Course 2, Module 1
Monte Carlo

CMPUT 397
Fall 2019

Live question submission in class

- goto: https://www.tricider.com/brainstorming/3ZFvVmeuw4N
- Submit a question (anonymous if you like)

These submissions are not acceptable

- What similarities are there between the **monte carlo method** and 10 **armed bandit**?
- **what is Monte Carlo's disadvantages**
- MDPs with large state-spaces we prefer **asynchronous DP** ...
- How can RL apply to real life?
- What's the difference between the Monte Carlo value function and the **generalized value function**?
- Is it true that the first few episodes of the **off policy model don't have a policy**? Will the **off policy model** has a policy until it has reached every state?
- What is MC suitable for all situation?

In general, the discussion topics you submit should not prove you didn't watch the videos

Clarifications

- When do we **need exploration**?
  - >> Whenever we don't have a model
- What does it mean to **not have the model**?
  - >> What is the model for you interacting with your friends on the weekend? Do you know the transition probs?
- When would learning a value of a state **independent of other states** be useful? As one of the major differences between MC and DP.
  - >> Both are tabular methods so they both learn independent values for each state. DP does bootstrapping because states that transition to each other have similar values as shown in the Bellman Equation

Clarifications

- **How many episodes** do we need to sample from so that exploring starts would guarantee the agent will explore all the state actions pairs?

- Do we need **1000's of episodes from each state** to learn the value function with MC?

    - >> It depends. Lot's of variance in transitions and rewards, like blackjack or the gamblers problem -> lot's. Nearly deterministic problems-> very few

    - >> In some domains, we don't know the model so we don't have much choice

- Can we use MC methods in **non-stationary tasks**? What would be the challenges?

    - >> Yes. It learns from interaction data! But just like in bandits, sample averages are bad for changing environments. What else could we do?

Clarifications

- Doesn't computing the **importance sampling ratio** require **knowledge** of the **optimal policy**?
    - >> We need to know the probabilities of the current target and behavior policies
    - >> if the target policy pi is greedy with respect to Q(s,a) what are the probabilities for each action in state s? pr(a*|S) = 1.0, pr(b|S) = 0.0, for all b not equal to a*
    - >> if the behavior policy is random what are the probabilities for each action in s?

Clarifications

- Why **can't we get** the **optimal policy** with **epsilon-soft** MC control
    - >> we can only get pi* is we act greedily with respect to q*
    - >> if we learn an epsilon soft policy we don't learn q*
- Why would we want an **epsilon-soft policy instead of an epsilon-greedy policy**? It just seems like unnecessary complexity.
    - >> epsilon soft policies are the general class. epsilon greedy is one example, there are others
    - >> we talk about epsilon soft MC control to demonstrate the MC control can be combined with many different types of stochastic policies

Clarifications

- Doesn't the **coverage** condition in off-policy mean pi must be equal to b?
  - >> it just means if pi chooses some action a in state s sometimes, then b must also sometimes choose a in state s also
- How do we **choose the behavior policy** b?
  - >> there is no easy answer. We want b to achieve coverage. We might also want it to be smart: not explore actions in states we already know are bad; not to explore actions in states where we already have a good estimate of Q(s,a)

A family of Monte Carlo methods

Exploring Starts
MC Control

on-policy:

we learn about
the optimal policy,
we follow the optimal policy

Use when:
can easily start the
agent in any state

Epsilon-soft
MC Control

on-policy:

we learn about the epsilon-soft optimal policy,
we follow the epsilon-soft
optimal policy

Use when:
can't do random starts,
optimal policy not
essential

Off-policy
MC Control

off-policy:

we learn about the optimal
policy,
we follow an exploratory
policy (e.g., epsilon-greedy)

Use when:
no obvious restrictions

Clarifications

- How do we know importance sampling will work? >> It is unbiased. There is a proof
- How do we know the value function we learn is correct? >> We need v_pi
- Do we face overfitting with MC? >> No. This is not an issue in the tabular setting
- Can we combine UCB with MC control? >> Yes! How might that work?
- Can we decay epsilon? >> Yes!
- Can we update the behavior policy? >> Yes. It's complicated. Not commonly done

Discussion topics for today

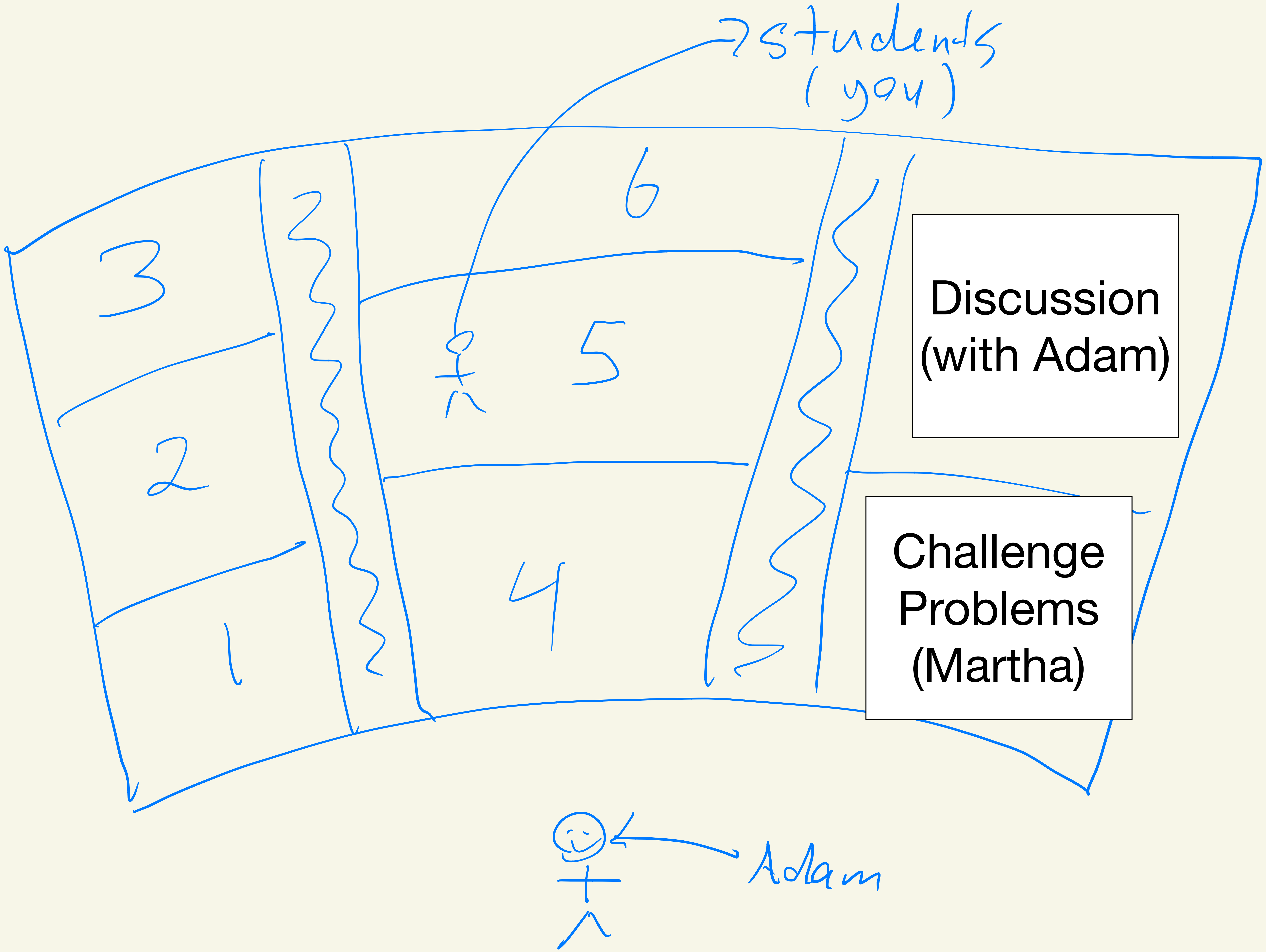
1. In some applications (for example, self-driving cars), there will be states where choosing actions randomly can be dangerous. Is it feasible to train agents in a simulated world using Monte Carlo method and have them to apply the optimal greedy policy that they found to the real world? Is that going to be safer or more dangerous for the people? How do we decide **which states** to update in **asynchronous** DP?

2. Think of some real world examples and explain why exploring starts is an impossible/impractical method of exploration (other than self-driving cars).

3. How can we mix Monte-Carlo and Dynamic Programming?

Challenge Questions

- **Challenge Question 1**: How do we handle **continuing tasks** with MC?
- **Challenge Question 2:**
    - When we had the model, we used DP to find the value function vpi.
    - Without the model, we use sampled experience from the environment, and do Monte Carlo updates
    - **Can you use Monte Carlo updating, if you have a model?** If so, how? Is there more than one way?



Discussion (with Adam)

Challenge Problems (Martha)