Course 2, Module 3
Temporal Difference Learning Methods for Control

CMPUT 397
Fall 2019

Any questions about course admin?

- Link for questions:
  - **http://www.tricider.com/brainstorming/35B8Mn3NZ5B**

Review of Course 2, Module 3
TD Control

Video 1: Sarsa: GPI with TD

- Building an algorithm to find near optimal policies: SARSA (**S**tate, **A**ction, **R**eward, Next **S**tate, **A**ction). Combining the ideas of *policy evaluation, policy improvement, TD,* and *epsilon-soft policies*

- Goals:

  - explain how **generalized policy iteration** can be used with TD to find improved policies

  - Describe the Sarsa Control algorithm

Video 2: Sarsa in the Windy Grid World

- We ran a fun **experiment with Sarsa** on a fancy gridworld
- Goals:
  - Understand how the Sarsa control algorithm operates in an example MDP.
    - the Windy Gridworld
  - Gain experience analyzing the performance of a learning algorithm.
    - understanding the plot of cumulative episodes completed vs steps

Video 3: What is Q-learning

- Just the most famous RL algorithm! Similar to SARSA, but learns the **optimal policy**
- Goals:
  - Describe the Q-learning algorithm
  - Explain the relationship between Q-learning and the Bellman optimality equations

Video 4: Q-learning in the Windy Gridworld

- How does Q-learning work in practice? We get some insight with an **experiment comparing with SARSA**
- Goals:
  - Gain insight into how Q-learning performs in an example MDP
  - Gain insight into the **differences between Q-learning and Sarsa**.

Video 5: How is Q-learning Off-policy?

- Q-learning **learns about** the greedy policy (which eventually becomes π^*), while **following a different policy** ε-greedy. That is off-policy, but there are no importance sampling corrections!

- Goals:

  - Understand how Q-learning can be off-policy **without using importance sampling**

  - Describe how learning on-policy or off-policy might affect performance in control.

    - SARSA (on-policy learning), can be better!

Video 6: Expected SARSA

- **A new TD Control method**! Uses the probability of each action under the current policy in its update!
- Goals:
  - explain the Expected Sarsa algorithm.

Video 7: Expected SARSA in the Cliff World

- Why all the fuss about Expected Sarsa? We find out with an **experiment** in another gridworld: The cliff world. Spoiler: **Expected Sarsa learns faster** AND **is more robust to our choice of alpha**

- Goals:

  - Describe Expected Sarsa's behaviour in an example MDP.

  - And Empirically compare Expected Sarsa and Sarsa

Video 8: The Generality of Expected SARSA

- Expected SARSA is pretty neat! It can perform better than either SARSA or Q-learning. In addition, the algorithm can be **used in different ways**
- Goals:
  - Understand how Expected Sarsa can do **off-policy** learning without using importance sampling
  - Explain how Expected Sarsa **generalizes Q-learning**

Terminology Review

- TD methods we have learned about are **tabular, one-step, model-free** learning algorithms

- **Tabular:** we store the value function in a table. One entry in the table per value, so each value is stored independently of the others. We are implicitly assuming the state-space ($\mathcal{S}$) is small

- **One-step**: we update a single state or state-action value on each time-step. Only the value of Q(S,A) from S -- A --->S',R. We never update more than one value per learning step

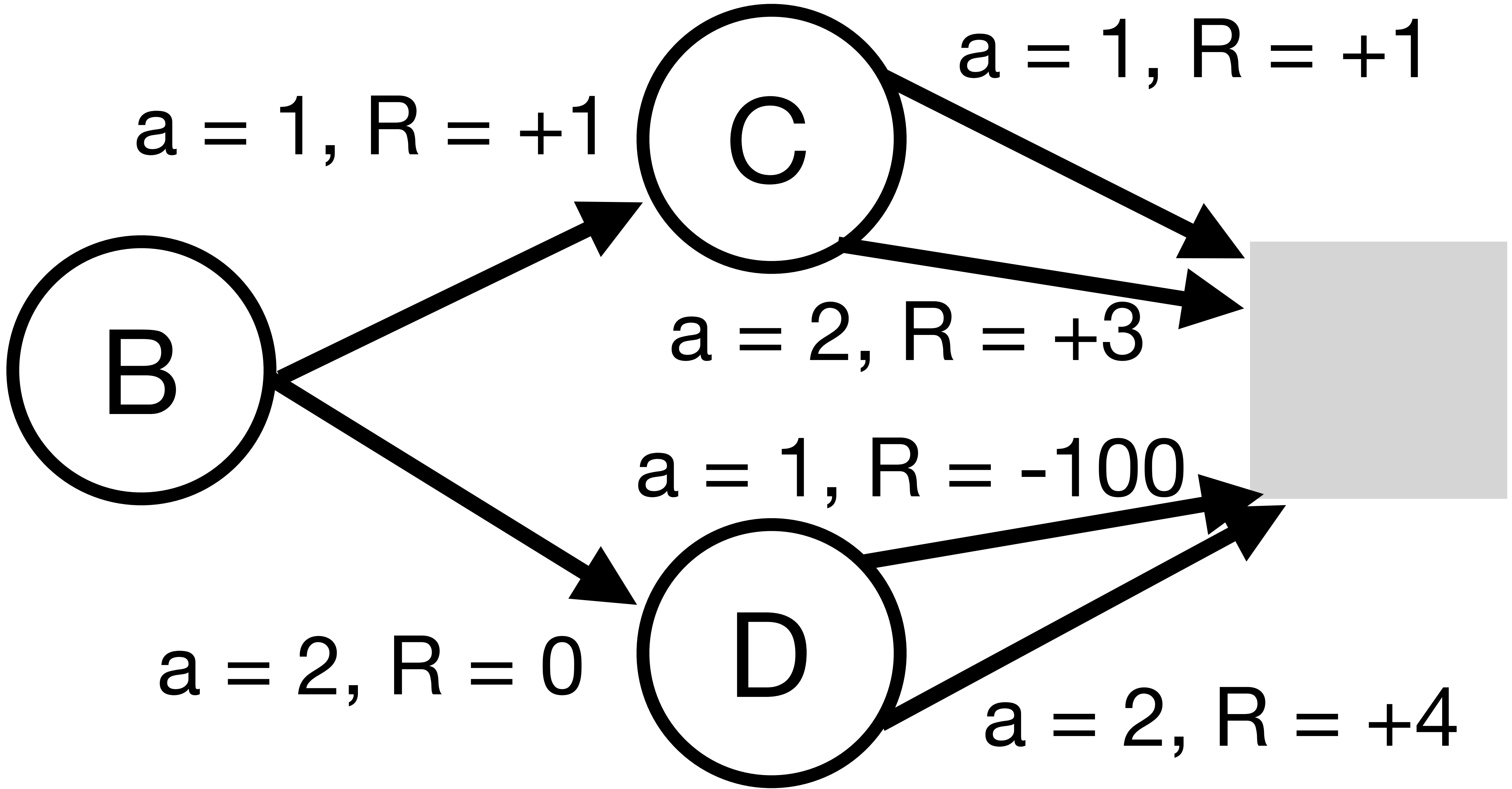
- **Model-free**: we don't assume access to or make use of a model of the world. All learning is driven by sample experience. Data generated by the agent interacting with the environment

1. Consider the following MDP, with three states B, C and D ($\mathcal{S} = \{B, C, D\}$), and 2 actions ($\mathcal{A} = \{1, 2\}$), with $\gamma = 1.0$. Assume the action values are initialized $Q(s, a) = 0 \;\forall\; s \in \mathcal{S}$ and $a \in \mathcal{A}$. The agent takes actions according to an ϵ-greedy with $\epsilon = 0.1$.

(a) What is the optimal policy for this MDP and what are the action-values corresponding to the optimal policy: $q^*(s, a)$?

(b) Imagine the agent experienced a single episode, and the following experience: $S_0 = B, A_0 = 2, R_1 = 0, S_1 = D, A_1 = 2, R_2 = 4$. What are the Sarsa updates during this episode? Start with state B, and perform the Sarsa update, then update the value of state D.

Deterministic transitions



(b) Imagine the agent experienced a single episode, and the following experience: $S_0 = B, A_0 = 2, R_1 = 0, S_1 = D, A_1 = 2, R_2 = 4$. What are the Sarsa updates during this episode? Start with state B, and perform the Sarsa update, then update the value of state D.

(c) Using the sample episode above, compute the updates Q-learning would make. Again start with state B, and then state D.

(d) Let's consider one more episode: $S_0 = B, A_0 = 2, R_1 = 0, S_1 = D, A_1 = 1, R_2 = -100$. What would the Sarsa updates be? And what would the Q-learning updates be?

(e) What policy does Q-learning converge to? What policy does Sarsa converge to?