Select all that are true:

1. UCB selects the arm with \( \text{arg max}_a \hat{Q}_t(a) + \sqrt{\frac{1}{N_t(a)} \log(1/\delta)} \)

2. Over an infinite trajectory, UCB will sample all arms an infinite number of times

3. UCB still would learn to pull the optimal arm the most if we instead used \( \text{arg max}_a \hat{Q}_t(a) + \sqrt{\frac{1}{\sqrt{N_t(a)}} \log(t/\delta)} \)

4. UCB uses a bonus on top of average empirical reward. If the bonus was \( \approx 0 \) but small, the resulting algorithm might still suffer linear regret

5. Algorithms that minimize regret also maximize reward

6. Not sure
Class Structure

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

- Fast learning is important when our decisions impact the real world
Over next couple lectures will consider 2 settings, multiple frameworks, and approaches

- **Settings**: Bandits (single decisions), MDPs

- **Frameworks**: evaluation criteria for formally assessing the quality of a RL algorithm. So far seen empirical evaluations, asymptotic convergence, regret

- **Approaches**: Classes of algorithms for achieving particular evaluation criteria in a certain set. So far for exploration seen: greedy, $\epsilon-$greedy, optimism
Recall: Multi-armed Bandit framework

Optimism Under Uncertainty for Bandits

Bayesian Bandits and Bayesian Regret Framework

Probability Matching

Framework: Probably Approximately Correct for Bandits

MDPs
Recall: Multiarmed Bandits

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

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

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

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

1 Recall: Multi-armed Bandit framework

2 Optimism Under Uncertainty for Bandits

3 Bayesian Bandits and Bayesian Regret Framework

4 Probability Matching

5 Framework: Probably Approximately Correct for Bandits

6 MDPs
Approach: Optimism Under Uncertainty

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

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

- Theorem: The UCB algorithm achieves logarithmic asymptotic total regret

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

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

- Pretend already observed one pull of each arm, and saw some optimistic reward
- Average these fake pulls and rewards in when computing average empirical reward
Simple optimism under uncertainty approach
- Pretend already observed one pull of each arm, and saw some optimistic reward
- Average these fake pulls and rewards in when computing average empirical reward

Comparing regret results:
- **Greedy**: Linear total regret
- **Constant $\epsilon$-greedy**: Linear total regret
- **Decaying $\epsilon$-greedy**: Sublinear regret if can use right schedule for decaying $\epsilon$, but that requires knowledge of gaps, which are unknown
- **Optimistic initialization**: Sublinear regret if initialize values sufficiently optimistically, else linear regret
Table of Contents

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

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

Initial prior over $\phi_i$ is $p(\phi_i)$.

Pull arm $i$ and observe reward $r_{i1}$.

Use Bayes rule to update estimate over $\phi_i$: 
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 $\phi_i$

Initial prior over $\phi_i$ is $p(\phi_i)$

Pull arm $i$ and observe reward $r_{i1}$

Use Bayes rule to update estimate over $\phi_i$:

$$p(\phi_i | r_{i1}) = \frac{p(r_{i1} | \phi_i)p(\phi_i)}{p(r_{i1})} = \frac{p(r_{i1} | \phi_i)p(\phi_i)}{\int_{\phi_i} p(r_{i1} | \phi_i)p(\phi_i)d\phi_i}$$
In Bayesian view, we start with a prior over the unknown parameters. Given observations / data about that parameter, update our uncertainty over the unknown parameters using Bayes Rule:

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

In general computing this update may be tricky to do exactly with no additional structure on the form of the prior and data likelihood.
In Bayesian view, we start with a prior over the unknown parameters.

Given observations/data about that parameter, update our uncertainty over the unknown parameters using Bayes Rule:

\[ p(\phi_i|r_{i1}) = \frac{p(r_{i1}|\phi_1)p(\phi_1)}{\int_{\phi_i} p(r_{i1}|\phi_i)p(\phi_i) d\phi_i} \]

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.
Consider a bandit problem where the reward of an arm is a binary outcome \( \{0, 1\} \) sampled from a Bernoulli with parameter \( \theta \)
- E.g. Advertisement click through rate, patient treatment succeeds/fails, ...

The Beta distribution \( \text{Beta}(\alpha, \beta) \) 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.
Consider a bandit problem where the reward of an arm is a binary outcome \( \{0, 1\} \) sampled from a Bernoulli with parameter \( \theta \).

- E.g. Advertisement click through rate, patient treatment succeeds/fails, ...

The Beta distribution \( \text{Beta}(\alpha, \beta) \) is conjugate for the Bernoulli distribution. The probability distribution is:

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

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

- Assume the prior over \( \theta \) is a \( \text{Beta}(\alpha, \beta) \) as above
- Then after observed a reward \( r \in \{0, 1\} \) then updated posterior over \( \theta \) is \( \text{Beta}(r + \alpha, 1 - r + \beta) \)
Bayesian Inference for Decision Making

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

1: Initialize prior over each arm $a$, $p(R_a)$
2: \textbf{loop}
3: For each arm $a$ \textbf{sample} a reward distribution $R_a$ from posterior
4: Compute action-value function $Q(a) = \mathbb{E}[R_a]$
5: \[a_t = \arg \max_{a \in A} Q(a)\]
6: Observe reward $r$
7: Update posterior $p(R_a | r)$ using Bayes law
8: \textbf{end loop}
Toy Example: Ways to Treat Broken Toes, Thompson Sampling

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

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

\[\text{Note: This is a made up example. This is not the actual expected efficacies of the various treatment options for a broken toe}\]
Toy Example: Ways to Treat Broken Toes, Thompson Sampling

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

- True (unknown) Bernoulli parameters for each arm/action
  - Surgery: $\theta_1 = 0.95$
  - Taping: $\theta_2 = 0.9$
  - Nothing: $\theta_3 = 0.1$

- Thompson sampling:
  - Place a prior over each arm’s parameter. Here choose $\theta_i \sim \text{Beta}(1,1)$
  1. Sample a Bernoulli parameter given current prior over each arm
     Beta(1,1), Beta(1,1), Beta(1,1): 0.3, 0.5, 0.6
  2. Select $a_t = \arg \max_{a \in A} Q(a) = \arg \max_{a \in A} \theta(a) = 3$
  3. Observe the patient outcome's outcome: 0
  4. Update the posterior over the $Q(a_t) = Q(a^3)$ value for the arm pulled
    - Beta($c_1$, $c_2$) is the conjugate distribution for Bernoulli
    - If observe 1, $c_1 + 1$ else if observe 0 $c_2 + 1$
  5. New posterior over Q value for arm pulled is:
  6. New posterior $p(Q(a^3)) = p(\theta(a^3)) = \text{Beta}(1, 2)$
Toy Example: Ways to Treat Broken Toes, Thompson Sampling

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

1. Sample a Bernoulli parameter given current prior over each arm
   - Beta(1,1), Beta(1,1), Beta(1,1): 0.3 0.5 0.6
2. Select $a_t = \arg\max_{a\in A} Q(a) = \arg\max_{a\in A} \theta(a) = 3$
3. Observe the patient outcome's outcome: 0
4. New posterior $p(Q(a^3)) = p(\theta(a_3) = \text{Beta}(1,2)$
Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- True (unknown) Bernoulli parameters for each arm/action
- Surgery: $\theta_1 = 0.95$ / Taping: $\theta_2 = 0.9$ / Nothing: $\theta_3 = 0.1$
- Thompson sampling:
  - Place a prior over each arm’s parameter. Here choose $\theta_i \sim \text{Beta}(1,1)$
  - Sample a Bernoulli parameter given current prior over each arm
    Beta(1,1), Beta(1,1), Beta(1,2): 0.7, 0.5, 0.3
Toy Example: Ways to Treat Broken Toes, Thompson Sampling

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

- True (unknown) Bernoulli parameters for each arm/action
- Surgery: $\theta_1 = 0.95$ / Taping: $\theta_2 = 0.9$ / Nothing: $\theta_3 = 0.1$
- Thompson sampling:
- Place a prior over each arm’s parameter. Here choose $\theta_i \sim \text{Beta}(1,1)$
  1. Sample a Bernoulli parameter given current prior over each arm
     Beta(2,1), Beta(1,1), Beta(1,2): 0.71, 0.65, 0.1
  2. Select $a_t = \arg \max_{a \in A} Q(a) = \arg \max_{a \in A} \theta(a) = 1$
  3. Observe the patient outcome’s outcome: 1
  4. New posterior $p(Q(a^1)) = p(\theta(a^1)) = \text{Beta}(3, 1)$
Toy Example: Ways to Treat Broken Toes, Thompson Sampling

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

- Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$

- How does the sequence of arm pulls compare in this example so far?

<table>
<thead>
<tr>
<th>Optimism</th>
<th>TS</th>
<th>Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a^1$</td>
<td>$a^3$</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>$a^1$</td>
<td>$a^1$</td>
<td></td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td></td>
</tr>
</tbody>
</table>
Toy Example: Ways to Treat Broken Toes, Thompson
Sampling vs Optimism

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

<table>
<thead>
<tr>
<th>Optimism</th>
<th>TS</th>
<th>Optimal</th>
<th>Regret Optimism</th>
<th>Regret TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a^1$</td>
<td>$a^3$</td>
<td>$a^1$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td>$a^1$</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>$a^3$</td>
<td>$a^1$</td>
<td>$a^1$</td>
<td>0.85</td>
<td></td>
</tr>
<tr>
<td>$a^1$</td>
<td>$a^1$</td>
<td>$a^1$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td>$a^1$</td>
<td>0.05</td>
<td></td>
</tr>
</tbody>
</table>
Now we will see how Thompson sampling works in general, and what it is doing.
# Table of Contents

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

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

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

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

1: Initialize prior over each arm \( a \), \( p(R_a) \)
2: loop
3: For each arm \( a \) sample a reward distribution \( R_a \) from posterior
4: Compute action-value function \( Q(a) = \mathbb{E}[R_a] \)
5: \( a_t = \arg \max_{a \in A} Q(a) \)
6: Observe reward \( r \)
7: Update posterior \( p(R_a|r) \) using Bayes law
8: end loop
\[ \pi(a \mid h_t) = \mathbb{P}[Q(a) > Q(a'), \forall a' \neq a \mid h_t] \]
\[ = \mathbb{E}_{\mathcal{R} \mid h_t} \left[ 1(a = \arg \max_{a \in \mathcal{A}} Q(a)) \right] \]
How do we evaluate performance in the Bayesian setting?

Frequentist regret assumes a true (unknown) set of parameters

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

Bayesian regret assumes there is a prior over parameters

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

- Regret(UCB,T)

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

- Posterior sampling has the same (ignoring constants) regret bounds as UCB
Thompson sampling implements probability matching

- Thompson sampling (1929) achieves Lai and Robbins lower bound
- Bounds for optimism are tighter than for Thompson sampling
- But empirically Thompson sampling can be extremely effective
Thompson Sampling for News Article Recommendation (Chapelle and Li, 2010)

- Contextual bandit: input context which impacts reward of each arm, context sampled iid each step
- Arms = articles
- Reward = click (+1) on article \( Q(a) = \text{click through rate} \)
- TS did extremely well! Lead to a big resurgence of interest in Thompson sampling.
Check Your Understanding: Thompson Sampling and Optimism

Consider an online news website with thousands of people logging on each second. Frequently a new person will come online before we see whether the last person has clicked (or not). Select all that are true:

1. Thompson sampling would be better than optimism here, because optimism algorithms are deterministic and would select the same action until we get feedback (click or not).
2. Optimism algorithms would be better than TS here, because they have stronger regret bounds.
3. Thompson sampling could cause much worse performance than optimism if the initial prior is very misleading.
4. Not sure
<table>
<thead>
<tr>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Recall: Multi-armed Bandit framework</td>
</tr>
<tr>
<td>2. Optimism Under Uncertainty for Bandits</td>
</tr>
<tr>
<td>3. Bayesian Bandits and Bayesian Regret Framework</td>
</tr>
<tr>
<td>4. Probability Matching</td>
</tr>
<tr>
<td>5. Framework: Probably Approximately Correct for Bandits</td>
</tr>
<tr>
<td>6. MDPs</td>
</tr>
</tbody>
</table>
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 $\epsilon$-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, $\epsilon$, $\delta$, etc)
Most PAC algorithms based on optimism or Thompson sampling
Toy Example: Probably Approximately Correct and Regret Surgery: $\theta_1 = .95$ / Taping: $\theta_2 = .9$ / Nothing: $\theta_3 = .1$

Let $\epsilon = 0.05$.

$O =$ Optimism, $TS = $ Thompson Sampling: $W/in \epsilon = \mathbb{I}(Q(a_t) \geq Q(a^*) - \epsilon)$

<table>
<thead>
<tr>
<th>O</th>
<th>TS</th>
<th>Optimal</th>
<th>O Regret</th>
<th>O W/in $\epsilon$</th>
<th>TS Regret</th>
<th>TS W/in $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a^1$</td>
<td>$a^3$</td>
<td>$a^1$</td>
<td>0</td>
<td>$0.85$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td>$a^1$</td>
<td>0.05</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$a^3$</td>
<td>$a^1$</td>
<td>$a^1$</td>
<td>0.85</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$a^1$</td>
<td>$a^1$</td>
<td>$a^1$</td>
<td>0</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$a^2$</td>
<td>$a^1$</td>
<td>$a^1$</td>
<td>0.05</td>
<td></td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Toy Example: Probably Approximately Correct and Regret

- Surgery: \( \theta_1 = 0.95 \) / Taping: \( \theta_2 = 0.9 \) / Nothing: \( \theta_3 = 0.1 \)
- Let \( \epsilon = 0.05 \).
- \( O = \text{Optimism}, \ TS = \text{Thompson Sampling}: \ W/\text{in } \epsilon = \mathbb{I}(Q(a_t) \geq Q(a^*) - \epsilon) \)

<table>
<thead>
<tr>
<th></th>
<th>TS</th>
<th>Optimal</th>
<th>O Regret</th>
<th>O W/in ( \epsilon )</th>
<th>TS Regret</th>
<th>TS W/in ( \epsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>Y</td>
<td>0.85</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0.05</td>
<td>Y</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.85</td>
<td>N</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Y</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0.05</td>
<td>Y</td>
<td>0</td>
<td>Y</td>
</tr>
</tbody>
</table>
Recall: Multi-armed Bandit framework

Optimism Under Uncertainty for Bandits

Bayesian Bandits and Bayesian Regret Framework

Probability Matching

Framework: Probably Approximately Correct for Bandits

MDPs
Very similar set of frameworks and approaches are relevant for fast learning in reinforcement learning

Frameworks
- Regret
- Bayesian regret
- Probably approximately correct (PAC)

Approaches
- Optimism under uncertainty
- Probability matching / Thompson sampling

Framework: Probably approximately correct
Very similar set of frameworks and approaches are relevant for fast learning in reinforcement learning.

- **Frameworks**
  - Regret
  - Bayesian regret
  - Probably approximately correct (PAC)

- **Approaches**
  - **Optimism under uncertainty**
  - Probability matching / Thompson sampling

- **Framework**: Probably approximately correct
Optimistic Initialization: Model-Free RL

- Initialize action-value function $Q(s,a)$ optimistically (for ex. $\frac{r_{\text{max}}}{1-\gamma}$)
  - where $r_{\text{max}} = \max_a \max_s R(s,a)$
  - Check your understanding: why is that value guaranteed to be optimistic?

- Run favorite model-free RL algorithm
  - Monte-Carlo control
  - Sarsa
  - Q-learning ... 

- Encourages systematic exploration of states and actions
Optimistic Initialization: Model-Free RL

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

1. Be very optimistic until confident that empirical estimates are close to true (dynamics/reward) parameters (Brafman & Tennenholtz JMLR 2002)

2. Be optimistic given the information have
   - Compute confidence sets on dynamics and reward models, or
   - Add reward bonuses that depend on experience / data

We will focus on the last class of approaches
Model-Based Interval Estimation with Exploration Bonus (MBIE-EB)

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

1: Given $\epsilon$, $\delta$, $m$
2: 
3: 
4: 
5: 
6: loop
7: $a_t = \text{arg max}_{a \in A} Q(s_t, a)$
8: Observe reward $r_t$ and state $s_{t+1}$
9: 
10: 
11: 
12: while not converged do
13: $\tilde{Q}(s, a) = \hat{R}(s, a) + \gamma \sum_{s'} \hat{T}(s'|s, a) \text{max}_{a'} \tilde{Q}(s', a') +$
14: end while
15: end loop
Model-Based Interval Estimation with Exploration Bonus (MBIE-EB)

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

1: Given \( \epsilon, \delta, m \)
2: \( \beta = \frac{1}{1-\gamma} \sqrt{0.5 \ln(2|S||A|m/\delta} \)
3: \( n_{sas}(s, a, s') = 0 \quad s \in S, a \in A, s' \in S \)
4: \( rc(s, a) = 0, n_{sa}(s, a) = 0, \tilde{Q}(s, a) = 1/(1 - \gamma) \quad \forall \ s \in S, a \in A \)
5: \( t = 0, s_t = s_{init} \)
6: loop
7: \( a_t = \arg \max_{a \in A} Q(s_t, a) \)
8: Observe reward \( r_t \) and state \( s_{t+1} \)
9: \( n_{sa}(s_t, a_t) = n(s_t, a_t) + 1, n_{sas}(s_t, a_t, s_{t+1}) = n_{sas}(s_t, a_t, s_{t+1}) + 1 \)
10: \( rc(s_t, a_t) = \frac{rc(s_t, a_t) n_{sa}(s_t, a_t) + r_t}{(n_{sa}(s_t, a_t) + 1)} \)
11: \( \hat{R}(s, a) = \frac{rc(s_t, a_t)}{n(s_t, a_t)} \) and \( \hat{T}(s' | s, a) = \frac{n_{sas}(s_t, a_t, s')}{n_{sa}(s_t, a_t)} \quad \forall s' \in S \)
12: while not converged do
13: \( \tilde{Q}(s, a) = \hat{R}(s, a) + \gamma \sum_{s'} \hat{T}(s' | s, a) \max_{a'} \tilde{Q}(s', a') + \frac{\beta}{\sqrt{n_{sa}(s, a)}} \quad \forall \ s \in S, a \in A \)
14: end while
15: end loop
For a given $\epsilon$ and $\delta$, a RL algorithm $\mathcal{A}$ is PAC if on all but $N$ steps, the action selected by algorithm $\mathcal{A}$ on time step $t$, $a_t$, is $\epsilon$-close to the optimal action, where $N$ is a polynomial function of $(|S|, |A|, \gamma, \epsilon, \delta)$.

Is this true for all algorithms?
Theorem 2. Suppose that $\epsilon$ and $\delta$ are two real numbers between 0 and 1 and $M = \langle S, A, T, R, \gamma \rangle$ is any MDP. There exists an input $m = m\left(\frac{1}{\epsilon}, \frac{1}{\delta}\right)$, satisfying $m\left(\frac{1}{\epsilon}, \frac{1}{\delta}\right) = O\left(\frac{|S|}{\epsilon^2(1-\gamma)^4} + \frac{1}{\epsilon^2(1-\gamma)^4} \ln \frac{|S||A|}{\epsilon(1-\gamma)^{\delta}}\right)$, and $\beta = (1/(1-\gamma))\sqrt{\ln(2|S||A|m/\delta)}/2$ such that if MBIE-EB is executed on MDP $M$, then the following holds. Let $A_t$ denote MBIE-EB’s policy at time $t$ and $s_t$ denote the state at time $t$. With probability at least $1 - \delta$, $V_M^{A_t}(s_t) \geq V_M^*(s_t) - \epsilon$ is true for all but $O\left(\frac{|S||A|}{\epsilon^3(1-\gamma)^6} (|S| + \ln \frac{|S||A|}{\epsilon(1-\gamma)^{\delta}}) \ln \frac{1}{\delta} \ln \frac{1}{\epsilon(1-\gamma)}\right)$ timesteps $t$. 
A Sufficient Set of Conditions to Make a RL Algorithm PAC

MBIE-EB Empirically: 6 Arms Results

![Bar chart showing cumulative reward for different algorithms: MBIE-EB, MBIE, R-Max, and E-3. The x-axis represents the algorithms, and the y-axis represents the cumulative reward.]
Summary so Far: Settings, Frameworks & Approaches

- Over 3 lectures will consider 2 settings, multiple frameworks, and approaches
- Settings: Bandits (single decisions), MDPs
- Frameworks: evaluation criteria for formally assessing the quality of a RL algorithm. So far seen empirical evaluations, asymptotic convergence, regret, probably approximately correct (PAC)
- Approaches: Classes of algorithms for achieving particular evaluation criteria in a certain set. So far for exploration seen: greedy, $\epsilon$-greedy, optimism, Thompson sampling