ROB 537:
Learning-Based Control

Week 4, Lecture 1
Reinforcement Learning

Announcements:
HW 2 due: 10/16
Project Background due: 10/23

Reading:
Kaelbing et al.
Stone et al.

Oregon State University

So far

- Neural Networks
  - Learn an unknown function
    - Regression
    - Classification
  - Direct mapping of input to output
    - Features to class (classification)
    - Features to real value (regression)
    - State to action (control)

Oregon State University

Reinforcement Learning

- Don't directly map state to action
- Learn expected rewards
  - For state
  - For state , action pairs

Oregon State University

Reinforcement Learning Concept

- Learn from interactions with the environment
  - Take action
  - Receive feedback from the environment
  - Modify your behavior
  - Achieve some goal
- Examples:
  - Baby playing
    - Connection to the environment guides sensory input/output
  - Driving a car
    - Learn from interaction

Oregon State University

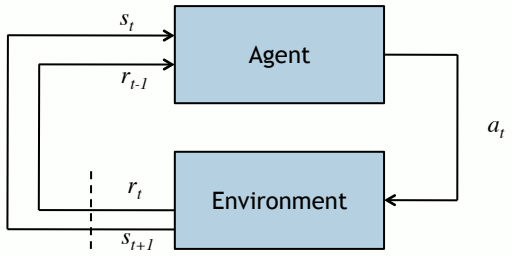
Reinforcement Learning

- The Reinforcement Learning Concept
- N-armed Bandit Problem
- Value Functions

- Markov Decision Processes

- Monte Carlo Methods
- Sarsa Learning
- Q-Learning

Agent Environment Interaction in RL



Policy : Map States to Actions

System dynamics

$$s_t \times a_t \rightarrow s_{t+1}, r_t$$

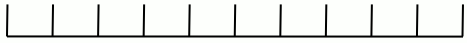
Policy (π_t): map states to probabilities of taking those actions

$$\pi_t(s,a) = P(a_t = a \mid s_t = s)$$

Oregon State University

N-Armed Bandit

- Think multiple armed slot machine



  - N actions
  - Each slot has a reward (randomly chosen from a distribution)
  - Aim: maximize long term reward
  - Which arm to pull?

Oregon State University

Value Function

- Reward is what you receive for taking an action
- Value:
  - Actual "worth" of that action
  - Long term reward
- N-armed bandit:
  - Action: which arm to pull
  - Reward: immediate gain on this time step
  - Value: long term payoff of a particular action

Oregon State University

N-Armed Bandit

- Let's say the actual value of an action is $V^*(a)$
- If action *a* has been sampled k_a times
  - The estimate at time step *t* is $V_t(a)$:

$$V_t(a) = \frac{r_1 + r_2 + r_2 + \cdots + r_{k_a}}{k_a}$$

  - By the law of large numbers

$$\lim_{t \to \infty} V_t(a) \to V^*(a)$$

Oregon State University

Computing V

- For large values of t:
  - Need to keep all values in memory.
  - How about we do this incrementally?
  - V_{k+1}: average at k+1
  - V_k : average after k

$$V_{k+1} = \frac{r_1 + r_2 + r_3 + \cdots + r_{k+1}}{k+1}$$

Oregon State University

Computing V

- For large values of t:
  - Need to keep all values in memory.
  - How about we do this incrementally?
  - V_{k+1}: average at k+1
  - V_k : average after k

$$\begin{aligned} V_{k+1} &= \frac{r_1 + r_2 + r_3 + \cdots + r_{k+1}}{k+1} \\ &= \frac{1}{k+1}\left(r_{k+1} + \sum_{i=1}^{k} r_i\right) \\ &= \frac{1}{k+1}\left(r_{k+1} + kV_k\right) \\ &= \frac{1}{k+1}\left(r_{k+1} + kV_k + V_k - V_k\right) \\ &= \frac{1}{k+1}\left(r_{k+1} + (k+1)V_k - V_k\right) \\ &= V_k + \frac{1}{k+1}\left(r_{k+1} - V_k\right) \end{aligned}$$

Oregon State University

Summary

- Result:

$$V_{k+1} = V_k + \frac{1}{k+1}\left(r_{k+1} - V_k\right)$$

- Key Reinforcement Learning Concept:

NewEstimate $\leftarrow$ *OldEstimate + Stepsize (Target – OldEstimate)*

Oregon State University

What if process is non-stationary?

- How about a fixed "stepsize"

$$V_{k+1} = V_k + \alpha\left(r_{k+1} - V_k\right)$$

- Impact:
  - New observations count more
  - Slowly forget old info
  - Track non-stationary process

Oregon State University

Example: N-Armed Bandit

- Given values at time t:

0.1	0.4	0	0.05	0.1	0	0.05	0.1	0	0.2

  - Greedy selection:
    - Action 2 (highest value)
  - ε-greedy selection:
    - Action 2 with probability (1-ε)
    - Another action with probability ε
  - Softmax:
    - Each action with probability based on its value

Oregon State University

Example: N-Armed Bandit, Softmax

- Given values at time t:

0.1	0.4	0	0.05	0.1	0	0.05	0.1	0	0.2

  - Softmax:
    - Select each action with probability $p_t(a)$ based on:
      - Temperature t
      - The Q-value of that action at that time step

$$p_t(a) = \frac{\exp\left(\frac{V_t(a)}{\tau}\right)}{\sum_b \exp\left(\frac{V_t(b)}{\tau}\right)}$$

      - If $\tau \rightarrow 0$, then greedy selection (highest value has probability 1)
      - If $\tau \rightarrow \infty$, then p(a) = 1/n for all actions

Oregon State University

Initial Values?

- How do we start?
- What are initial Q values?
  - Zero
  - Random
  - Optimistic
    - Encourage exploration of actions not yet taken

Oregon State University

Interesting Questions

- What if reward is not immediate?
  - Reward/value for sequences of actions
  - How do I assign/blame to sequences of actions
- Do I choose best reward?
  - What about potential for better reward of unexplored action
  - Exploration/Exploitation
    - Explore: try a new action
    - Exploit: repeat (so far) best action

Oregon State University

Rewards and Time Horizons

- Maximize reward
  - What does this really mean?
  - Maximize expected reward
    - Simplest case: sum of rewards:

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \cdots + r_T$$

    - Makes sense when there is a natural final step *T*
    - Episodic task:
      - Break learning into episodes of *T* steps
      - Example: Play a game

Oregon State University

Rewards and Time Horizons

What if there is no final time T ?

- Maximize discounted sum of rewards

$$R_t = r_{t+1} + \gamma\ r_{t+2} + \gamma^2 r_{t+3} + \cdots + \gamma^{T-t-1} r_T$$
$$= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

- A reward *k* step in the future is worth only γ^{k-1} times what it would be worth now
- $\gamma < 1$ means the infinite sum has a finite value as long each reward is bounded
- $\gamma = 0$ means agent is myopic: care only about immediate rewards

Oregon State University

Markov Property

- Markov property:
  - A state that retains all the information relevant to future actions and rewards is said to be *Markov*, or to have the *Markov property*
  - Key: How you got to the state doesn't matter
  - Example: Chess board: all relevant info is there
  - Important in Reinforcement Learning because decisions and values are based on state

Oregon State University

Markov Property

- Markov property depends on representation of problem, not the problem
- Example: Pole balancing
  - State: location of pole tip
  - Not Markov
  - State: location and velocity of pole tip
  - Not Markov
  - State: location/velocity/acceleration of pole tip, location/velocity of cart
  - Markov (approximation)

Oregon State University

Markov Decision Processes

- Markov Decision Process (MDP)
- Finite MDP (finite number of states) defined by:
  - Set of states *S*
  - Set of actions *A*
  - Transition probabilities

    $P^{a}_{ss'} = \Pr\{s_{t+1} = s' \mid s_t = s ; a_t = a\}$

  - Rewards

    $R^{a}_{ss'} = E[r_t \mid s_t = s ; a_t = a ; s_{t+1} = s']$

  - Transition probabilities and Rewards fully specify an MDP

Markov Decision Processes

- Markov Decision Process (MDP)
- Finite MDP denoted by a 4-tuple:

$$(S,\ A,\ P_a(s,s'),\ R_a(s,s'))$$