Mini-Course 1, Module 1
Monte Carlo Methods for Prediction & Control

CMPUT 397
Fall 2020

October 5, 2020

- Office hours split this week to give an earlier session
  - Tuesday at 10 am -11 am MDT
  - Wednesday at 2 pm - 3 pm MDT
- Any questions about course admin?

Review of C2M1
Monte Carlo

Video 1: What is Monte Carlo?

- The term “Monte Carlo” is often used more broadly for any estimation method that relies on **repeated random sampling**
- In RL, Monte-Carlo methods allow us to **estimate values** directly from experience: from **sequences of states, actions, and rewards**.
- Goals:
  - Understand how Monte-Carlo methods can be used to **estimate** value functions from **sample interaction**
  - **Identify problems** that can be solved using Monte-Carlo methods

Video 2: Using Monte Carlo for Prediction

- Discussed the **Monte Carlo Policy Evaluation algorithm.** We also looked at a **results** of using MC to evaluate one particular policy in Blackjack
- Goals:
  - Use Monte Carlo prediction to estimate the value function for a **given policy**.

Monte Carlo pseudocode

Input: a policy π **to be evaluated**

Initialize:

$V(s) \in \mathbb{R}$**, arbitrarily, for all** $s \in S$

$Returns(s) \leftarrow$ **an empty list, for all** $s \in S$

Loop forever (for each episode):

Generate an episode following $\pi : S_0, A_0, R_1, S_1 \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \ldots, 0$

$G \leftarrow \gamma G + R_{t+1}$

Append G **to** $Returns(S_t)$

$V(S_t) \leftarrow$ **average**$(Returns(S_t))$

Every-Visit Monte Carlo prediction, for estimating V

Input: a policy π **to be evaluated**

Initialize:

$V(s) \in \mathbb{R}$**, arbitrarily, for all** $s \in S$

$Returns(s) \leftarrow$ **an empty list, for all** $s \in S$

Loop forever (for each episode):

Generate an episode following $\pi : S_0, A_0, R_1, S_1 \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \ldots, 0$

$G \leftarrow \gamma G + R_{t+1}$

Append G **to** $Returns(S_t)$

$V(S_t) \leftarrow$ **average**$(Returns(S_t))$

Video 3: Using Monte Carlo to Estimate Action-Values

- How to estimate q_π instead of v_π with MC: Q(St, At) instead of V(St). We also tackled the exploration problem in MC.
- Goals:
    - Estimate **action-value functions** using Monte Carlo and
    - Understand the importance of **maintaining exploration** in Monte Carlo algorithms

Video 4: Using Monte Carlo Methods for Generalized Policy Iteration

- Our first **control Monte Carlo** algorithm. Using **Exploring Starts** to handle the exploration problem
- Goals:
  - Understand how to use Monte Carlo methods to implement a **GPI algorithm**.

Video 5: Solving the Blackjack Example

- Using Monte Carlo Control with Exploring Starts to learn an optimal policy in Blackjack!

- **Goals**:

    - Apply Monte Carlo with exploring starts to solve an example MDP.

Video 6: Epsilon-Soft Policies

- Exploring starts is not always the best idea. Think of estimating the value function for a car on a freeway. Turns out we can combine Monte-Carlo control with epsilon-greedy

- **Goals**:

  - Understand why **Exploring Starts can be problematic in real problems**

  - Describe an alternative exploration method for Monte Carlo control, using **Epsilon-soft policies**

Video 7: Why Does Off-Policy Learning Matter?

- Off-policy learning is **another way to handle exploration**. You have one policy called the **behavior policy** in charge of acting, and another policy, called the **target policy** that you want to learn the value function for.
- **Goals**:
  - Understand how off-policy learning can **help** deal with the **exploration problem**.
  - Examples of target policies
  - and examples of behavior policies.

Video 8: Importance Sampling

- **Statistics review:** estimating the expected value of one random variable, with samples drawn according to a different distribution: estimate $E_\pi[X]$ with samples drawn according to distribution *b, where* π != b
- **Goals**:
    - use **importance sampling** to estimate the expected value of a target distribution using samples from a different distribution.

Video 9: Off-Policy MC Prediction

- Now that we know how to use importance sampling, we can use it with Monte Carlo to estimate v_π off-policy. We will do off-policy control later. We keep it simple for now!
- **Goals**:
  - Understand how to **use importance sampling to correct returns**
  - And you will understand how to modify the **Monte Carlo prediction** algorithm for off-policy learning.

Practice Question

. (*Exercise 5.5 S&B*) Consider an MDP with a single nonterminal state s and a single action that transitions back to s with probability p and transitions to the terminal state with probability $1-p$. Let the rewards be $+1$ on all transitions, and let $\gamma = 1$. Suppose you observe one episode that lasts 10 steps, with return of 10. What is the (every-visit) Monte-carlo estimator of the value of the nonterminal state s?

Generate an episode following $\pi : S_0, A_0, R_1, S_1 \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$
    $G \leftarrow \gamma G + R_{t+1}$
    Append G **to** $Returns(S_t)$
    $V(S_t) \leftarrow$ **average**$(Returns(S_t))$

Terminology Review

- In Monte Carlo there are **no models, and no bootstrapping**
- **Experience**: data generate by the agent taking actions and getting reward feedback for the action it selected.
  - different from what Dynamic Programming does. DP updates the value of states using p(s',r|s,a). DP knows all the rewards in each state via p
- **Sample episodes**: starting in the start state, run policy pi (select actions according to pi) until termination, recording the states, actions, and rewards observed
- MC methods update the value estimates on an **episode-by-episode** basis. Must wait until the end of an episode to update the values of each state the agent observed

Terminology Review (2)

- **Maintaining exploration:** Why we need exploration in MC. Assume pi never takes action b in state S. If we want to estimate q(S,b) we will have no data about the reward you get from state S when pi chooses action b
- **Exploring starts:** every episode must begin in a random state, and the first action must be randomly selected, even if that action is not what pi would do
  - guarantees we visit every state-action pair
- **Epsilon-soft policies:** a stochastic policy. A policy where each action is selected with at least epsilon probability. (e.g., epsilon-greedy)

Terminology Review (3)

- **Off-policy:** learning about one policy, while following another
  - e.g., learning the value function for the optimal policy (q*) while following some exploration policy b (i.e. b=random_policy)
- **Target policy:** the policy you want to learn *about. W*e always call it pi. We either want to learn v_π or (q* and pi*)
- **Behavior policy:** the policy used to select actions, to generate the data. We always call it *b.* It is usually an exploratory policy (e.g., epsilon-greedy with respect to Q)
- **Importance sampling:** a statistical technique for estimating the expected value when the samples used to compute the average don't match the distribution you want.

Slido question: On-policy vs Off-policy

- “How do we determine what the target policy should be in off-policy learning? From the videos, we have assumed that an optimal policy is the target, but how do we know what the optimal policy is?”