"Where did you go to, if I may ask?" said Thorin to Gandalf as they rode along.

"To look ahead," said he.

"And what brought you back in the nick of time?"

"Looking behind," *said he.*

J.R.R. Tolkien, *The Hobbit*

CMPUT 365
Introduction to RL

Marlos C. Machado

Class 15/35

Reminder I

You **should be enrolled in the private session** we created in Coursera for CMPUT 365.

I **cannot** use marks from the public repository for your course marks.

You **need** to **check**, **every time**, if you are in the private session and if you are submitting quizzes and assignments to the private section.

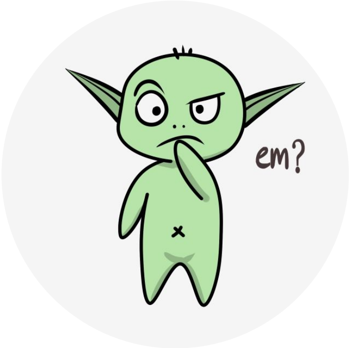
There were **12 pending invitations** last time I checked.

If you have any questions or concerns, **talk with the TAs** or email us `cmput365@ualberta.ca`.

Plan / Reminder II

- The time of my office hours has changed.
  - Thursday 10:00am - 12:00pm in ATH 3-08.
- On the midterm:
  - I plan on marking it next week, worst case scenario next next week you should have your marks.
- What **I** plan to do today:
  - Where are we?
  - Overview of Monte Carlo Methods for Prediction & Control (Chapter 5 of the textbook).
- What I recommend **YOU** to do for next class:
  - Read Chapter 5 up to Section 5.5.
  - Graded Quiz (Off-policy Monte Carlo).
  - Programming Assignment is not graded this week.

Please, interrupt me at any time!



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Interlude

An overview

- Main features of a reinforcement learning problem:
  - Trial-and-error learning
  - Exploration
  - Delayed credit assignment

An overview

- Main features of a reinforcement learning problem:
  - <u>Trial-and-error learning</u>
  - <u>Exploration</u>
  - Delayed credit assignment

A flavour of RL: Bandits (Chapter 2)

An overview

- Main features of a reinforcement learning problem:
  - Trial-and-error learning
  - Exploration
  - Delayed credit assignment

But what does that mean?
What is this sequential decision-making
problem we are trying to solve?
What does solution mean here?

A problem formulation: MDPs (Chapter 3)

An overview

- Main features of a reinforcement learning problem:
  - Trial-and-error learning
  - Exploration
  - Delayed credit assignment

- What about the solution?

A first solution: Dynamic Programming (Chapter 4)

An overview

- Main features of a reinforcement learning problem:
  - Trial-and-error learning
  - Exploration
  - Delayed credit assignment

- What about the solution?
  - Dynamic programming! ← We need to know $p(s', r \mid s, a)$ and it can be computationally expensive to solve the system of linear equations.

Our first learning algorithm: Monte Carlo Methods (Chapter 5)

Chapter 5

Monte Carlo Methods

Monte Carlo Methods – Why?

- This is our **first learning** method.
- We do not assume complete knowledge of the environment.
- “Monte Carlo methods **require only experience** — sample sequences of states, actions, and rewards from actual or simulated interaction with an environment.” 🤯
- It works! And different variations are used everywhere in the field (n-step returns, TD(λ), MCTS–AlphaGo/AlphaZero–, etc).
- … but we still need a model, albeit only a sample model.

MC Methods are ways of solving the RL problem based on avg. sample returns (similar to bandits, but instead of rewards we are sampling returns).

Monte Carlo Prediction

$\boxed{\text{First-visit}}$ MC prediction, for estimating $V \approx v_\pi$

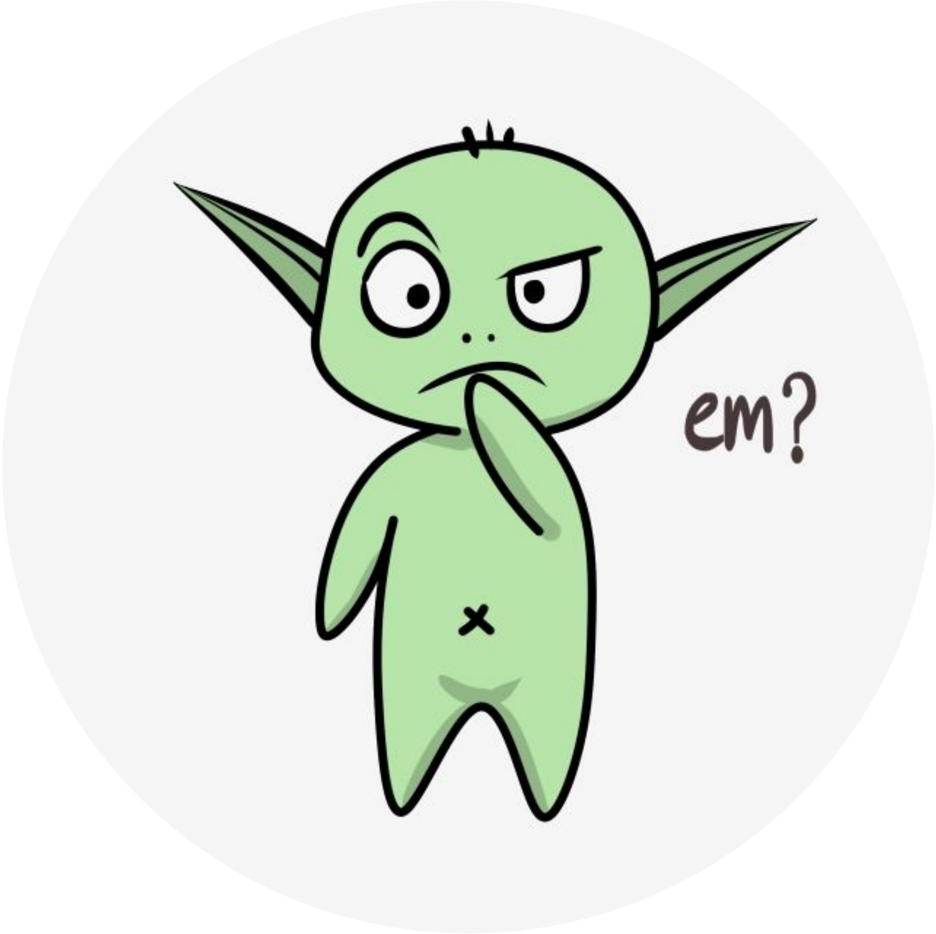
Input: a policy π to be evaluated

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):

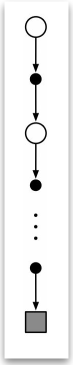
  Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:
      Append G to $Returns(S_t)$
      $V(S_t) \leftarrow \text{average}(Returns(S_t))$



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Some useful information / reminders about MC Methods

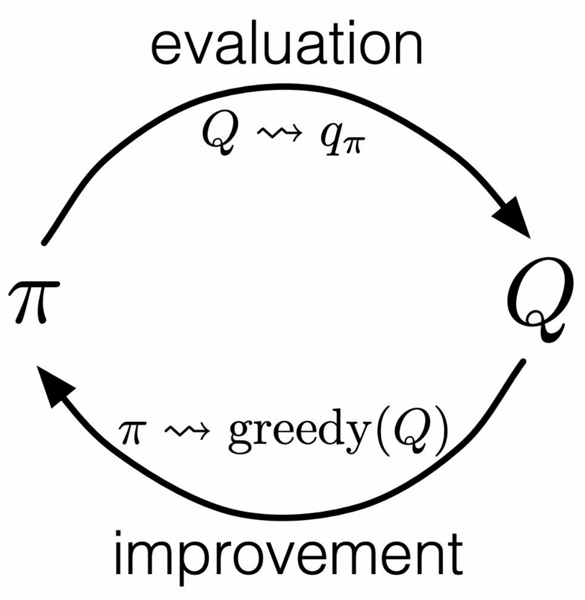
- Often it is much easier to get samples than to get the distribution of next events. Recall the Blackjack example in the textbook.
- Monte Carlo methods do not *bootstrap* (the estimate for one state does not build upon the estimate of any other state).
- First/every-visit MC converge to $v_\pi(s)$ as the number of visits to s goes to infinity. In first-visit MC, each return is i.i.d. and has finite variance ¯_(ツ)_/¯
- The computational cost of estimating the value of a single state is independent of the number of states.



Monte Carlo Estimation of Action Values

- If we don't have access to a model, we need to estimate *action* values.
- Same as before, but now we visit state-action pairs ¯_(ツ)_/¯
  But to estimate q_* we need to estimate the value of *all* actions from each state.
  Solution? Exploration! … or exploring starts 😒

Monte Carlo Control



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

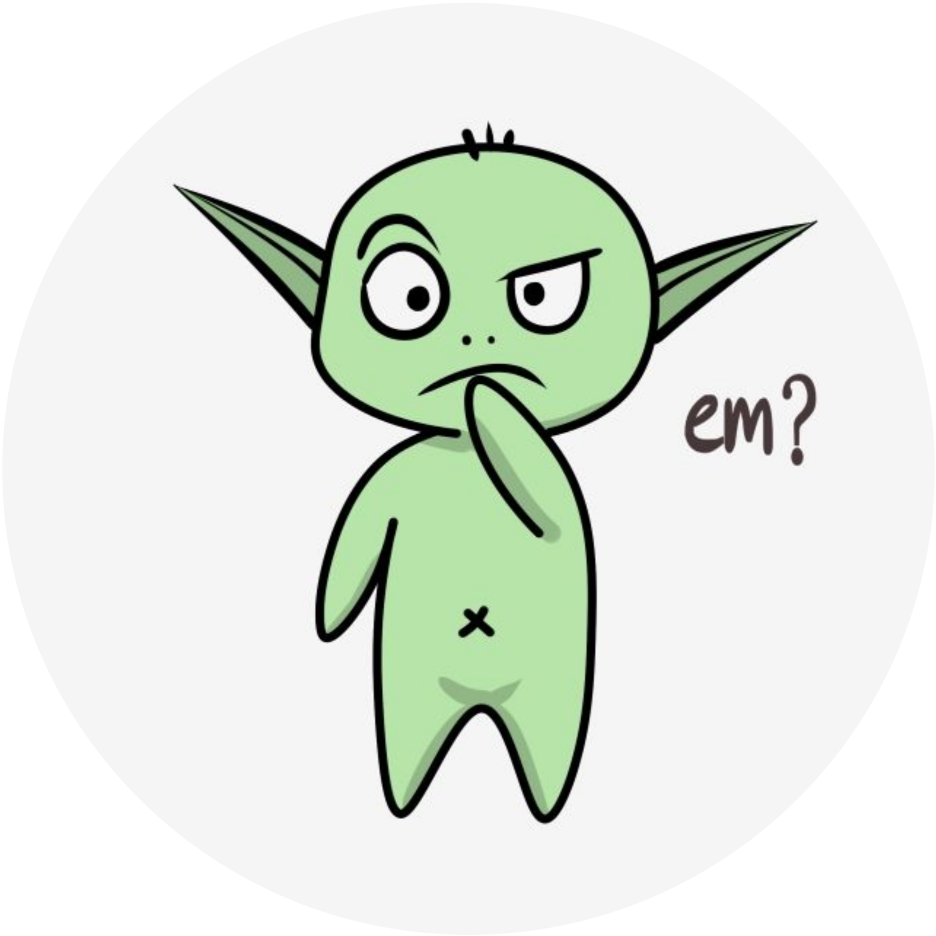
$$\pi(s) \doteq \arg\max_a q(s,a)$$

Monte Carlo ES

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



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

MC Control without Exploring Starts

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

Algorithm parameter: small $\varepsilon > 0$

Initialize:

$\quad \pi \leftarrow$ an arbitrary ε-soft policy

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

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

Repeat forever (for each episode):

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

$\quad G \leftarrow 0$

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

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

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

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

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

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

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

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

We need to ensure that the probability we select each action is not zero.

MC Control without Exploring Starts

On-policy: You learn about the policy you used to make decisions.

Off-policy: You learn about a policy that is different from the one you used to make decisions.

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

Algorithm parameter: small $\varepsilon > 0$

Initialize:

$\quad \pi \leftarrow$ an arbitrary ε-soft policy

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

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

Repeat forever (for each episode):

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

$\quad G \leftarrow 0$

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

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

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

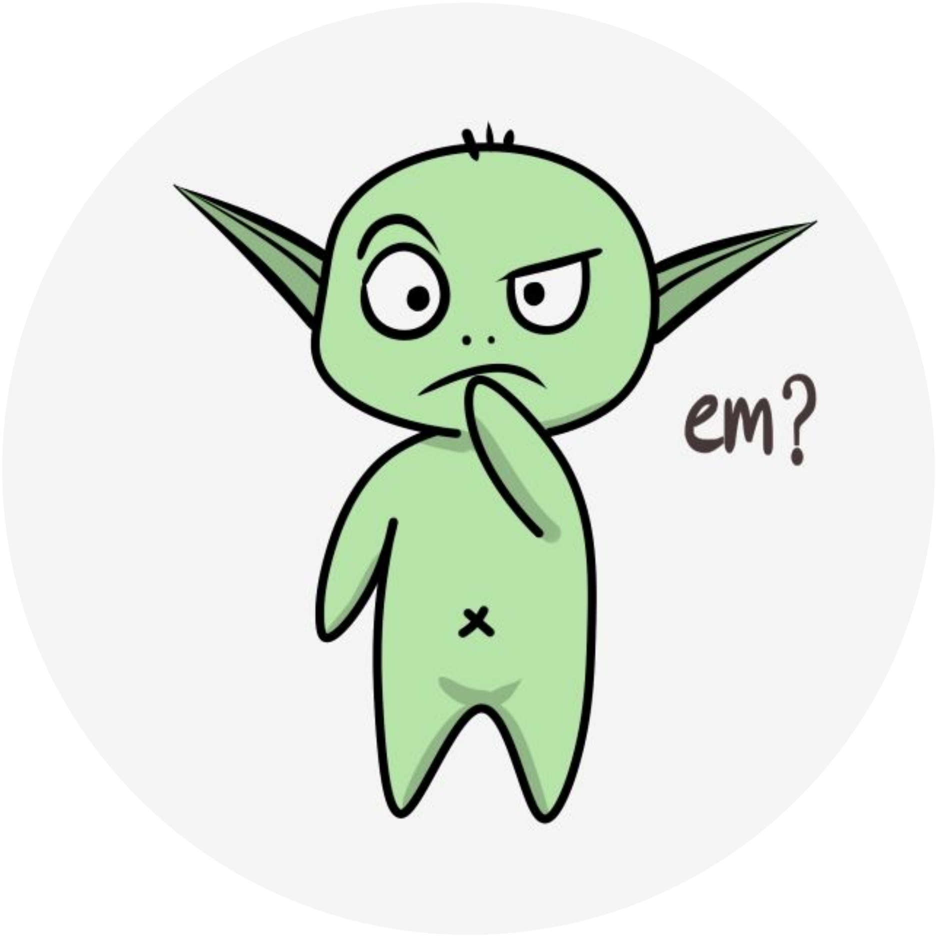
$\quad\quad\quad$ Append G to $Returns(S_t, A_t)$

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

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

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

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



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Learning with exploration

- We stopped after *On-policy first-visit MC control (for ε-soft policies).*
- … but how can we learn about the optimal policy while behaving according to an exploratory policy? We need to behave non-optimally in order to explore 🤔.
- So far we have been *on-policy*, which is a compromise: we learn about a near-optimal policy, not the optimal one.
- But what if we had two policies? We use one for exploration but we learn about another one, which would be the optimal policy?

Behaviour policy

Target policy

That's off-policy learning!

Pros and cons of off-policy learning

Pros

- It is more general.
- It is more powerful.
- It can benefit from external data
  - and other additional use cases.

Cons

- It is more complicated.
- It has much more variance.
  - Thus it can be much slower to learn.
- It can be unstable.

Check Example 5.5 in the textbook about Infinite Variance

What's the actual issue?

Let π denote the target policy, and let b denote the behaviour policy.

We want to estimate $\mathbb{E}_{\pi}[G_t]$, but what we can actually directly estimate is $\mathbb{E}_{b}[G_t]$.

In other words, $\mathbb{E}[G_t \mid S_t = s] = v_{b}(s)$.

Importance Sampling

A general technique for estimating expected values under one distribution given samples from another. It is based on re-weighting the probabilities of an event.

Importance Sampling

In RL, the probability of a trajectory is:

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

Importance Sampling

In RL, the probability of a trajectory is:

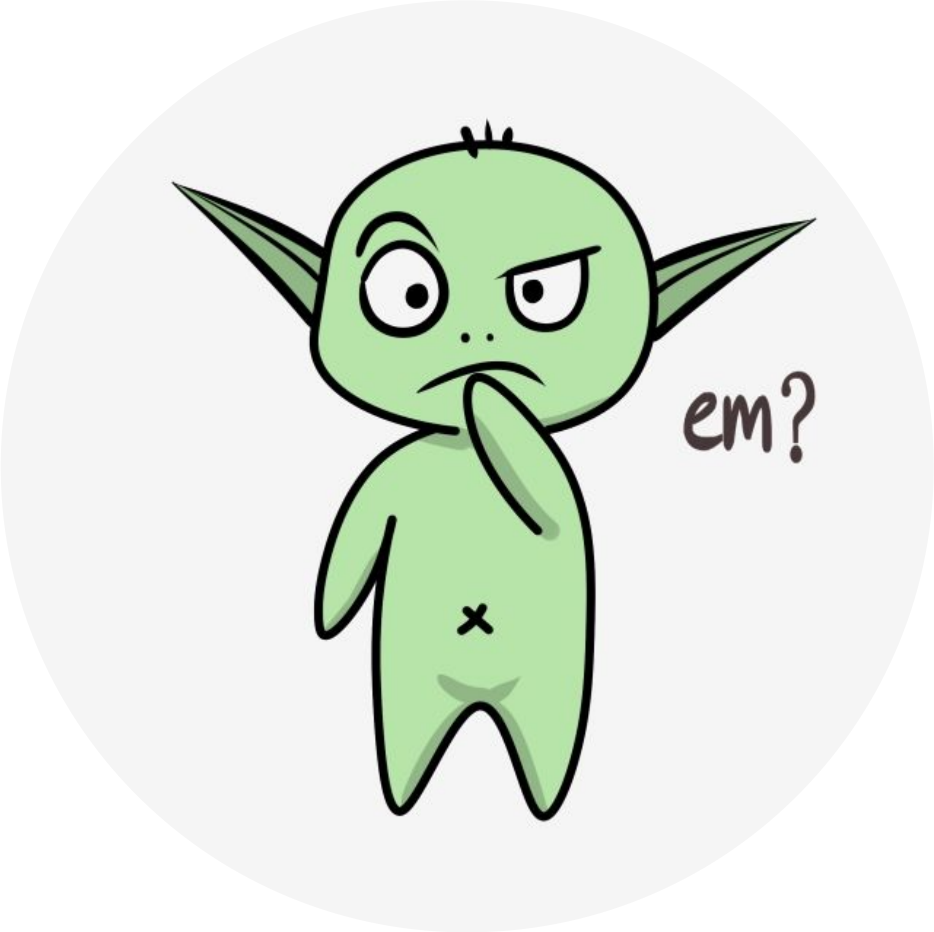
$$\begin{aligned}
&\Pr\{A_t, S_{t+1}, A_{t+1}, \ldots, S_T \mid S_t, A_{t:T-1} \sim \pi\} \\
&\quad = \pi(A_t|S_t)p(S_{t+1}|S_t, A_t)\pi(A_{t+1}|S_{t+1}) \cdots p(S_T|S_{T-1}, A_{T-1}) \\
&\quad = \prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k),
\end{aligned}$$

the relative prob. of the traj. under the target and behavior policies (the IS ratio) is:

We require *coverage*:
$b(a|s) > 0$ when $\pi(a|s) > 0$

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

The IS ratio does not depend on the MDP, that is, on $p(s', r \mid s, a)$!



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

The solution

The ratio $\rho_{t:T-1}$ transforms the returns to have the right expected value:

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

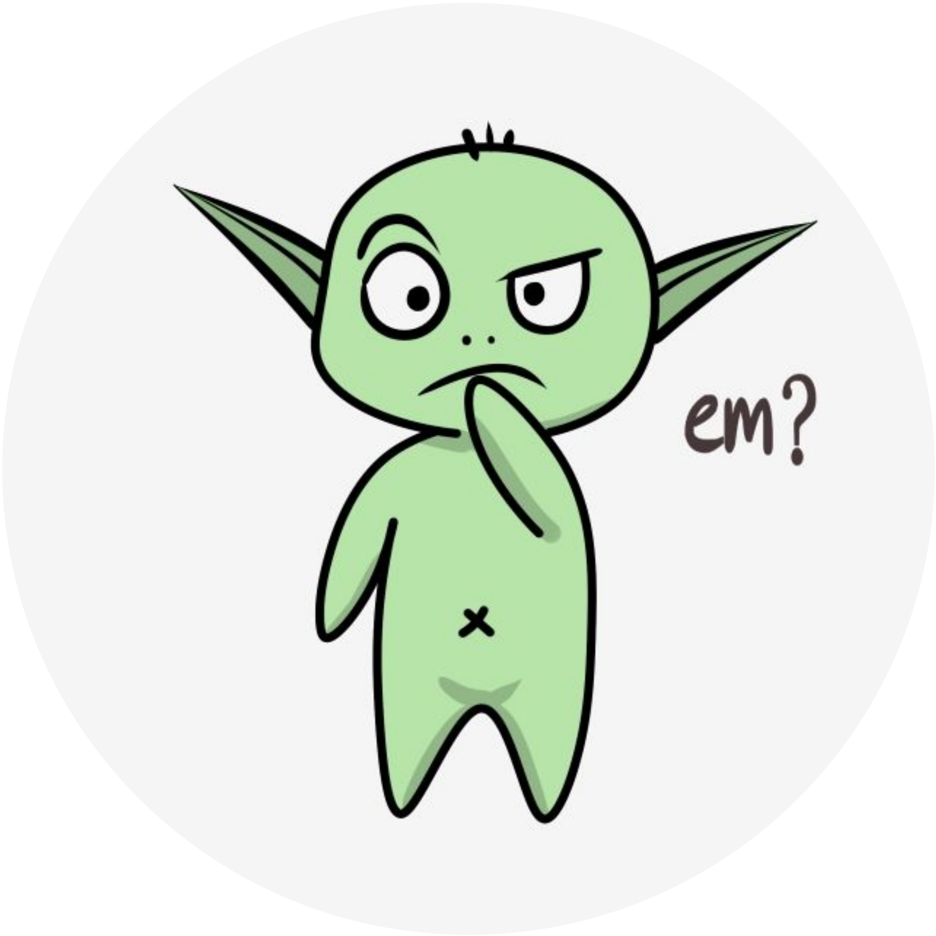
Ordinary importance sampling:

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

Set of all time steps in which state s is visited.

Weighted importance sampling:

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



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht