

# Stanford·CS234 | Reinforcement Learning (2019)

## CS234 (2019)· 课程资料包 @ShowMeAI



视频

中英双语字幕



课件

一键打包下载



笔记

官方笔记翻译



代码

作业项目解析



视频·B站【扫码或点击链接】

<https://www.bilibili.com/video/BV1H64y1x7GH>



课件 & 代码·博客【扫码或点击链接】

<http://blog.showmeai.tech/cs234>

斯坦福

reinforcement learning  
马尔可夫决策过程

DQN

强化学习

值函数方法

policy  
gradient

Q-learning

梯度策略

Model free  
蒙特卡洛搜索树

Awesome AI Courses Notes Cheatsheets 是 [ShowMeAI](#) 资料库的分支系列，覆盖最具知名度的 **TOP20+** 门 AI 课程，旨在为读者和学习者提供一整套高品质中文学习笔记和速查表。

点击课程名称，跳转至课程资料包页面，**一键下载**课程全部资料！

机器学习	深度学习	自然语言处理	计算机视觉
Stanford · CS229	Stanford · CS230	Stanford · CS224n	Stanford · CS231n
# Awesome AI Courses Notes Cheatsheets · 持续更新中			
知识图谱	图机器学习	深度强化学习	自动驾驶
Stanford · CS520	Stanford · CS224W	UCBerkeley · CS285	MIT · 6.S094



微信公众号

资料下载方式 2: 扫码点击底部菜单栏

称为 AI 内容创作者? 回复【添砖加瓦】

# Lecture 11: Fast Reinforcement Learning <sup>1</sup>

Emma Brunskill

CS234 Reinforcement Learning

Winter 2019

---

<sup>1</sup>With many slides from or derived from David Silver, Examples new

# Class Structure

- Last time: Midterm
- **This time: Fast Learning**
- Next time: Fast Learning

# Up Till Now

- Discussed optimization, generalization, delayed consequences

# Teach Computers to Help Us



education  
healthcare  
consumer  
marketing



# Computational Efficiency and Sample Efficiency

Q learning

Computational Efficiency

driving car at 60mph  
simulators

Sample Efficiency

experience costly/hard to  
gather

- education - students
- patients
- customers

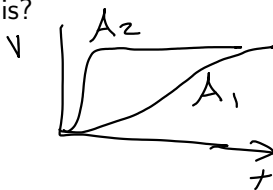
# Algorithms Seen So Far

- How many steps did it take for DQN to learn a good policy for pong?

*5 millions*

# Evaluation Criteria

- How do we evaluate how "good" an algorithm is?
- If converges?  $t \rightarrow \infty$
- If converges to optimal policy?  $t \rightarrow \infty$
- How quickly reaches optimal policy?
- Mistakes make along the way?
- Will introduce different measures to evaluate RL algorithms





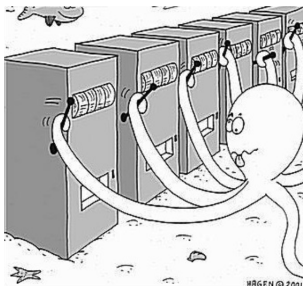
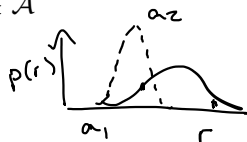
# Settings, Frameworks & Approaches

- Over next couple lectures will consider 2 settings, multiple frameworks, and approaches
- Settings: Bandits (single decisions), MDPs *tabular*
- 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

- Setting: Introduction to multi-armed bandits
- Framework: Regret
- 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  $\underline{m}$  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

- **Action-value** is the mean reward for action  $a$

$$\underline{Q(a)} = \mathbb{E}[r \mid a]$$

- **Optimal value**  $V^*$

$$V^* = Q(a^*) = \max_{a \in \mathcal{A}} Q(a)$$

- **Regret** is the opportunity loss for one step

*arm selection*  
 $\downarrow$   
 $Q(a^*) - Q(a_t)$

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

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

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

*unknown* (pointing to  $V^*$ )  
*unknown* (pointing to  $Q(a_\tau)$ )  
 $r_t \sim P(r|a_t)$

- Maximize cumulative reward  $\iff$  minimize total regret

# Evaluating Regret

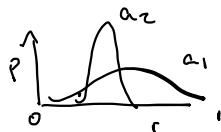
- **Count**  $N_t(a)$  is expected number of selections for action  $a$
- **Gap**  $\Delta_a$  is the difference in value between action  $a$  and optimal action  $a^*$ ,  $\Delta_i = \underbrace{V^* - Q(a_i)}$   $\Delta_a \leftarrow 0$
- Regret is a function of gaps and counts

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

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

# Greedy Algorithm

- We consider algorithms that estimate  $\hat{Q}_t(a) \approx Q(a)$
- 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 action with highest value

.2     $a_1$   
.5     $a_2$

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

- Greedy can lock onto suboptimal action, forever

# $\epsilon$ -Greedy Algorithm

- The  $\epsilon$ -**greedy** algorithm proceeds as follows:
  - With probability  $1 - \epsilon$  select  $a_t = \arg \max_{a \in \mathcal{A}} \hat{Q}_t(a)$
  - 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

# Toy Example: Ways to Treat Broken Toes<sup>1</sup>

- 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

---

<sup>1</sup>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<sup>1</sup>

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

---

<sup>1</sup>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<sup>1</sup>



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

---

<sup>1</sup>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, Greedy<sup>1</sup>

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

$$p(a_1) = p(a_2) = 1/2$$

<sup>1</sup>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 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:  $Q(a^3) = \theta_3 = .1$

← optimal

$$\frac{Q(a^*) - Q(a_t)}{\sqrt{t}}$$

Greedy

r	Action	Optimal Action	Regret
1	$a^1$	$a^1$	0
2	$a^2$	$a^1$	$.95 - .9 = .05$
3	$a^3$	$a^1$	$.95 - .1 = .85$
4	$a^1$	$a^1$	0
5	$a^2$	$a^1$	.05

init

- Will greedy ever select  $a^3$  again? If yes, why? If not, is this a problem?

no

# Toy Example: Ways to Treat Broken Toes, $\epsilon$ -Greedy<sup>1</sup>

- 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:  $Q(a^3) = \theta_3 = .1$
- $\epsilon$ -greedy
  - 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,  $\hat{Q}(a^3) = 0$
  - Let  $\epsilon = 0.1$
  - What is the probability  $\epsilon$ -greedy will pull each arm next? Assume ties are split uniformly.  
$$\frac{\epsilon}{3} : a_1 \text{ or } a_2 \text{ or } a_3$$
$$\frac{1-\epsilon}{2} : a_1 \text{ or } a_2$$

<sup>1</sup>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 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  $\epsilon$ -greedy ever select  $a^3$  again? If  $\epsilon$  is fixed, how many times will each arm be selected? *yes*

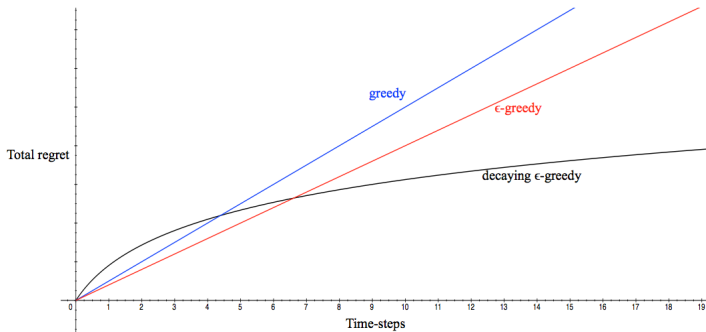
# Check Your Understanding

- **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

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

- A good algorithm ensures small counts for large gap, but gaps are not known
- Check your understanding: Does fixed  $\epsilon = 0.1$  greedy have large regret ?

# "Good": Sublinear or below regret



- **Explore forever:** have linear total regret
- **Explore never:** have linear total regret
- Is it possible to achieve sublinear regret?

$T$



# 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  $a^*$

$a_1$	$a_2$
$E[r]$	$E[r]$
1	.001
.53	.525

# 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  $\Delta_a$  and the similarity in distributions  $D_{KL}(\mathcal{R}^a \parallel \mathcal{R}^{a^*})$  *KL div*
- Theorem (Lai and Robbins): Asymptotic total regret is at least logarithmic in number of steps

$$\lim_{t \rightarrow \infty} L_t \geq \underbrace{\log t}_{\text{regret}} \sum_{a | \Delta_a > 0} \frac{\Delta_a}{D_{KL}(\mathcal{R}^a \parallel \mathcal{R}^{a^*})}$$

- Promising in that lower bound is sublinear

# Approach: Optimism in the Face of Uncertainty

Kaelbling  
1993

- Choose actions that that might have a high value
- Why?
- Two outcomes:  $a_1$

1)  $a_1$  has high reward

2)  ~~$a_1$  has~~  $r(a_1)$  with low reward information

# Upper Confidence Bounds

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

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

UCB    init phase: pull each arm once, compute  $U_t(a_t)$   
for  $t = 1 \dots$   
     $a_t = \arg \max_a U_t(a)$   
     $r \sim$  reward distrib  $a_t$   
    Update  $U_t(a_t)$  & all other arms

# Hoeffding's Inequality

if confidence bounds hold

$$U_t(a_t) = \hat{Q}(a_t) + \sqrt{\frac{1}{2n(a_t)} \log(1^2/\delta)} \\ > Q(a_t)$$

- Theorem (Hoeffding's Inequality): Let  $X_1, \dots, X_n$  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

$$\mathbb{P} \left[ \underbrace{\mathbb{E}[X]}_{\text{true mean}} > \underbrace{\bar{X}_n}_{\text{empirical mean}} + u \right] \leq \exp(-2nu^2) = \delta/t^2$$

$\nwarrow$  constant       $\nearrow$  # samples

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

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

$$\bar{X}_n + u \geq \mathbb{E}[X] \text{ w. prob } \geq 1 - \delta/t^2$$

$$U_t(a_t) = \hat{Q}(a_t) + \sqrt{\frac{1}{2n(a_t)} \cdot \log(1^2/\delta)}$$

# High Probability Regret Bound for UCB Multi-armed Bandit

# High Probability Regret Bound for UCB Multi-armed Bandit

# High Probability Regret Bound for UCB Multi-armed Bandit



# High Probability Regret Bound for UCB Multi-armed Bandit

$$\text{Regret}(\text{UCB}, T) = \sum_{t=1}^T Q(a^*) - Q(a_t)$$

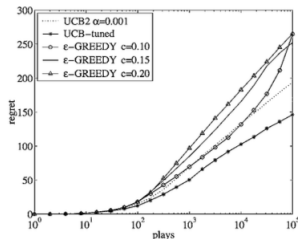
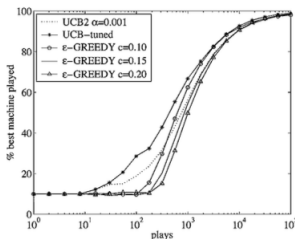
# UCB Bandit Regret

- This leads to the UCB1 algorithm

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

- Theorem: The UCB algorithm achieves logarithmic asymptotic total regret

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



# Toy Example: Ways to Treat Broken Toes, Thompson Sampling<sup>1</sup>

- 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

---

<sup>1</sup>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<sup>1</sup>

- 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,  $\hat{Q}(a^3) = 0$

---

<sup>1</sup>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<sup>1</sup>

- 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

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

---

<sup>1</sup>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<sup>1</sup>

- 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

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

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

# Toy Example: Ways to Treat Broken Toes, Optimism<sup>1</sup>

- 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

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

- 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

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

- True (unknown) parameters for each arm (action) are
  - surgery:  $Q(a^1) = \theta_1 = .95$
  - buddy taping:  $Q(a^2) = \theta_2 = .9$
  - doing nothing:  $Q(a^3) = \theta_3 = .1$
- UCB1 (Auer, Cesa-Bianchi, Fischer 2002)

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



# Check Your Understanding

- An alternative would be to always select the arm with the highest lower bound
- Why can this yield linear regret?
- Consider a two arm case for simplicity

# Bayesian Bandits

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

# Regret and Bayesian Regret

- Frequentist regret assumes a true (unknown) set of parameters

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

- Bayesian regret assumes there is a prior over parameters

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

- \*Note: Bayes regret and regret can be related using Markov inequality

# Bayesian UCB Example: Independent Gaussians

- Assume reward distribution is Gaussian,  $\mathcal{R}_a(r) = \mathcal{N}(r; \mu_a, \sigma_a^2)$
- Compute Gaussian posterior over  $\mu_a$  and  $\sigma_a^2$  (by Bayes law)

$$p[\mu_a, \sigma_a^2 \mid h_t] \propto p[\mu_a, \sigma_a^2] \prod_{t \mid a_t = a} \mathcal{N}(r_t; \mu_a, \sigma_a^2)$$

- Pick action that maximizes standard deviation of  $Q(a)$

$$a_t = \arg \max_{a \in \mathcal{A}} \mu_a + c \frac{c \sigma_a}{\sqrt{N(a)}}$$

# Probability Matching

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

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

- Probability matching is optimistic in the face of uncertainty
  - Uncertain actions have higher probability of being max
- Can be difficult to compute analytically from posterior

# Thompson sampling implements probability matching

- Thompson sampling:

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

# Thompson sampling implements probability matching

- Thompson sampling:

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

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

# 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<sup>2</sup>

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

---

<sup>2</sup>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:  $\theta_1 = .95$  / 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  
 $\text{Beta}(1,1), \text{Beta}(1,1), \text{Beta}(1,1): 0.3 \ 0.5 \ 0.6$
  - 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  $Q(a_t) = Q(a^1)$  value for the arm pulled
    - $\text{Beta}(c_1, c_2)$  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  
 $\text{Beta}(1,1)$ ,  $\text{Beta}(1,1)$ ,  $\text{Beta}(1,1)$ : 0.3 0.5 0.6
  - 2 Select  $a_t = \arg \max_{a \in A} Q(a) = \arg \max_{a \in A} \theta(a) = 1$
  - 3 Observe the patient outcome's outcome: 0
  - 4 New posterior  $p(Q(a^1)) = p(\theta(a_1) = \text{Beta}(1, 2)$

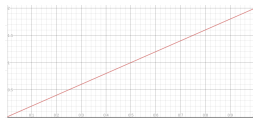


# Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- True (unknown) Bernoulli parameters for each arm/action
- Surgery:  $\theta_1 = .95$  / Taping:  $\theta_2 = .9$  / Nothing:  $\theta_3 = .1$
- Thompson sampling:
- Place a prior over each arm's parameter. Here choose  $\theta_i \sim \text{Beta}(1,1)$ 
  - ① Sample a Bernoulli parameter given current prior over each arm  
Beta(1,1), Beta(1,1), Beta(1,2): 0.7, 0.5, 0.3

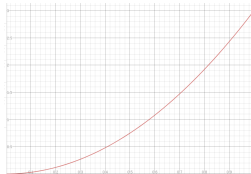
# Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- True (unknown) Bernoulli parameters for each arm/action
- Surgery:  $\theta_1 = .95$  / Taping:  $\theta_2 = .9$  / Nothing:  $\theta_3 = .1$
- Thompson sampling:
- Place a prior over each arm's parameter. Here choose  $\theta_i \sim \text{Beta}(1,1)$ 
  - 1 Sample a Bernoulli parameter given current prior over each arm  
Beta(1,1), Beta(1,1), Beta(1,2): 0.7, 0.5, 0.3
  - 2 Select  $a_t = \arg \max_{a \in A} Q(a) = \arg \max_{a \in A} \theta(a) = 1$
  - 3 Observe the patient outcome's outcome: 1
  - 4 New posterior  $p(Q(a^1)) = p(\theta(a_1)) = \text{Beta}(2, 1)$



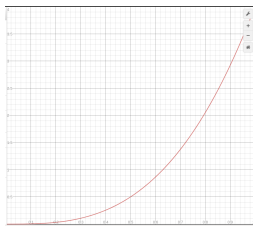
# Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- True (unknown) Bernoulli parameters for each arm/action
- Surgery:  $\theta_1 = .95$  / Taping:  $\theta_2 = .9$  / Nothing:  $\theta_3 = .1$
- Thompson sampling:
- Place a prior over each arm's parameter. Here choose  $\theta_i \sim \text{Beta}(1,1)$ 
  - 1 Sample a Bernoulli parameter given current prior over each arm  
 $\text{Beta}(2,1)$ ,  $\text{Beta}(1,1)$ ,  $\text{Beta}(1,2)$ : 0.71, 0.65, 0.1
  - 2 Select  $a_t = \arg \max_{a \in A} Q(a) = \arg \max_{a \in A} \theta(a) = 1$
  - 3 Observe the patient outcome's outcome: 1
  - 4 New posterior  $p(Q(a^1)) = p(\theta(a_1) = \text{Beta}(3, 1)$



# Toy Example: Ways to Treat Broken Toes, Thompson Sampling

- True (unknown) Bernoulli parameters for each arm/action
- Surgery:  $\theta_1 = .95$  / Taping:  $\theta_2 = .9$  / Nothing:  $\theta_3 = .1$
- Thompson sampling:
- Place a prior over each arm's parameter. Here choose  $\theta_i \sim \text{Beta}(1,1)$ 
  - 1 Sample a Bernoulli parameter given current prior over each arm  
Beta(2,1), Beta(1,1), Beta(1,2): 0.75, 0.45, 0.4
  - 2 Select  $a_t = \arg \max_{a \in A} Q(a) = \arg \max_{a \in A} \theta(a) = 1$
  - 3 Observe the patient outcome's outcome: 1
  - 4 New posterior  $p(Q(a^1)) = p(\theta(a_1)) = \text{Beta}(4, 1)$





# Toy Example: Ways to Treat Broken Toes, Thompson Sampling vs Optimism

- Surgery:  $\theta_1 = .95$  / Taping:  $\theta_2 = .9$  / Nothing:  $\theta_3 = .1$
- How does the sequence of arm pulls compare in this example so far?

Optimism	TS	Optimal	Regret Optimism	Regret TS
$a^1$	$a^3$			
$a^2$	$a^1$			
$a^3$	$a^1$			
$a^1$	$a^1$			
$a^2$	$a^1$			

# Toy Example: Ways to Treat Broken Toes, Thompson Sampling vs Optimism

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

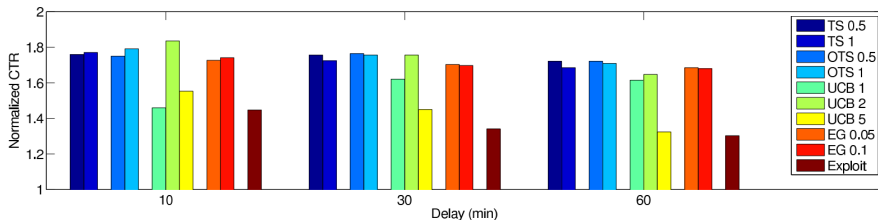
Optimism	TS	Optimal	Regret Optimism	Regret TS
$a^1$	$a^3$	$a^1$	0	0
$a^2$	$a^1$	$a^1$	0.05	
$a^3$	$a^1$	$a^1$	0.85	
$a^1$	$a^1$	$a^1$	0	
$a^2$	$a^1$	$a^1$	0.05	

# Thompson sampling implements probability matching

- Thompson sampling(1929) achieves Lai and Robbins lower bound
- Bounds for optimism are tighter than for Thompson sampling
- But empirically Thompson sampling can be extremely effective

# Thompson Sampling for News Article Recommendation (Chapelle and Li, 2010)

- Contextual bandit: input context which impacts reward of each arm, context sampled iid each step
- Arms = articles
- Reward = click (+1) on article ( $Q(a)$ =click through rate)



# Bayesian Regret Bounds for Thompson Sampling

- $\text{Regret}(\text{UCB}, T)$

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

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

# Optimistic Initialization

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

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

- 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?

# Greedy Bandit Algorithms and Optimistic Initialization

- **Greedy**: Linear total regret
- **Constant  $\epsilon$ -greedy**: Linear total regret
- **Decaying  $\epsilon$ -greedy**: Sublinear regret but schedule for decaying  $\epsilon$  requires knowledge of gaps, which are unknown
- **Optimistic initialization**: Sublinear regret if initialize values sufficiently optimistically, else linear regret
- Check your understanding: why does fixed  $\epsilon$ -greedy have linear regret? (Do a proof sketch)

# Consider Montezuma's revenge

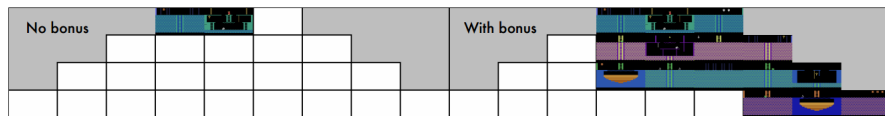


Figure 3: “Known world” of a DQN agent trained for 50 million frames with (**right**) and without (**left**) count-based exploration bonuses, in MONTEZUMA’S REVENGE.

- EB: move this to generalization and efficiency later on
- Bellemare et al. "Unifying Count-Based Exploration and Intrinsic Motivation"
- Enormously better than standard DQN with  $\epsilon$ -greedy approach
- Uses principle of optimism under uncertainty which we will see today



# Calculating UCB

- Pick a probability  $p$  that true value exceeds UCB
- Now solve for  $U_t(a)$

$$\exp(-2N_t(a)U_t(a)^2) = p$$

$$U_t(a) = \sqrt{\frac{-\log p}{2N_t(a)}}$$

- Reduce  $p$  as we observe more rewards, e.g.  $p = t^{-4}$
- Ensures we select optimal action as  $t \rightarrow \infty$

$$U_t(a) = \sqrt{\frac{2 \log t}{N_t(a)}}$$

- This leads to the UCB1 algorithm

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

# Stanford·CS234 | Reinforcement Learning (2019)

## CS234 (2019)· 课程资料包 @ShowMeAI



视频

中英双语字幕



课件

一键打包下载



笔记

官方笔记翻译



代码

作业项目解析



视频·B站【扫码或点击链接】

<https://www.bilibili.com/video/BV1H64y1x7GH>



课件 & 代码·博客【扫码或点击链接】

<http://blog.showmeai.tech/cs234>

斯坦福

reinforcement learning  
马尔可夫决策过程

DQN

强化学习

值函数方法

policy  
gradient

Q-learning

梯度策略

Model free

蒙特卡洛搜索树

Awesome AI Courses Notes Cheatsheets 是 [ShowMeAI](#) 资料库的分支系列，覆盖最具知名度的 **TOP20+** 门 AI 课程，旨在为读者和学习者提供一整套高品质中文学习笔记和速查表。

点击课程名称，跳转至课程资料包页面，**一键下载**课程全部资料！

机器学习	深度学习	自然语言处理	计算机视觉
Stanford · CS229	Stanford · CS230	Stanford · CS224n	Stanford · CS231n
# Awesome AI Courses Notes Cheatsheets · 持续更新中			
知识图谱	图机器学习	深度强化学习	自动驾驶
Stanford · CS520	Stanford · CS224W	UCBerkeley · CS285	MIT · 6.S094



微信公众号

资料下载方式 2: 扫码点击底部菜单栏

称为 AI 内容创作者? 回复【添砖加瓦】