CS292F StatRL Lecture 6
RL Algorithms + Bandits

Instructor: Yu-Xiang Wang

Spring 2021

UC Santa Barbara

Recap: Lecture 5

- Model-free RL that combines MC and DP
  - Temporal Difference methods and their variants
  - Function approximation ← MDP is large $|S|$ is large
  - TD methods (TD prediction / policy evaluation) as one that minimizes the square Bellman error a semi-gradient update.
  - LSTD and its fast computation

- Policy gradients methods

Recap: TD-method in a nutshell --- attempting to simulate Bellman equations with roll-out data

Bellman eqn. $V^\pi = r^\pi + \gamma P^\pi V^\pi \iff V^\pi(s) = \mathbb{E}_{a\sim\pi(a|s)}\left[r(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)}\left[V^\pi(s')\right]\right]$

TD-prediction $\hat{V}^\pi_{t+1}(s) = \hat{V}^\pi_t(s) + \alpha\left[R_{t+1} + \gamma \hat{V}^\pi_t(S_{t+1}) - \hat{V}^\pi(S_t)\right]$

G_t if this is $V^\pi(S_{t+1})$

$\mathbb{E}[G_t] = r^\pi(S_t) + \gamma \mathbb{E}_{S_{t+1}} V^\pi(S_{t+1})$

Setting $\alpha = 1$ $\quad \alpha \ll 1$

"Bootstrap" use $\hat{V}^\pi_t(S_{t+1})$. not unbiased, lower variance

"MC" $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$

unbiased but large variance

TD-methods: what I did not cover

- n-step TD
- TD(λ) / Eligible traces
- Off-policy methods and the "deadly triad"
- Semi-gradient vs actual SGD?
- Projected Bellman error and Gradient-TD

1. off policy
2. bootstrap
3. function approx.

(A large body of associated work on these.
Read Sutton and Barto Ch 7, CH 11-12 and the references therein.)

This lecture

- Continue with policy gradient methods
- Start exploration
  - Multi-armed bandits

Recap: Policy class and policy gradient methods

- Policy $\pi \in \Pi$
- Parametric policy class:

$$\Pi = \{\pi_\theta | \theta \in \mathbb{R}^d\}$$

- Goal: optimize the value
- Policy gradient methods
  - aim at learning the policy parameter by SGD.

How to estimate the gradient?

$R(\tau) = R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots$

- Policy gradient theorem:

Claim 1: $\nabla V^{\pi_\theta}(u) = \mathbb{E}_{\tau \sim P_u^{\pi_\theta}}\left[R(\tau) \sum_{t=0}^{\infty} \nabla \log \pi_\theta(A_t|S_t)\right]$

(τ: trajectory)

"Reinforce" Alg.

① Roll out a trajectory τ with π_θ

② $R(\tau) \sum_{t=0}^{\infty} \nabla \log \pi_\theta(A_t|S_t)$

Claim 2: $\nabla V^{\pi_\theta}(u) = \mathbb{E}_{\tau \sim P_u^{\pi_\theta}}\left[\sum_{t=1}^{\infty} \gamma^t Q^{\pi_\theta}(S_t, A_t) \nabla \log \pi_\theta(A_t|S_t)\right]$

$= \sum_s V_u^{\pi_\theta}(s) \sum_a \pi_\theta(a|s) \cdot Q^{\pi_\theta}(s,a) \nabla \log \pi_\theta(a|s)$

$= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_u^{\pi_\theta}} \mathbb{E}_{A \sim \pi_\theta(\cdot|s)}\left[Q^{\pi_\theta}(s,A) \nabla \log \pi_\theta(A|s)\right]$

Claim 3: $\nabla V^{\pi_\theta}(u) = \sum_s V_u^{\pi_\theta}(s) \sum_a \pi_\theta(a|s) \cdot \left(Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)\right) \nabla \log \pi_\theta(a|s)$

$A^{\pi_\theta}(s,A)$

Proof of Policy Gradient Theorem

Claim 1: $\nabla V^{\pi_\theta}(\mu) = \nabla \sum_\tau R(\tau) \cdot P_\mu^{\pi_\theta}(\tau)$

$= \sum_\tau R(\tau) \cdot \nabla P_\mu^{\pi_\theta}(\tau)$ "score function trick"

$= \sum_\tau R(\tau) \cdot P_\mu^{\pi_\theta}(\tau) \cdot \nabla \log P_\mu^{\pi_\theta}(\tau)$

$= \sum_\tau R(\tau) \cdot P_\mu^{\pi_\theta}(\tau) \nabla_\theta \log \left[\mu(s_0) \pi_\theta(A_0|S_0) \cdot P(S_1, R_1|A_0, S_0) \right.$

$\left. \pi_\theta(A_1|S_1) \cdot P(S_2, R_2|A_1, S_1) \right.$

$\left. \vdots \qquad \vdots \right]$

$= \sum_\tau R(\tau) \cdot P_\mu^{\pi_\theta}(\tau) \sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(A_t|S_t)$

* this works for non-stationary π_θ
* ~~X~~ works for POMDP
* ~~X~~ works for case when MDP model itself is wrong

□

Proof of Policy Gradient Theorem

Claim 2. $\forall s \in S$: $\nabla^{\theta} V^{\pi_\theta}(s_0) = \nabla \sum_{a_0} \pi_\theta(a_0|s_0) \cdot Q^{\pi_\theta}(s_0, a_0)$

$(x \cdot y)' = x'y + xy'$

$= \sum_{a_0} \nabla \pi_\theta(a_0|s_0)\, Q^{\pi_\theta}(s_0, a_0) + \sum_{a_0} \pi_\theta(a_0|s_0) \cdot \nabla Q^{\pi_\theta}(s_0, a_0)$

$= \sum_{a_0} \pi_\theta(a_0|s_0) \nabla \log \pi_\theta(a_0|s_0) \cdot Q^{\pi_\theta}(s_0, a_0)$

$+ \sum_{a_0} \pi_\theta(a_0|s_0) \nabla \left(r(s_0, a_0) + \gamma \sum_{s_1} P(s_1|s_0, a_0)\, V^{\pi_\theta}(s_1) \right)$

$= \sum_{a_0} \pi_\theta(a_0|s_0) \left[\nabla \log \pi_\theta(a_0|s_0) \right] Q^{\pi}(s, a_0)$

$+ \gamma E\left[\nabla V^{\pi_\theta}(s_1) \right]$

$\gamma \sum_{s_1} P(s_1|s_0, a_0) \nabla V^{\pi_\theta}(s_1)$

$= \gamma E\left[\nabla V^{\pi_\theta}(s_1) \,|\, s_0, A_0 \right]$

$\nabla V^{\pi_\theta}(\mu) = \sum_{s_0} \mu(s_0) \nabla V^{\pi_\theta}(s_0)$

$= E\left[\nabla \log \pi_\theta(A_0|s_0) \cdot Q^{\pi_\theta}(s_0, A_0) \right] + \gamma E_{s_0 \sim \mu} E_{s_1 \sim P_1^{\pi_\theta}}\left[\nabla V^{\pi}(s_1) \right]$

$\gamma E\left[\nabla \log \pi_\theta(A_1|s_1)\, Q^{\pi_\theta}(s_1) \right] + \gamma^2 \sum_{s_2} \left[\nabla V^{\pi}(s_2) \right]$

REINFORCE Algorithm

REINFORCE, A Monte-Carlo Policy-Gradient Method (episodic)

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$
Repeat forever:
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
  For each step of the episode $t = 0, \ldots, T-1$:
    $G \leftarrow$ return from step t
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla_{\boldsymbol{\theta}} \ln \pi(A_t|S_t, \boldsymbol{\theta})$

"Claim 1 of the Thm"

Variance of the gradient estimate and convergence

- This is a non-convex optimization problem
- Convergence to a stationary point.
  - Choice of learning rate:
  - Then (Lemma 9.8 AJKS)

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla V^{\pi_{\theta_t}}(\mu)\|^2\right] \leq \frac{2\beta\left(V^* - V^{\pi_{\theta_0}}\right)}{T} + \sqrt{\frac{2G}{T}}$$

- Does it converge to the global optimal solution?
  - Yes, sometimes (under additional assumptions, see Ch 10 AJKS).

REINFORCE with a baseline

- REINFORCE with a given (arbitrary) baseline

$$\theta^{t} = \theta + \alpha \gamma^{t} \cdot G \cdot \nabla_{\theta} \log \pi_{\theta}(A_t|S_t,\theta)$$

REINFORCE with baseline

$$\theta^{t} = \theta + \alpha \gamma^{t} [G - f(S_t)] \cdot \nabla_{\theta} \log \pi_{\theta}(A_t|S_t)$$

- Choose it to approximate the value function.
- Why does it work?

It suffices $\mathbb{E}\left[f(S_t) \cdot \nabla_{\theta} \log \pi_{\theta}(A_t|S_t)\right] = 0$

$$\mathbb{E}\left[\mathbb{E}_{\pi}\left[f(S_t) \cdot \nabla_{\theta} \log \pi_{\theta}(A_t|S_t) \mid S_t\right]\right]$$

$$= \mathbb{E}\left[\sum_{a} \pi_{\theta}(a|S_t) \cdot f(S_t) \nabla_{\theta} \log \pi_{\theta}(a|S_t) \mid S_t\right] = \mathbb{E}\left[\sum_{a} \pi_{\theta}(a|S_t) \frac{\nabla_{\theta} \pi(a|S_t)}{\pi_{\theta}(a|S_t)} f(S_t)\right]$$

$f(s): \mathcal{S} \to \mathbb{R}$

$f = V^{\pi_\theta}$

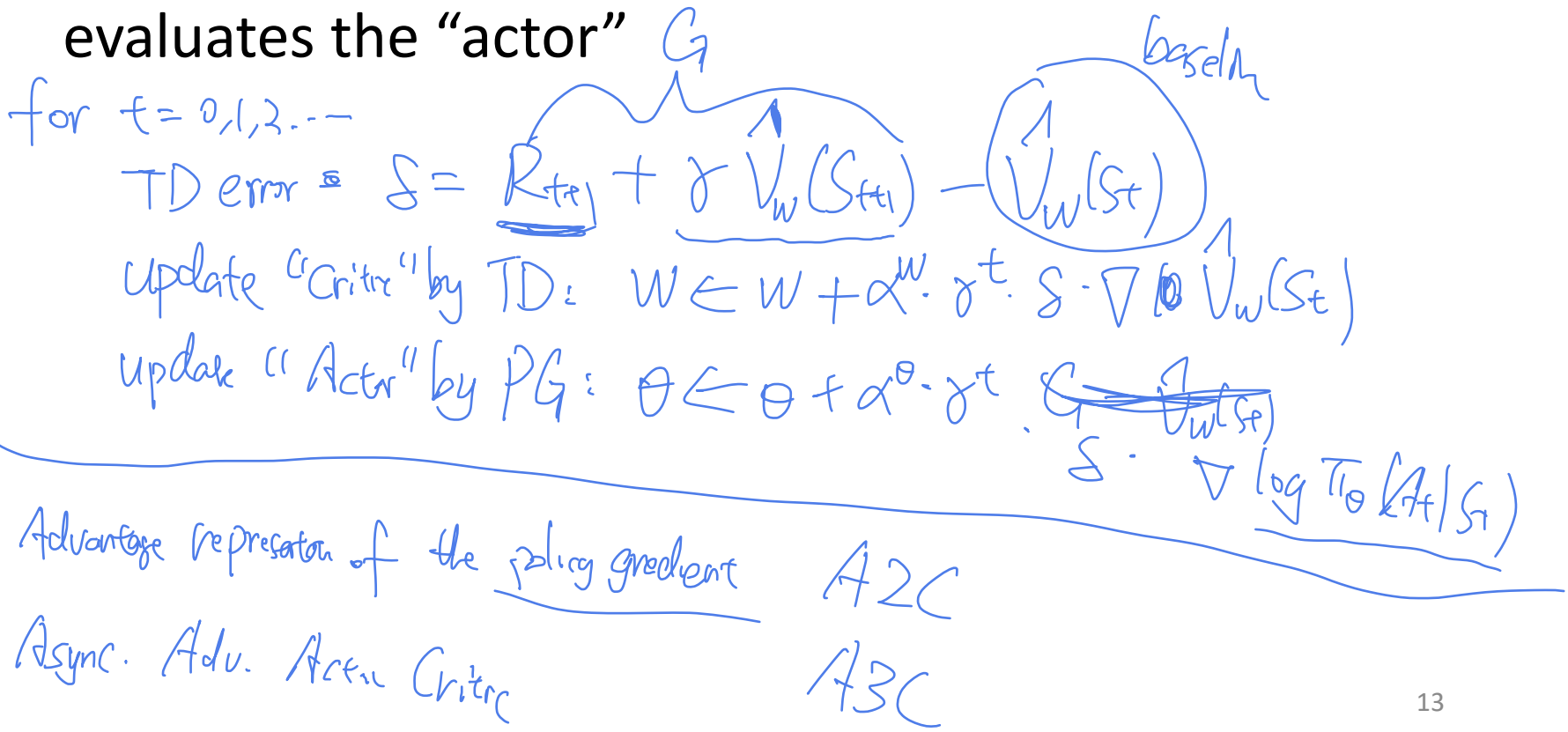
$f = V^{\pi_\theta}$

$$\mathbb{E}\left[\nabla_{\theta}\left[\sum_{a} \pi(a|S_t)\right] f(S_t)\right]$$

Exercise: Calculate the variance when the baseline is chosen to be the correct value function.

Actor-Critic: Learn the baseline and use the baseline for "bootstrapping"

- **Actor:** The policy that performs actions.
- **Critic:** A value function approximation that evaluates the "actor"



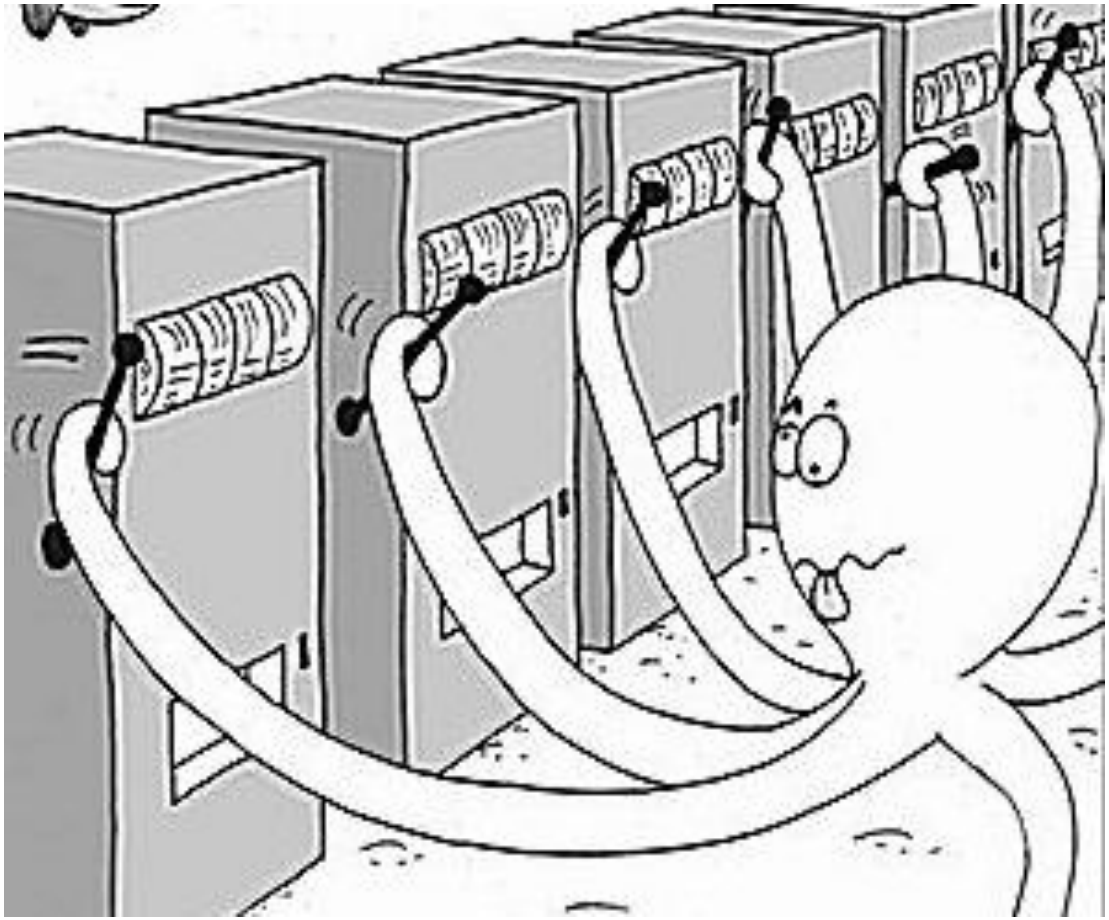
What I did not talk about

- Natural Policy Gradient
  - Extension of Natural Gradient (i.e., Fisher-Scoring)
  - Multiplicative form in updating the policy
  - Function approximation
- Global convergence of PG / NPG
- KL-divergence regularization / imitation learning / conservative policy iteration / trust-region policy iteration

Recap: Let us tackle different aspects of the RL problem one at a time

- Markov Decision Processes: (Lecture 1-3)
  - Dynamics are given no need to learn. planning only.
- RL algorithms: (Lecture 4-5)
  - Ideas on how to use “sampled” transitions / trials-and-errors to learn and plan at the same time (without touching upon exploration).
- **Strategic exploration: (Lecture 6- 9)**
  - Bandits: Explore-Exploit in simple settings
  - RL: Explore-Exploit in Learning MDPs

Slot machines and Multi-arm bandits



Multi-arm bandits: Problem setup

- No state. k-actions $a \in \mathcal{A} = \{1, 2, ..., k\}$
- You decide which arm to pull in every iteration

$$A_1, A_2, ..., A_T$$

- You collect a cumulative payoff of $\sum_{t=1}^{T} R_t$
- The goal of the agent is to maximize the expected payoff.
  - For future payoffs?
  - For the expected cumulative payoff?

How do we measure the performance of an **online learning agent**?

- The notion of "Regret":
  - I wish I have done things differently.
  - Comparing to the best actions in the hindsight, how much worse did I do.

- For MAB, the regret is defined as follow

oracle agent

$$T \max_{a \in [k]} \mathbb{E}[R_t|a] - \sum_{t=1}^{T} \mathbb{E}_{a \sim \pi} \left[\mathbb{E}[R_t|a] \right]$$

$$T \max_{a} \mathbb{E}[R_t|a] - \sum_{t=1}^{T} \mathbb{E}[R_t|A_t]$$

No-regret learning alg $= o(T)$

Greedy strategy

- Expected reward

$$r(s,a) = Q(s,a) = \mathbb{E}[R_t \mid A_t = a].$$

- Estimate the expected reward

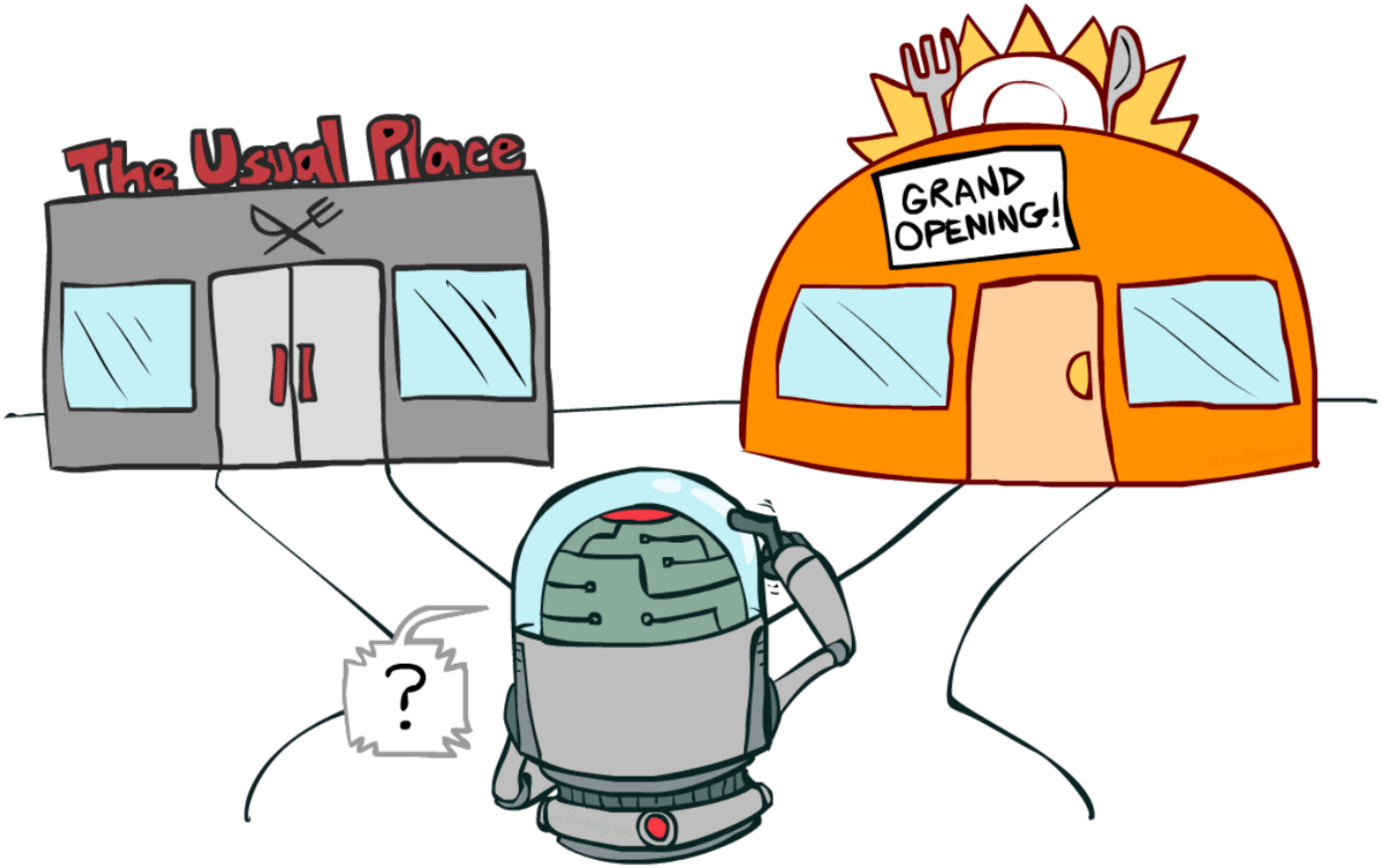
$$\widehat{Q}_t(a) \doteq \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t}$$

$$= \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_i=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_i=a}}$$

- Choose $A_t \doteq \arg\max_a \widehat{Q}_t(a),$

What is the issue with this strategy?

Exploration vs. Exploitation



(Illustration from Dan Klein and Pieter Abbeel's course in UC Berkeley)

Exploration first strategy

- Let's spend the first N step exploring.
  - Play each action for N / k times.

$$\hat{Q}_t(a) = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_i=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_i=a}}$$

- For t = N +1, N+2, …, T:

$$A_t \doteq \arg\max_a Q_t(a),$$

Recap: Concentration inequalities --- finite-sample bounds of LLN and CLT

- **Hoeffding's inequality**: Assume X_1, …, X_n are independent and their support bounded:

$$S_n = X_1 + \cdots + X_n$$

$$\mathrm{P}(S_n - \mathrm{E}[S_n] \geq t) \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right),$$

- Easy version, if $0<X_i<B$, with probability $1-\delta$:

$$\left|\bar{X} - \mathbb{E}[\bar{X}]\right| \leq \sqrt{\frac{B^2}{2n}\log(2/\delta)}$$

Regret analysis of Exploration First

Regret analysis of Exploration First

ε-Greedy strategy: one way to balance exploration and exploitation

- You choose with probability 1- ε

$$A_t \doteq \arg\max_a Q_t(a),$$

- With probability ε, choose an action uniformly at random!
  - Including the argmax.

- Carefully choose ε parameter.

A sketch of the analysis for ε-greedy

- In expectation, each arm is chosen for at least εt times.
- Condition on the number of times, apply Hoeffding's inequality / union bound for all t and a
- Regret bound is

$$\epsilon T \ + \sum_{t=1}^{T} C\sqrt{\frac{k}{\epsilon t}}$$

Optimism-in-the-face of uncertainty: Upper Confidence Bound algorithm

Martingale

- We say that a sequence of r.v. $X_1,\ldots,X_n,\ldots$ is a Martingale if for any n

$$\mathbf{E}(|X_n|) < \infty$$

$$\mathbf{E}(X_{n+1} \mid X_1, \ldots, X_n) = X_n.$$

- Example:
  - Random-walk: Total number of heads minus tails in n coin tosses

Azuma-Hoeffding's inequality

- **Azuma-Hoeffding's inequality**: Assume $X_1, \ldots, X_n$ are Martingale differences

$$S_n = X_1 + \ldots + X_n$$

$$\mathbb{P}\left[S_n \geq \epsilon\right] \leq e^{-\frac{2\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}}$$

- Apply Azuma-Hoeffding's inequality to our problem

Regret analysis of UCB

Regret analysis of UCB

Summary of Exploration in Multi-Armed Bandits

- Explore-First
- eps-greedy
- UCB