CS188 Discussion 6A:
Reinforcement Learning Continued

Austen Liao (austenliao@berkeley.edu)
Slides credit: Joy Liu
July 25, 2023
https://tinyurl.com/austen-su23

Today’s Agenda

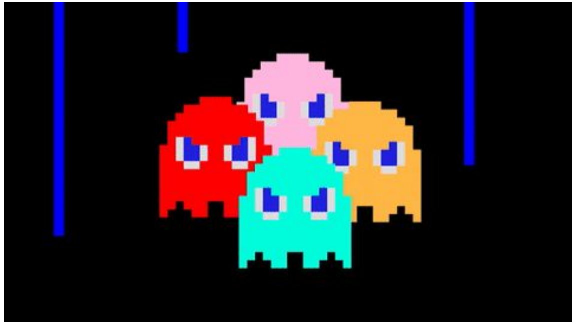
1. Warmup
2. Administrivia
3. Reinforcement Learning

Warmup!

- Make groups of three and discuss for ~1 minute:
  - **name, pronouns, year, etc. (if meeting someone new)**
  - **classes you want to take next semester (alternatively, just any plans you have for the fall)**
  - **anything you did this past weekend**

Administrivia

- **Homework 5** due July 24 (today)
- **Project 5** due July 27 (Thursday)
- **Homework 6** due July 28 (Friday)
- For any DSP or extension requests, email cs188@berkeley.edu



Reinforcement Learning Review

Overview of RL

- Problem set-up: like MDPs, except we don’t know T(s, a, s’) or R(s, a, s’)

- Passive RL: how to learn values from experiences under a given policy
  - Model-based: keep track of reward and transition counts for every sample (s, a, s’), then solve MDP using value/policy iteration
  - Model-free: directly learn V or Q using Direct Evaluation or Temporal Difference Learning

- Active RL: how to learn from experiences and improve our policy directly
  - Q-learning

$$V^{\pi}(s) = \sum_{s'} T(s,\pi(s),s')[R(s,\pi(s),s') + \gamma V^{\pi}(s')]$$

Model-free passive RL: TD Learning

Idea: learn at every timestep (under the policy we follow)

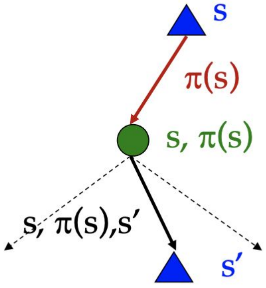
1. Initialize all $V^{\pi}(s)$ to 0, determine $\pi(s)$ and α in (0,1]
2. Repeat
   a. Take sample (s, π(s), s')

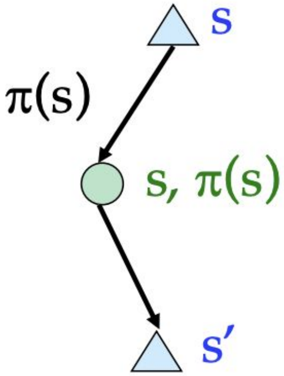
   $$\boxed{\text{sample} = R(s,\pi(s),s') + \gamma V^{\pi}(s')}$$

   b. Incorporate sample into exponential moving average of $V^{\pi}(s)$

   $$\boxed{V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + \alpha \cdot \text{sample}}$$

   (Slowly decrease α from 1 to 0)





Active RL: Q-learning

Idea: learn Q-values so we can extract policy

1. Initialize all Q(s, a) to 0 and α in (0,1]
2. Repeat
    a. Sample (s, a, s')

    $$\boxed{\text{sample} = R(s,a,s') + \gamma \max_{a'} Q(s',a')}$$

    b. Incorporate sample into exponential moving average of $V^{\pi}(s)$

    $$\boxed{Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha \cdot \text{sample}}$$

If we explore for long enough and decrease α slowly enough, Q-learning will converge on the optimal Q-values and optimal policy

Active RL: Approximate Q-learning

Idea: Q-learning is not feasible if there are thousands of states
Fix: store each state as linear combinations of features

Every V(s) & Q(s, a) has a feature vector, and you want to tune the weight vector

$$
\begin{aligned}
V(s) &= w_1 \cdot f_1(s) + w_2 \cdot f_2(s) + \ldots + w_n \cdot f_n(s) = \vec{w} \cdot \vec{f}(s) \\
Q(s,a) &= w_1 \cdot f_1(s,a) + w_2 \cdot f_2(s,a) + \ldots + w_n \cdot f_n(s,a) = \vec{w} \cdot \vec{f}(s,a)
\end{aligned}
$$

Update rule:

$$\text{difference} = [R(s,a,s') + \gamma \max_{a'} Q(s',a')] - Q(s,a)$$

$$w_i \leftarrow w_i + \alpha \cdot \text{difference} \cdot f_i(s,a)$$

Exploration vs Exploitation

- How do we ensure that we’ve explored a sufficient amount?

Strategies:

Exploration vs Exploitation

- How do we ensure that we've explored a sufficient amount?

Strategies:

- ε-greedy strategy - every timestep,
  - "explore" w. p. ε (choose action randomly)
  - "exploit" w. p. 1 - ε (follow established policy)

Exploration vs Exploitation

- How do we ensure that we’ve explored a sufficient amount?

Strategies:

- ε-greedy strategy - every timestep,
  - “explore” w. p. ε (choose action randomly)
  - “exploit” w. p. 1 - ε (follow established policy)
- exploration function strategy - bias toward less explored regions
  - exploration function:

$$f(s,a) = Q(s,a) + \frac{k}{N(s,a)}$$

  - new update rule:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha \cdot [R(s,a,s') + \gamma \max_{a'} f(s',a')]$$

Attendance: https://tinyurl.com/cs188su23
Feedback: https://tinyurl.com/austen-su23-new