note: This is a made up example. This is not the actual expected efficacies of the various treatment options for a broken toe} \)
Toy Example: Ways to Treat Broken Toes, Optimism

- True (unknown) parameters for each arm (action) are
  - surgery: $Q(a^1) = \theta_1 = 0.95$
  - buddy taping: $Q(a^2) = \theta_2 = 0.9$
  - doing nothing: $Q(a^3) = \theta_3 = 0.1$

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

---

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

True (unknown) parameters for each arm (action) are
- surgery: \( Q(a^1) = \theta_1 = 0.95 \)
- buddy taping: \( Q(a^2) = \theta_2 = 0.9 \)
- doing nothing: \( Q(a^3) = \theta_3 = 0.1 \)

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

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

\[
\text{UCB}(a^1) = 1 + \sqrt{\frac{2 \log \frac{1}{\delta}}{N_3(a^1)}}
\]
\[
\text{UCB}(a^2) = 1 + \sqrt{\frac{2 \log \frac{1}{\delta}}{N_3(a^2)}}
\]
\[
\text{UCB}(a^3) = 0 + \sqrt{\frac{2 \log \frac{1}{\delta}}{N_3(a^3)}}
\]

\[1\]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 \( a^1 \) (\( r \sim \text{Bernoulli}(0.95) \)), get +1, \( \hat{Q}(a^1) = 1 \)
     - Take action \( a^2 \) (\( r \sim \text{Bernoulli}(0.90) \)), get +1, \( \hat{Q}(a^2) = 1 \)
     - Take action \( a^3 \) (\( r \sim \text{Bernoulli}(0.1) \)), get 0, \( \hat{Q}(a^3) = 0 \)
  2. Set \( t = 3 \), Compute upper confidence bound on each action
     \[
     UCB(a) = \hat{Q}(a) + \sqrt{\frac{2 \log \frac{1}{\delta}}{N_t(a)}}
     \]
  3. \( t = 3 \), Select action \( a_t = \arg \max_a UCB(a) \), \( a = 1 \)
  4. Observe reward 1
  5. Compute upper confidence bound on each action
     \[
     UCB(a_1) = 1 + \sqrt{\frac{2 \log \frac{1}{\delta}}{3}}
     \]
     \[
     UCB(a_2) = 1 + \sqrt{\frac{2 \log \frac{1}{\delta}}{3}}
     \]
     \[
     UCB(a_3) = 0 + \sqrt{\frac{2 \log \frac{1}{\delta}}{1}}
     \]

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

Emma Brunskill (CS234 Reinforcement Learning) Lecture 11: Fast Reinforcement Learning Spring 2024 45 / 53
Toy Example: Ways to Treat Broken Toes, Optimism

- True (unknown) parameters for each arm (action) are:
  - surgery: $Q(a^1) = \theta_1 = 0.95$
  - buddy taping: $Q(a^2) = \theta_2 = 0.9$
  - doing nothing: $Q(a^3) = \theta_3 = 0.1$

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

  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 various treatment options for a broken toe.
## 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)

<table>
<thead>
<tr>
<th>Action</th>
<th>Optimal Action</th>
<th>Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a^1$</td>
<td>$a^1$</td>
<td></td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td></td>
</tr>
<tr>
<td>$a^3$</td>
<td>$a^1$</td>
<td></td>
</tr>
<tr>
<td>$a^1$</td>
<td>$a^1$</td>
<td></td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td></td>
</tr>
</tbody>
</table>
Confidence Level $\delta$

- If there are a fixed number of time steps $T$ for the problem setting, can set $\delta = \frac{\delta}{T}$
  - Union bound: $P(\bigcup E_i) \leq \sum_i P(E_i)$
- Often want to do this in other settings
Any sub-optimal arm \( a \neq a^* \) is pulled by UCB at most

\[
\mathbb{E} N_T(a) \leq C' \frac{\log \frac{1}{\delta}}{\Delta_a^2} + \pi^2 + 1.
\]

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 \( \in [0, 1] \))

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

(1)

Chp 7

for Lattimore, Szepesvari

Ch 7

true empirically

UCB (loose with the 8s)

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

\[
\Delta_a = Q(a^*) - Q(a)
\]

\[
\sum_a \Delta_a \mathbb{E} N_T(a) \leq \sum_a C' \frac{\log T}{\Delta_a} + |A| (\frac{\pi^2}{3} + 1).
\]

\[
\sum_a \Delta_a \mathbb{E} N_T(a) \leq \sum_a C' \frac{\log T}{\Delta_a} + |A| (\frac{\pi^2}{3} + 1).
\]

\[
|Q(a) - \hat{Q}_t(a)| \geq \frac{C \log \frac{1}{\delta}}{N_t(a)}
\]

\[
\frac{C \log \frac{1}{\delta}}{N_t(a)}
\]

under UCB algorithm

\[
\text{UCB}(a) > \text{UCB}(a^*)
\]

\[
\hat{Q}_t(a) + \sqrt{\frac{C \log \frac{1}{\delta}}{N_t(a)}}
\]

\[
\hat{Q}_t(a^*) + \sqrt{\frac{C \log \frac{1}{\delta}}{N_t(a^*)}}
\]

\[
\hat{Q}_t(a) + \sqrt{\frac{C \log \frac{1}{\delta}}{N_t(a)}} > \hat{Q}_t(a^*) + \sqrt{\frac{C \log \frac{1}{\delta}}{N_t(a^*)}} > Q(a^*)
\]

\[
Q(a) + \sqrt{\frac{C \log \frac{1}{\delta}}{N_t(a)}} > Q(a^*)
\]

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

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 \( \in [0, 1] \))

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

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

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

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

\[
\frac{C \log \frac{1}{\delta}}{N_t(a)} \geq \Delta_a^2 \implies \frac{N_t(a)}{N_t(a)} \leq \frac{4C \log \frac{1}{\delta}}{\Delta_a^2} \tag{6}
\]
This leads to the UCB1 algorithm

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

Theorem: The UCB algorithm achieves logarithmic asymptotic total regret

\[ \lim_{t \to \infty} L_t \leq 8 \log t \sum_{a|\Delta_a > 0} \frac{1}{\Delta_a} \]
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: \( \epsilon \)-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