# Reinforcement Learning

## Lesson 5: Monte Carlo Methods

**Edoardo Fazzari, 2022**

# Overview

- Monte Carlo Prediction
- Monte Carlo Estimation
- Monte Carlo Control
- Monte Carlo Control without Exploring Starts
- Off-policy Prediction via Importance Sampling
- Incremental Implementation
- Off-policy Monte Carlo Control

# Monte Carlo methods

## An introduction

- Monte Carlo methods require only *experience*
    - This requires no prior knowledge of the environment's dynamics, yet can still attain optimal behavior
- Learning form *simulated* experience is also powerful, although a model is required
    - The model need only generate sample transition, not the complete probability distributions of all possible transitions (as required for DP)
- Monte Carlo methods are ways of solving the RL problems averaging sample results
- We define Monte Carlo methods only for episodic tasks
- Only on the completion of an episode value estimates and policies are changed

# Monte Carlo Prediction

# Two MC methods

- Suppose we wish to estimate $v_\pi(s)$ given a set of episodes obtained by following $\pi$ and passing through *s*
    - *s* may be visited multiple times in the same episode
    - Let us call the first time it is visited in an episode the *first visit to s*
- The *first-visit MC method* estimates $v_\pi(s)$ as the average of the returns following first visit to *s*
- The *every-visit MC method* averages the returns following all visits to *s*
- Both methods converge to $v_\pi(s)$ as the number of visits (or first visits) to *s* goes to infinity

# First-visit MC Prediction

## Estimating $V \approx v_\pi$

Input: a policy $\pi$ to be evaluated

Initialize:

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

$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):

Generate an episode following $\pi$: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

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

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

Unless $S_t$ appears in $S_0, S_1, \ldots, S_{t-1}$:

Append $G$ to $Returns(S_t)$

$V(S_t) \leftarrow \text{average}(Returns(S_t))$

# First-visit MC

## Convergence Considerations

- In this case each return is an independent, identically distributed estimate of $v_\pi(s)$ with finite variance
- By the law of large numbers the sequence of averages of these estimates converges to their expected value
- Each average is itself an unbiased estimate
    - The standard deviation of its error falls as $1/\sqrt{n}$ (*n* number of returns averaged)

- [*Every-visit MC estimates* converge quadratically to $v_\pi(s)$]

# Important fact

- An important fact about Monte Carlo methods is that the estimates for each state are independent
    - Monte Carlo do not *bootstrap!*
- *Note*: the computational expense of estimating the value of a single state is indecent of the number of states
    - This can make MC methods attractive when one requires the value of only one or a subset of states

# Monte Carlo Estimation of Action Values

# Without a model

- Without a model state values alone are not sufficient
    - One must explicitly estimate the value of each action in order for the values to be useful in suggesting a policy
- One of our primary goals for Monte Carlo methods is to estimate $q_*$
    - To achieve this, we consider the policy evaluation problem for action values

# Policy evaluation problem for action values

**Estimate** $q_\pi(s, a)$

- MC methods for this are essentially the same as for state values
- A state-action pair *s, a* is said to be visited in an episode if ever the state *s* is visited and action *a* is taken in it
    - The *every visit MC method*:
        - Estimates the value of a state-action pair as the average of the returns that have followed all the visits to it
    - The *first-visit MC method*:
        - averages the returns following the first time in each episode that the state was visited and the action was selected
- These methods converge quadratically

# Maintaining exploration

- The only complication is that many state-action pairs may never be visited
- If $\pi$ is a deterministic policy, then in following $\pi$ one will observe returns only for one of the actions from each state
- With no returns to average the Monte Carlo estimates of the other actions will not improve with experience
- This is the general problem of *maintaining exploration*
    - For policy evaluation to work for action values, we must assure continual exploration
        - *Exploring start assumption*: the episodes start in a state-action pair, and every pair has a nonzero probability of being selected as the start
        - This guarantees that all state-action pairs will be visited an infinite number of times in the limit of an infinite number of episodes

# Monte Carlo Control

# Assumption made

- Consider policy improvement
    - MC methods can be used to find optimal policies given only sample episodes and no other knowledge of the environment's dynamics
- We made two assumptions above in order to easily obtain the guarantee of convergence for the MC methods:
    - Episodes have exploring starts
    - Policy evaluation could be done with an infinite number of episodes *(let's remove this assumption now)*

# Two ways to solve the problem

## First approach

- One is to hold firm to the idea of approximating $q_{\pi_k}$ in each policy evaluation
    - Measurements and assumptions are made to obtain bounds on the magnitude and probability of error in the estimates, and then sufficient steps are taken during policy evaluation to assure that the bounds are sufficiently small
- This approach can probably be made completely satisfactory in the sense of guaranteeing correct convergence up to some level of approximation
- However, it is also likely to require far too many episodes to be useful in practice on any but the smallest problems

# Two ways to solve the problem

## Second approach

- We give up trying to complete policy evaluation before returning to policy improvement
- On each evaluation step we move the value function *toward* $q_{\pi_k}$, but we do not expect to actually get close except over many steps

- One extreme form of this idea is value iteration, in which only one interaction of iterative policy evaluation is performed between each step of policy improvement

# Monte Carlo ES (with Exploring Starts)

- For MC policy interaction it is natural to alternate between evaluation and improvement on an episode-by-episode basis
    - After each episode, the observed returns are used for policy evaluation, and then the policy is improved at all the states visited in the episode
- This approach is called *Monte Carlo ES*

# Monte Carlo ES (with Exploring Starts)

## Pseudocode

Initialize:

$\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$
$Q(s, a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
$Returns(s, a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):

Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability $> 0$
Generate an episode from $S_0, A_0$, following $\pi$: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$
Unless the pair $S_t, A_t$ appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

Append $G$ to $Returns(S_t, A_t)$
$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

# Monte Carlo ES

## Considerations

- All the returns for each state-action pair are accumulated and averaged, irrespective of what policy was in force when they were observed
- Monte Carlo ES cannot converge to any suboptimal policy
    - If it did, then the value function would eventually converge to the value function for that policy, and that in turn would cause the policy to change
- Stability is achieved only when both the policy and the value function are optimal
    - Convergence to this optimal fixed points seems inevitable as the changes to the action value function decreases over time, *BUT has not yet been formally proved*

# Monte Carlo Control without Exploring Starts

# Avoid exploring starts assumption

- The only general way to ensure that all actions are selected infinitely often is for the agent to continue to select them
- There are two approaches to ensuring this:
    - On-policy methods attempt to evaluate or improve the policy that is used to make decisions (e.g., Monte Carlo ES)
    - Off-policy methods evaluate or improve a policy different form that used to generate the data

# On-policy control methods

- In on-policy control methods the policy is generally *soft*
    - $\pi(a \mid s) > 0$ for all $s \in \mathcal{S}$ and all $a \in \mathcal{A}(s)$, but gradually shifted closer and closer to a deterministic optimal policy
- The overall idea of on-policy MC control is still that of GPI
    - GPI does not require that the policy be taken all the way to a greed policy, only that it be moved *toward* a greedy policy
        - We will move it only to an $\epsilon$-*greedy policy*
            - For any $\epsilon$-soft policy, $\pi$, $\epsilon$-*greedy policy with respect to* $q_\pi$ *is guaranteed to be better than or equal to* $\pi$

# On-policy first-visit MC control (for $\epsilon$-soft policies)

## Estimating $\pi \approx \pi_*$

Algorithm parameter: small $\varepsilon > 0$

Initialize:

$\pi \leftarrow$ an arbitrary $\varepsilon$-soft policy

$Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

$Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Repeat forever (for each episode):

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

$G \leftarrow 0$

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

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

Unless the pair $S_t, A_t$ appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

Append $G$ to $Returns(S_t, A_t)$

$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

$A^* \leftarrow \text{argmax}_a \, Q(S_t, a)$ (with ties broken arbitrarily)

For all $a \in \mathcal{A}(S_t)$:

$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(S_t)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(S_t)| & \text{if } a \neq A^* \end{cases}$$

# Policy iteration

## We skip the demonstration!

- Policy iteration works for $\epsilon$-soft policies.
    - Using the natural notion of greedy policy for $\epsilon$-soft policies, one is assured of improvement on every step, except when the best policy has been found among the $\epsilon$-soft policies

# Off-policy Prediction via Importance Sampling

# Dilemma

- All learning control methods seek to learn action values conditional on subsequent *optimal* behavior, but they need to behave non-optimally in order explore all actions (to *find* the optimal actions)

- How can they learn about the optimal policy while behaving according to an exploratory policy?

# Off-policy learning

- A straightforward approach is to use two polices:
    - one that is learned about and that becomes the optimal policy *(target policy)*
    - And one that is more exploratory and is used to generate behavior *(behavior policy)*
- In this case we say that learning is from data “off” the target policy, and overall process is termed *off-policy learning*

# An important consideration

- Off-policy methods require additional concepts and notation, and because the data is due to a different policy, off-policy methods are often of great variance and are slower to converge
- On the other hand, off-policy methods are *more powerful and general*
    - They include on-policy methods as the special case in which the target and behavior policies are the same

# Prediction problem

- Both target and behavior policies are fixed
    - Suppose we wish to estimate $v_\pi$ or $q_\pi$, but all we have are episodes following another policy *b,* where $b \neq \pi$
    - In this case, $\pi$ is the target policy, *b* is the behavior policy, and both polices are considered fixed and given

# Coverage assumption

- In order to use episodes from *b* to estimate values for $\pi$, we require that every action taken under $\pi$ is also taken, at least occasionally, under *b*
    - That is, we require that $\pi(a \mid s) > 0$ implies $b(a \mid s) > 0$

- It follows from coverage that *b* must be stochastic in states where it is not identical to $\pi$
- The target policy, $\pi$, may be deterministic

# Importance sampling

## Almost all off-policy methods utilize it

- It is a general technique for estimating expected values under one distribution given samples from another

- We apply importance sampling to off-policy learning by weighting returns according to the relative probability of their trajectories occurring under the target and behavior policies
    - called *importance-sampling ratio*

# Importance-sampling ratio

## Part 1

- Given a starting state $S_t$, the probability of the subsequent state-action trajectory, $A_t$, $S_{t+1}$, $A_{t+1}$, …, $S_T$, occurring under any policy $\pi$ is

$$\begin{aligned}
&\Pr\{A_t, S_{t+1}, A_{t+1}, \ldots, S_T \mid S_t, A_{t:T-1} \sim \pi\} \\
&\quad = \pi(A_t|S_t)p(S_{t+1}|S_t, A_t)\pi(A_{t+1}|S_{t+1}) \cdots p(S_T|S_{T-1}, A_{T-1}) \\
&\quad = \prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k),
\end{aligned}$$

State-transition probability function (defined by 3.4!)

# Importance-sampling ratio

## Part 2

- The relative probability of the trajectory under the target and behavior policies *(the importance-sampling ratio)* is

$$\rho_{t:T-1} \doteq \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k,A_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)p(S_{k+1}|S_k,A_k)} = \prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)} \qquad (5.3)$$

- Although the trajectory probabilities depend on the MDP's transition probabilities, which are generally unknown, they appear identically in both the numerator and denominator, and thus cancel
- The importance sampling ratio ends up depending only on the two policies and the sequence

# Expected value

- We wish to estimate the expected returns (values) under the target policy, but all we have are returns $G_t$ due to the behavior policy
- These returns have the wrong expectation $\mathbb{E}[G_T \mid S_t = s] = v_b(s)$ and so cannot be averaged to obtain $v_\pi$
    - *This is where importance sampling comes in*!
- The ration $\rho_{t:T-1}$ transforms the returns to have the right expected value:

$$\mathbb{E}[\rho_{t:T-1} G_t \mid S_t = s] = v_\pi(s) \qquad (5.4)$$

# Monte Carlo algorithm

## Every-visit method

- Now we are ready to give a Monte Carlo algorithm that averages returns from a batch of observed episodes following policy *b* to estimate $v_\pi(s)$
- It is convenient here to number time steps in a way that increases across episode boundaries:
    - e.g., if the first episode of the bench ends in a terminal state at time 100, then the next episode begins at time t=101
- This enables to use time-step numbers to refer to particular steps in particular episodes
- We can define the set of all time steps in which state *s* is visited, denoted $\mathcal{T}(s)$

# Monte Carlo algorithm

## First-visit method

- For a first-visit method, $\mathcal{T}(s)$ would only include time steps that were first visits to *s* within their episodes
- Also,
    - Let $T(t)$ denote the first time of termination following time *t*
    - Let $G_t$ denote the return after *t* up through $T(t)$
- Then $\{G_t\}_{t \in \mathcal{T}(s)}$ are the returns that pertain to state *s*, and
    - $\{\rho_{t:T(t)-1}\}_{t \in \mathcal{T}(s)}$ are the corresponding importance-sampling ratios

# Ordinary importance sampling

- To estimate $v_\pi(s)$, we simply scale the returns by the ratios and average the results

$$V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1} G_t}{|\mathcal{T}(s)|} \qquad (5.5)$$

# Weighted importance sampling

- Alternative which uses a *weighted* average

$$V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1} G_t}{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1}} \qquad (5.6)$$

- Or zero if the denominator is zero

# Difference between the two sampling

## First-visit methods

- The difference is expressed in their biases and variances
- Ordinary importance sampling is unbiased whereas weighted importance sampling is biased (though the bias converges asymptotically to zero)
- The variance of ordinary importance sampling is in general unbounded because the variance of the ratios can be unbounded, whereas in the weighted estimator the largest weight on any single return is one
  - In *practice*, the weighted estimator usually has dramatically lower variance and is strongly preferred

# Difference between the two sampling

## Every-visit methods

- They are both biased, though, again, the bias falls asymptotically to zero as the number of samples increase
- In practice, every-visit methods are often preferred because they remove the need to keep track of which states have been visited and because they are much easier to extend to approximations

# Incremental Implementation

# MC incremental implementation

- Monte Carlo prediction methods can be implemented incrementally, on an episode-by-episode basis
- Previously we averaged *rewards*, in Monte Carlo methods we average *returns!*
- For *off-policy* Monte Carlo methods, we need to separately consider
    - those that use *ordinary importance sampling*
    - those that use *weighted importance sampling*

# Ordinary importance sampling case

- The returns are scaled by the importance sampling ratio $\rho_{t:T(t)-1}$, then simply averaged as in (5.5)
- For these methods we can again use the incremental methods saw in Lesson 2, but using the scaled returns in place of the rewards

# Weighted importance sampling case

## Part 1

- Suppose we have a sequence of returns $G_1, G_2, \ldots, G_{n-1}$, all starting in the same state and each with a corresponding random weight $W_i$ (e.g., $W_i = \rho_{t_i:T(t_i)-1}$
- We wish to form the estimate

$$V_n \doteq \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, \qquad n \geq 2 \tag{5.7}$$

- And keep it up-to-date as we obtain a single additional return $G_n$

# Weighted importance sampling case

## Part 2

- In addition to keeping track of $V_n$, we must maintain for each state the cumulative sum $C_n$ of the weights given to the first *n* returns
- The update rule for $V_n$ is

$$V_{n+1} \doteq V_n + \frac{W_n}{C_n}\Big[G_n - V_n\Big], \qquad n \geq 1 \tag{5.8}$$

- And

$$C_{n+1} \doteq C_n + W_{n+1}$$

- Where $C_0 \doteq 0$ (and $V_1$ is arbitrary and thus need not be specified)

# Off-policy MC prediction for estimating $Q \approx q_\pi$

## Using weighed importance sampling

Input: an arbitrary target policy $\pi$
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
    $Q(s,a) \in \mathbb{R}$ (arbitrarily)
    $C(s,a) \leftarrow 0$

Loop forever (for each episode):
    $b \leftarrow$ any policy with coverage of $\pi$
    Generate an episode following $b$: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
    $G \leftarrow 0$
    $W \leftarrow 1$
    Loop for each step of episode, $t = T-1, T-2, \ldots, 0$, while $W \neq 0$:
        $G \leftarrow \gamma G + R_{t+1}$
        $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
        $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$
        $W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$

# Off-policy Monte Carlo Control

# Off-policy MC control

- Off-policy MC control methods follow the behavior policy while learning about and improving the target policy
- These techniques require that the behavior policy has a nonzero probability of selecting all actions that might be selected by the target policy (coverage)
    - To explore all possibilities, we require that the behavior policy be soft

# Off-policy MC control

## Based on GPI and weighted importance sampling, for estimating $\pi_*$ and $q_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
$Q(s,a) \in \mathbb{R}$ (arbitrarily)
$C(s,a) \leftarrow 0$
$\pi(s) \leftarrow \arg\max_a Q(s,a)$ (with ties broken consistently)

Loop forever (for each episode):
$b \leftarrow$ any soft policy
Generate an episode using $b$: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
$W \leftarrow 1$
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
$G \leftarrow \gamma G + R_{t+1}$
$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$
$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken consistently)
If $A_t \neq \pi(S_t)$ then exit inner Loop (proceed to next episode)
$W \leftarrow W \frac{1}{b(A_t|S_t)}$

# A problem!

- A potential problem is that this method learns only from the tails of episodes, when all of the remaining actions in the episode are greedy
- If nongreedy actions are common, then learning will be slow, particularly for states appearing in the early portions of long episodes
    - *Potentially, this could greatly slow learning*!
    - But there has been insufficient experience to assess how serious this problem is
- If it is serious, the most important way to address it is probably by incorporating *temporal-difference learning*
    - Alternatively, if $\gamma$ is less than 1 we can use *discounting-aware importance sampling (which we do not discuss)*

# Bibliography:

Reinforcement Learning An Introduction (Second Edition), R. S. Sutton & A. G. Barto