Monte Carlo Methods

Reinforcement learning – LM Artificial Iintelligence
(2022-23)

Alberto Castellini
University of Verona

Summary

- Introduction
- Monte Carlo Prediction
- Monte Carlo Estimation of Action Values
- Monte Carlo Control
- Monte Carlo Control without Exploring Starts
- Off-policy Prediction via Importance Sampling (hints)
- Incremental Implementation of Monte Carlo Prediction
- Off-Policy Monte Carlo Control

Introduction

Introduction

- Unlike in DP, here **we do not assume complete knowledge of the environment** (i.e., model of the dynamics p(s',r | s,a))

  → **Monte Carlo (MC)** methods are **model free RL methods**

  → First **learning methods** for estimating value function and discovering optimal policies

- Monte Carlo methods require only **experience** (sample **sequences of states, actions, and rewards** from **actual** or **simulated** interactions with the environment)

- **Learning from actual experience** is striking: it requires **no prior knowledge of the environment**
- **Learning from simulated experience** is also powerful: a model that generates sample transitions is required (**easier** than complete probability distributions over all possible transitions required in DP)

Introduction

- MC methods solve RL problems by **averaging sample returns** over episodes

- We **assume** experience is split in **episodes**.

- Values and policies are updated **after each episode** (**not after each step**, as in Temporal Difference methods, next lecture)

- MC methods adapt the idea of **general policy iteration (GPI)** defined in DP methods, however
  - **DP** methods require the **model** of the **dynamics**
  - **MC** methods **learn** the value function from **sample returns**

- Policy evaluation (**prediction**)
- Policy improvement
- Optimal policy approximation (**control**)

Monte Carlo Prediction

Monte Carlo Prediction (policy evaluation)

- Given a **policy**, we aim to compute its **value function**

- Recall: the **value of a state** is its expected return (expected cumulative future discounted reward)

- **Main idea of MC**: to average the returns observed after visits of the state

- Given a set of **episodes** obtained following π and passing through state *s*. Each occurrence of state *s* in an episode is called ***visit* of s**. *s* may be visited multiple times

- **First-visit MC method** estimates $v_\pi(s)$ as the average of the returns following the **first** visit to *s*

- **Every-visit MC method** averages the returns following **all** visits to *s*

Monte Carlo Prediction (policy evaluation)

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π 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 π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$

  $G \leftarrow 0$

  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 $S_0, S_1, \ldots, S_{t-1}$: // If S_t does not appears in S_0, S_1, …, S_{t-1}

      Append G to $Returns(S_t)$

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

Monte Carlo Prediction (policy evaluation)

- Both first-visit and every-visit MC **converge** to $v_\pi(s)$ as the number of visits (or first visits) to *s* goes to infinity.

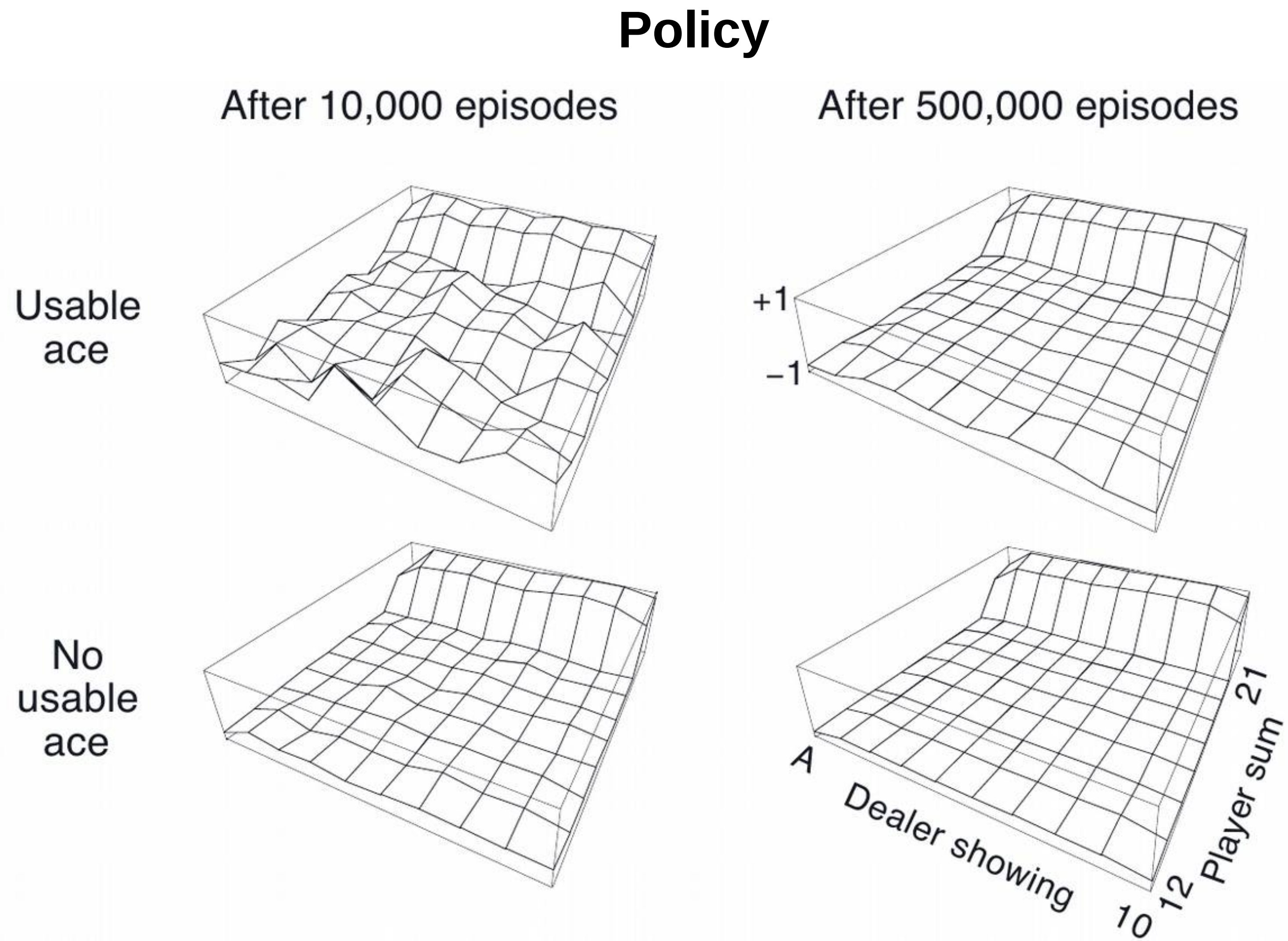
First-visit MC convergence (1940s)

- Each return is an independent, identically distributed estimate of $v_\pi(s)$ with finite variance
- Law of large numbers: the **sequence of averages converges** to the **expected value**
- Each average is an unbiased estimate
- The **standard deviation** of its **error** falls as $1/\sqrt{n}$, where *n* is the number of returns averaged.

Every-visit MC convergence (Singh and Sutton, 1996): the proof is less simple but **the estimate also converges quadratically to** $v_\pi(s)$

$$\lim_{n \to \infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^q} = \mu \qquad q=2$$

Example: Blackjack



Homework: read Example 5.5 in the Sutton and Barto book (page 93). Try to understand MDP elements (states, actions, transition model, reward function) in the blackjack domain. Try to answer questions of Exercize 5.1 (page 94).

Example: Blackjack

- Notice: although we have **complete knowledge of the environment** we **do not have the distribution *p* of next events** (i.e., model of the dynamics)
  - E.g., player's sum is 14, he sticks. What is the probability of terminating with a reward of +1 as a function of the dealer's showing card? **Very difficult to know.**
  - All these probabilities must be computed in advance when DP methods are used

    → It is **not easy to apply DP for blackjack**

    → In contrast, **generating the sample games required by MC methods is easy**

  - **This happens surprisingly often in practice and makes MC methods very useful**

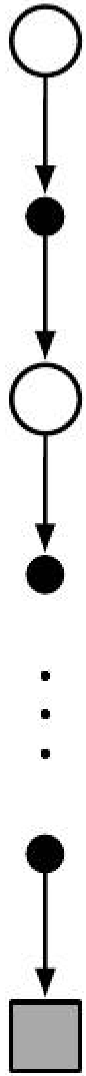
Extension of backup diagrams to MC methods

General idea of **backup** methods:

- **On top**: root node to be updated
- **Below**: all transitions and leaf nodes whose rewards and estimated values contribute to the update

MC estimation of v_π

- **Root**: state node
- **Below**: entire trajectory of transitions along a single episode ending at the terminal state



Important Observation

The **estimate for one state in MC** methods does **not** build upon the **estimate of any other state**, as is the case in DP

→ **MC methods do not perform bootstrapping**

In MC methods the **computational expense** of estimating the value of a **single state** is **independent on the number of states**

Useful for **online** **estimation** or estimation of **subsets of states**

MC Estimation of Action Values

MC Estimation of Action Values

- **Problem: Without a transition model state values are not sufficient to determine a policy** (which action should I select to reach the target state?)

- In MC methods **we must esplicitly estimate the value of each action** $q_\pi(s, a)$ to finally estimate q_*

- The **MC methods** are the **same** used for estimating state **values** but focused on **state-action pairs**.

- A **state-action pair *s,a* is visited** in an episode if the state *s* is visited and action *a* is taken in it

First-visit MC and **every-visit MC converge quadratically** to the **true values** (expected returns) as the number of visits to each state-action pair approaches **infinity**

MC Estimation of Action Values

- **Problem:** many state-variable pairs may never be visited.

- E.g., if the policy is **deterministic** one will observe returns only for **one of the actions** from each state → **estimates of the other actions will not improve with experience**

- This is a problem because the **purpose of learning action values** is to help **choosing** among the actions available in each state (in the policy **improvement** step)

- **We need to estimate values of all the actions from each state**
  → **Problem of maintaining exploration**
  → **We must assure continual exploration**

- **Solution:** specify that **episodes start in a state-action pair** and every pair has **non-zero probability** to be selected (**exploring starts**)

MC Estimation of Action Values

- **Problem:** Exploring starts **cannot be relied upon in general** (e.g., when learning from a real environment)

- Most common **alternative**: consider only **stochastic policies** with nonzero probability of selecting all actions in each state

  (e.g., ε -greedy policies)

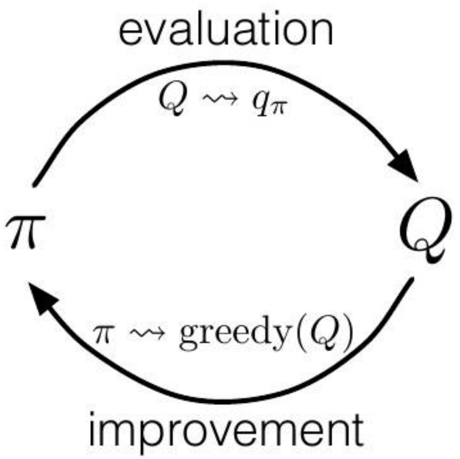
- In the following we will analyze **MC Control** (i.e., optimal policy approximation) first **with** and then **without** exploring starts.

Monte Carlo Control

Monte Carlo Control (i.e., MC-based GPI)

- **MC estimation** can be used **in control** (**control**=**optimal policy approximation**)

- **GPI approach**:
  - Maintain both **approximate policy** and **approximate value function**
  - Value function is altered to better approximate the value function of the current policy
  - The policy is improved w.r.t. the current value function



$$\pi_0 \xrightarrow{\mathrm{E}} q_{\pi_0} \xrightarrow{\mathrm{I}} \pi_1 \xrightarrow{\mathrm{E}} q_{\pi_1} \xrightarrow{\mathrm{I}} \pi_2 \xrightarrow{\mathrm{E}} \cdots \xrightarrow{\mathrm{I}} \pi_* \xrightarrow{\mathrm{E}} q_*$$

Monte Carlo Control (i.e., MC-based GPI)

- **Policy evaluation:** is performed using MC prediction (let's assume to observe an infinite number of episodes, hence we get the exact q_{π_k})

- **Policy improvement:** is done by making the **policy greedy** w.r.t. the current value function
  - We have an **action-value function** hence **no model is needed** to construct the greedy policy

$$\pi(s) \doteq \arg\max_a q(s,a)$$

- For the **policy improvement theorem** we have

$$\begin{aligned} q_{\pi_k}(s, \pi_{k+1}(s)) &= q_{\pi_k}(s, \arg\max_a q_{\pi_k}(s,a)) \\ &= \max_a q_{\pi_k}(s,a) \\ &\geq q_{\pi_k}(s, \pi_k(s)) \\ &= v_{\pi_k}(s). \end{aligned}$$

$$\Rightarrow v_{\pi_{k+1}}(s) \geq v_{\pi_k}(s)$$

Monte Carlo Control (i.e., MC-based GPI)

- Hence the policy is ensured to **improve** and to **converge** to the **optimal** policy (and value function)

MC methods can be used to find optimal policies given only sample episodes and no other knowledge of the environment

- **Problem: we made 2 unlikely assumptions:**
  - **A1:** Availability of **infinite number of episodes**

    → similar to DP. **Solution 1**: determine # iterations to guarantee theoretical bounds (expansive). **Solution 2**: reduce iterations in evaluation (it works in practice, e.g., value iteration)
  - **A2:** **Exploring starts** → removed later on

- In MC it is however natural to **alternate** between **evaluation** and **improvement** on an **episode-by-episode-basis**

Monte Carlo Control with Exploring Starts (MCES)

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

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):

$\boxed{\text{Choose } S_0 \in \mathcal{S},\ A_0 \in \mathcal{A}(S_0) \text{ randomly}}$ such that all pairs have probability > 0

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

$G \leftarrow 0$

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 $Returns(S_t, A_t)$

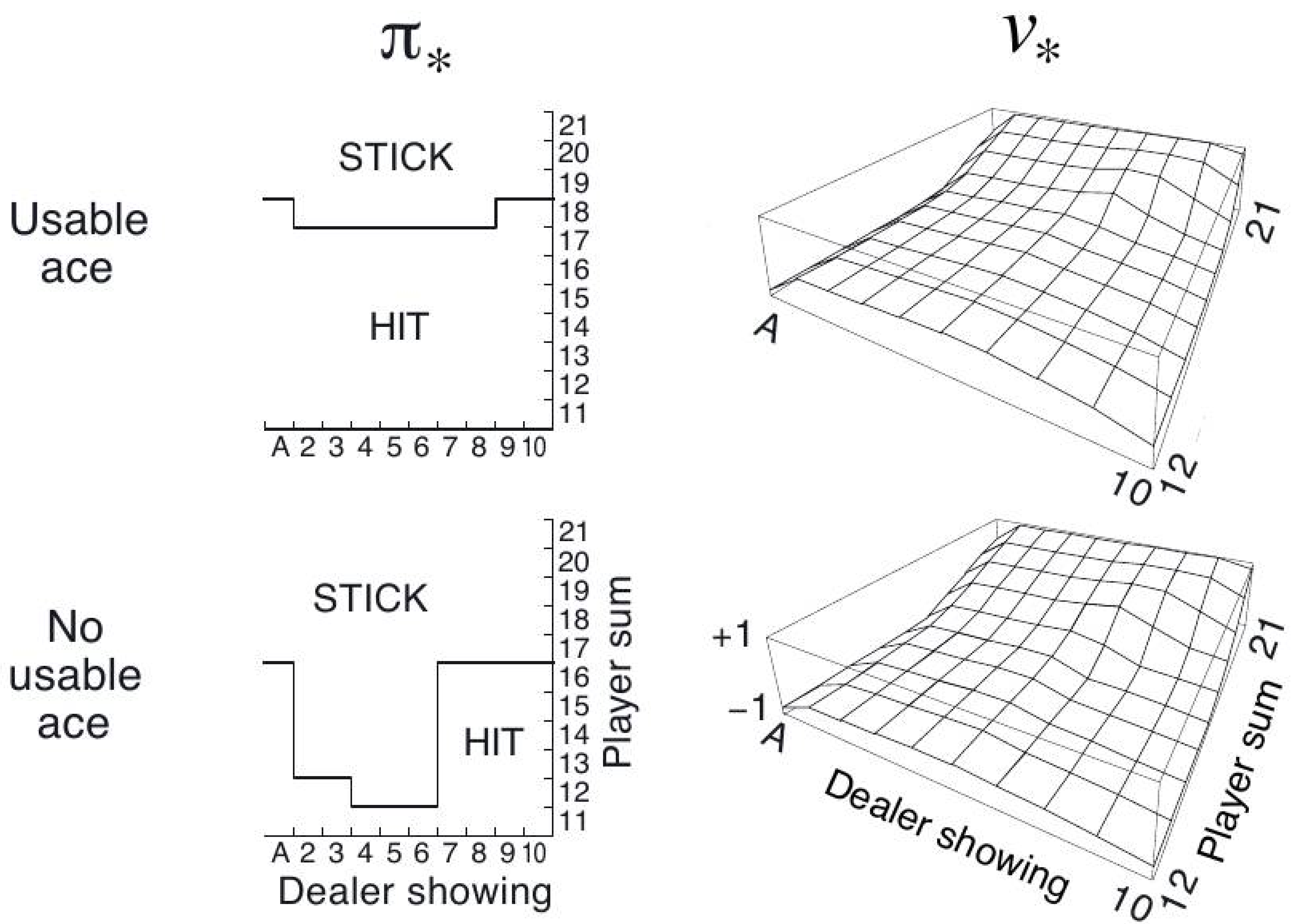
$\boxed{Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))}$

$\boxed{\pi(S_t) \leftarrow \text{argmax}_a Q(S_t, a)}$

Convergence: MC ES cannot converge to any suboptimal policy. If it did, than the value function would eventually converge to the value function of that policy, which would cause the policy to change.

Convergence seems inevitable but has not yet been formally proved.

Solving blackjack



Monte Carlo Control without Exploring Starts

Monte Carlo Control without Exploring Starts

- **How can we avoid the unlikely assumption of exploring starts?**

- Two approaches:
  - **On-policy methods:** evaluate or improve the policy that is used to make decisions (and produce data)
  - **Off-policy methods:** evaluate or improve a policy **different** from that used to generate the data

- **MC ES** is an example of an **on-policy** method
- An **alternative on-policy method** which does **not** use **exploring starts** is defined here. **Off-policy** methods will be defined afterwards

- In **on-policy** control methods the policy is in general **soft**

  i.e., $\pi(s,a)>0 \; \forall s\in S, a\in A$

  but gradually shifted closer and closer to deterministic optimal policies

Monte Carlo Control without Exploring Starts

- The on-policy method here presented uses ε **-greedy policies**, i.e.,

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

- They choose the action with maximal estimated action value most of the times but with probability ε they instead select an action **at random**.

- The methodology uses the **GPI** idea

- As in MC ES we use **first-visit MC** methods to estimate the action-value function for the current policy

- **GPI** does **not require** that the improved policy is **always greedy** but it requires only that it moves **towards** a **greedy** policy
- We move the policy toward an ε-greedy policy. For any ε -soft policy π , any ε -greedy policy w.r.t. q_π is guaranteed to be better than or equal to π

Monte Carlo Control without Exploring Starts

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Algorithm parameter: small $\boxed{\varepsilon > 0}$

Initialize:

$\boxed{\pi \leftarrow \text{an arbitrary } \varepsilon\text{-soft policy}}$

$Q(s,a) \in \mathbb{R}$ (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 π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

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 $Returns(S_t, A_t)$

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

$\boxed{A^* \leftarrow \arg\max_a Q(S_t, a)}$ (with ties broken arbitrarily)

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

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

Monte Carlo Control without Exploring Starts

- **Convergence:** the **policy improvement theorem** assures that any ε -greedy policy w.r.t. q_π is an improvement over any ε -soft policy. Let π' be the ε-greedy policy, for each state *s:*

$$
\begin{aligned}
\boxed{q_\pi(s,\pi'(s))} &= \sum_a \pi'(a|s) q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \;+\; (1-\varepsilon) \max_a q_\pi(s,a) \\
&\boxed{\geq} \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \;+\; (1-\varepsilon) \sum_a \frac{\pi(a|s) - \frac{\varepsilon}{|\mathcal{A}(s)|}}{1-\varepsilon} q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \;-\; \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \;+\; \sum_a \pi(a|s) q_\pi(s,a) \\
&\boxed{= v_\pi(s).}
\end{aligned}
$$

- Thus $\boxed{\pi' \geq \pi}$. The equality can hold only when both policies are optimal among the ε -soft policies (proof in the SutBar, Sec 5.4).

Off-policy Prediction via Importance Sampling (hints)

Off-policy Prediction via Importance Sampling

- **Dilemma of learning control methods:** they seek to learn action values conditional on subsequent optimal behaviour, but they need to behave non-optimally to explore all actions and find optimal ones

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

- The **on-policy** approach is a compromise. It learns action values not for the optimal policy but for a **near-optimal** policy (i.e., ε-greedy) that still explores

- **Solution:** use two policies
  - **Target policy:** learned policy, it becomes the optimal policy
  - **Behavior policy:** exploratory policy, it is used to generate data

- Learning is from data “off” the target policy → **Off-policy learning**

Off-policy Prediction via Importance Sampling

- We will consider both on-policy and off-policy methods

- **On-policy** methods are **simpler** and considered first

- **Off-policy** methods require additional concepts, they are often of **greater variance and slower to converge** but also **more powerful and general**

  → They include **on-policy** methods as a **special case** (target=behavior)

  → Additional uses in applications, e.g., **learning from** data generated by **non-learning controllers** or **human experts**

Off-policy Prediction via Importance Sampling

- Almost all **off-policy** methods utilize **importance sampling**, a general technique for **estimating expected values** under one **distribution given samples from another**

- **Idea:** we **weight returns** according to the **relative probability** of their trajectories occuring under target and behavior policies

- Given a starting state S_t, the **probability of the subsequent trajectory** occurring under any policy π is

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

where *p* is the state-transition probability function

Off-policy Prediction via Importance Sampling

- **Importance sampling ratio**: the relative probability of the trajectory under the target and behavior policies is

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

- **The ratio depends ony on the two policies and the sequence, not on the MDP (i.e., transition model)**

- Goal: We want to **estimate** expected returns (**values**) under the **target** policy but we have **returns** G_t due to the **behavior** policy

- Problem: These returns have the wrong expectation $\mathbb{E}[G_t|S_t=s] = v_b(s)$ hence they cannot be averaged to obtain v_π

- The importance-sampling ratio transforms the return:

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

Summary

- **MC** methods learn from **sample** episodes

- Four **advantages over DP** methods

  1) **no model of the environment** is required

  2) they can be used with **simulators** of the environment

  3) they can focus on **subset of states** (scaling)

  4) they do **not bootstrap**, hence they may be less harmed by violation of the Markov property

- **Problem** of maintaining sufficient exploration:
  - **Exploring starts**: ok only for simulated episodes
  - **On-policy prediction/control**: not completely precise
  - **Off-policy prediction/control**: the best method but more complex
    - Target/Behavior policy
    - Ordinary/weighted Importance Sampling

References

- R. S. Sutton, A. G. Barto. Reinforcement learning, An Introduction. Second edition. Chapter 5