“The rotten tree-trunk, until the very moment when the storm-blast breaks it in two, has all the appearance of might it ever had.”

Isaac Asimov, *Foundation*

CMPUT 365
Introduction to RL

Marlos C. Machado

Classes 7-9/35

Plan

- Value Functions and Bellman Equations
  - A roadmap to the course
  - Content overview
    - We are still not talking about solution methods, we are only formalizing things

Reminder

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

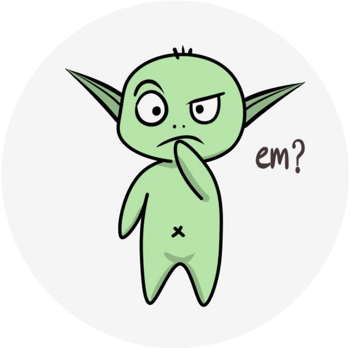
I **cannot** use marks from the public repository for your course marks.

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.

Some students who are enrolled in Coursera **haven't submitted any quizzes or assignments** in the private session, and that's all I can see.

The deadlines in the public session **do not align** with the deadlines in Coursera.

Please, interrupt me at any time!



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

Why? Where are we?! We need a roadmap

- Reinforcement learning is about solving sequential decision-making problems from interactions with the environment
    - Key features:
        - Trial-and-error
        - Exploration-exploitation trade-off
        - Delayed credit-assignment

Why? Where are we?! We need a roadmap

- Reinforcement learning is about solving sequential decision-making problems from interactions with the environment
  - Key features:
    - Trial-and-error
    - Exploration-exploitation trade-off
    - Delayed credit-assignment
- That's too abstract! Can we be more concrete and start from a simple example?

Why? Where are we?! We need a roadmap

- Reinforcement learning is about solving sequential decision-making problems from interactions with the environment
  - Key features:
    - Trial-and-error
    - Exploration-exploitation trade-off
    - Delayed credit-assignment
- That's too abstract! Can we be more concrete and start from a simple example?
  - Yes! Bandits

Chapter 2 of the textbook
Week 1 of *Fundamentals of RL*

Why? Where are we?! We need a roadmap

- Reinforcement learning is about solving sequential decision-making problems from interactions with the environment
  - Key features:
    - Trial-and-error
    - Exploration-exploitation trade-off
    - Delayed credit-assignment
- That's too abstract! Can we be more concrete and start from a simple example?
  - Yes! Bandits
- What if actions have consequences? What's a sequential decision-making problem? What does "solving" a sequential decision-making problem means?

Why? Where are we?! We need a roadmap

- Reinforcement learning is about solving sequential decision-making problems from interactions with the environment
  - Key features:
    - Trial-and-error
    - Exploration-exploitation trade-off
    - Delayed credit-assignment
- That's too abstract! Can we be more concrete and start from a simple example?
  - Yes! Bandits
- What if actions have consequences? What's a sequential decision-making problem? What does "solving" a sequential decision-making problem means?
  - We need a formal language for that: MDPs

Chapter 3 of the textbook
Weeks 2 & 3 of *Fundamentals of RL*

Why? Where are we?! We need a roadmap

- How can we do that?

Why? Where are we?! We need a roadmap

- How can we do that?
  - We can leverage Bellman equations and do Dynamic Programming

Chapter 4 of the textbook
Week 4 of *Fundamentals of RL*

Why? Where are we?! We need a roadmap

- How can we do that?
  - We can leverage Bellman equations and do Dynamic Programming
- But what if you don’t know how the world works (you don’t know $p(s', r \mid s, a)$)?

Why? Where are we?! We need a roadmap

- How can we do that?
  - We can leverage Bellman equations and do Dynamic Programming
- But what if you don't know how the world works (you don't know $p(s', r \mid s, a)$)?
  - Well, we can use Monte Carlo methods

Chapter 5 of the textbook
Week 2 of *Sample-based Learning Methods*

Why? Where are we?! We need a roadmap

- How can we do that?
  - We can leverage Bellman equations and do Dynamic Programming
- But what if you don't know how the world works (you don't know $p(s', r \mid s, a)$)?
  - Well, we can use Monte Carlo methods
- Do we really need to wait until episodes are over to learn something? What about continuing tasks?

Why? Where are we?! We need a roadmap

- How can we do that?
  - We can leverage Bellman equations and do Dynamic Programming
- But what if you don't know how the world works (you don't know p(s', r | s, a)?
  - Well, we can use Monte Carlo methods
- Do we really need to wait until episodes are over to learn something? What about continuing tasks?
  - Nope! Temporal-difference learning

Chapter 6 of the textbook
Weeks 3 & 4 of *Sample-based Learning Methods*

Why? Where are we?! We need a roadmap

- How can we do that?
  - We can leverage Bellman equations and do Dynamic Programming
- But what if you don't know how the world works (you don't know $p(s', r \mid s, a)$?
  - Well, we can use Monte Carlo methods
- Do we really need to wait until episodes are over to learn something? What about continuing tasks?
  - Nope! Temporal-difference learning
- Can't we learn more efficiently? Can we only learn from interactions with the environment?

Why? Where are we?! We need a roadmap

- How can we do that?
  - We can leverage Bellman equations and do Dynamic Programming
- But what if you don't know how the world works (you don't know $p(s', r \mid s, a)$?
  - Well, we can use Monte Carlo methods
- Do we really need to wait until episodes are over to learn something? What about continuing tasks?
  - Nope! Temporal-difference learning
- Can't we learn more efficiently? Can we only learn from interactions with the environment?
  - We can be more efficient, we can do planning alongside learning

Chapter 8 of the textbook
Week 5 of *Sample-based Learning Methods*

Why? Where are we?! We need a roadmap

- But what if we have many (maybe infinite) states? This doesn't scale!

Why? Where are we?! We need a roadmap

- But what if we have many (maybe infinite) states? This doesn't scale!
  - We then do function approximation

Chapters 9 & 10 of the textbook
Weeks 1, 2, & 3 of *Prediction and Control with Function Approximation*

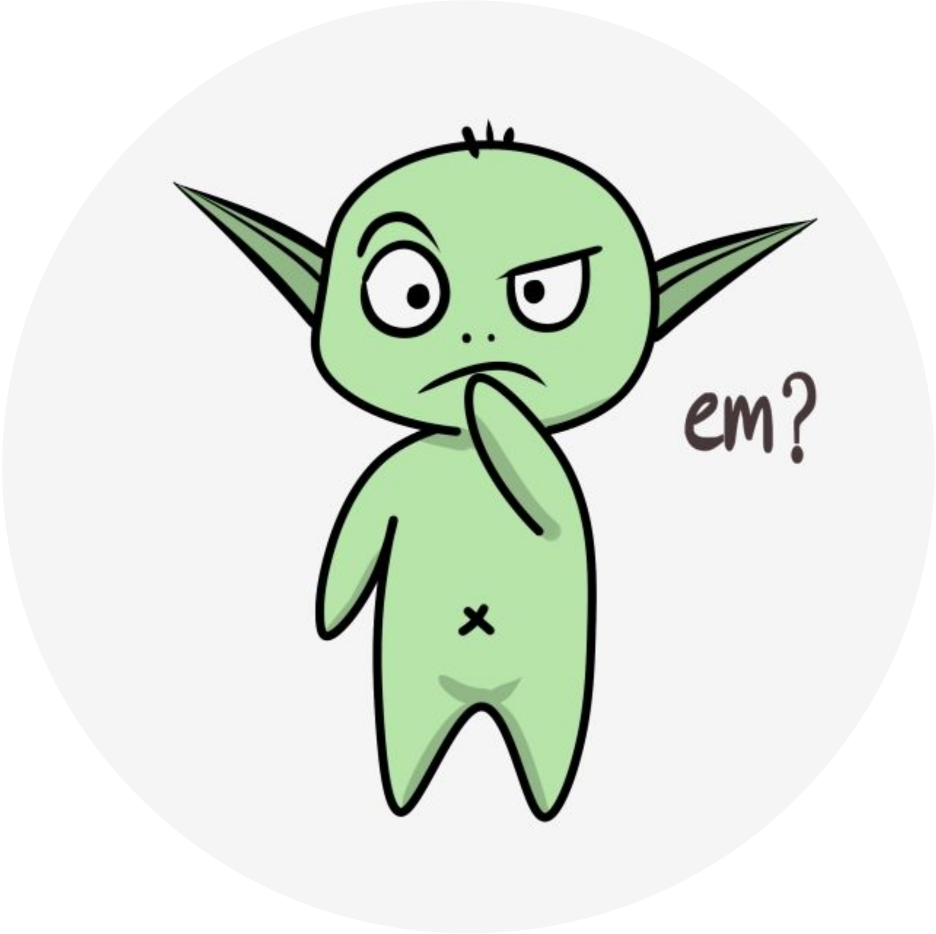
Why? Where are we?! We need a roadmap

- But what if we have many (maybe infinite) states? This doesn't scale!
  - We then do function approximation
- What about many (maybe infinite) actions? And stochastic policies?

Why? Where are we?! We need a roadmap

- But what if we have many (maybe infinite) states? This doesn't scale!
  - We then do function approximation
- What about many (maybe infinite) actions? And stochastic policies?
  - A way to tackle this problem is with policy gradient methods

Chapter 13 of the textbook
Week 4 of *Prediction and Control with Function Approximation*



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

Value Functions and Policies

- *Value functions are "functions of states (or state-action pairs) that estimate how good it is for the agent to be in a given state".*
- "How good" means expected return.
- Expected returns depend on how the agent behaves, that is, its *policy*.

Policy

- A policy is a mapping from states to probabilities of selecting each possible action:

$$\pi : \mathcal{S} \to \Delta(\mathcal{A})$$

in other words, $\pi(a|s)$ is the probability that $A_t = a$ if $S_t = s$.

Exercise 3.11 If the current state is S_t, and actions are selected according to a stochastic policy π, then what is the expectation of R_{t+1} in terms of π and the four-argument function p (3.2)? □

Value Function

- The value function of a state s under a policy π, denoted $v_\pi(s)$ is the expected return when starting in s and following π thereafter.

state-value function for policy π

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right]$$

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s, A_t = a\right]$$

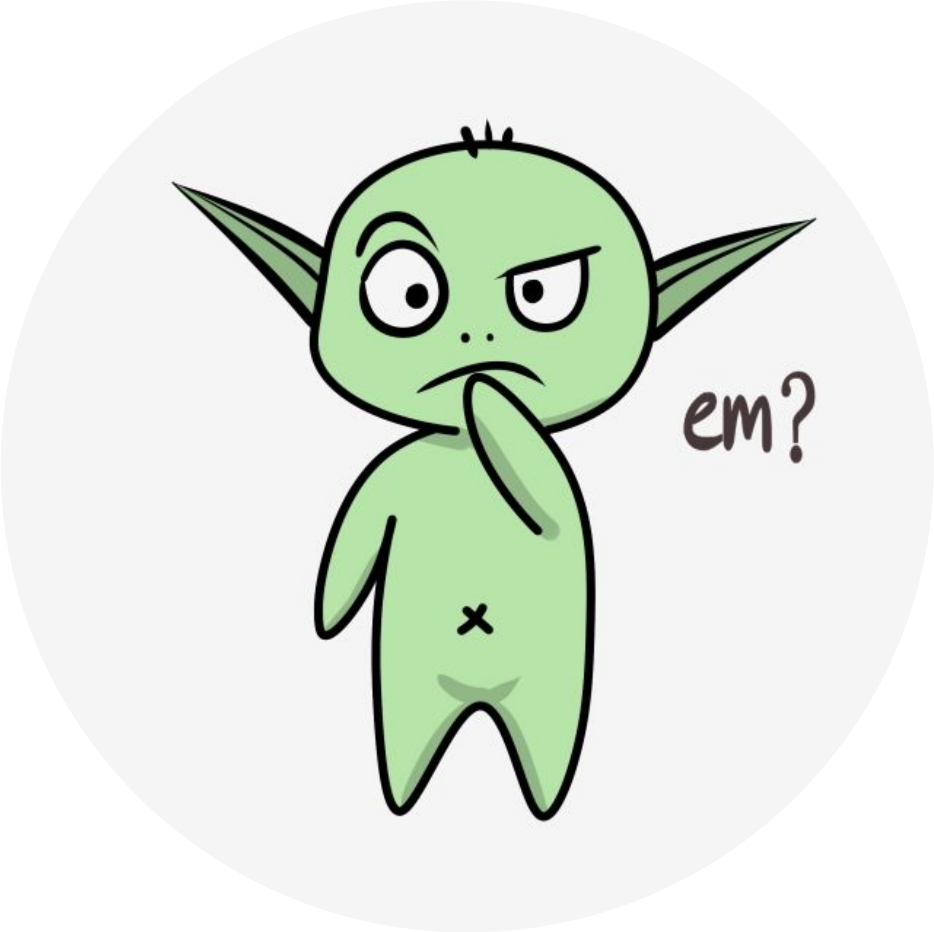
action-value function for policy π

Why is this difference important?

Exercises from the Textbook

Exercise 3.12 Give an equation for v_π in terms of q_π and π. □

Exercise 3.13 Give an equation for q_π in terms of v_π and the four-argument p. □



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

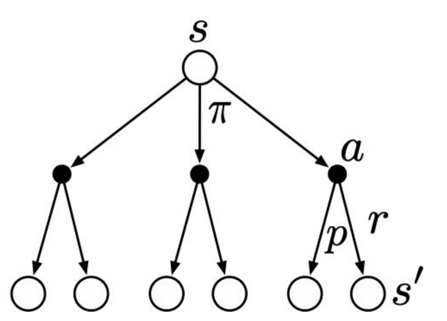
Value Functions Satisfy Recursive Relationships

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s]$$

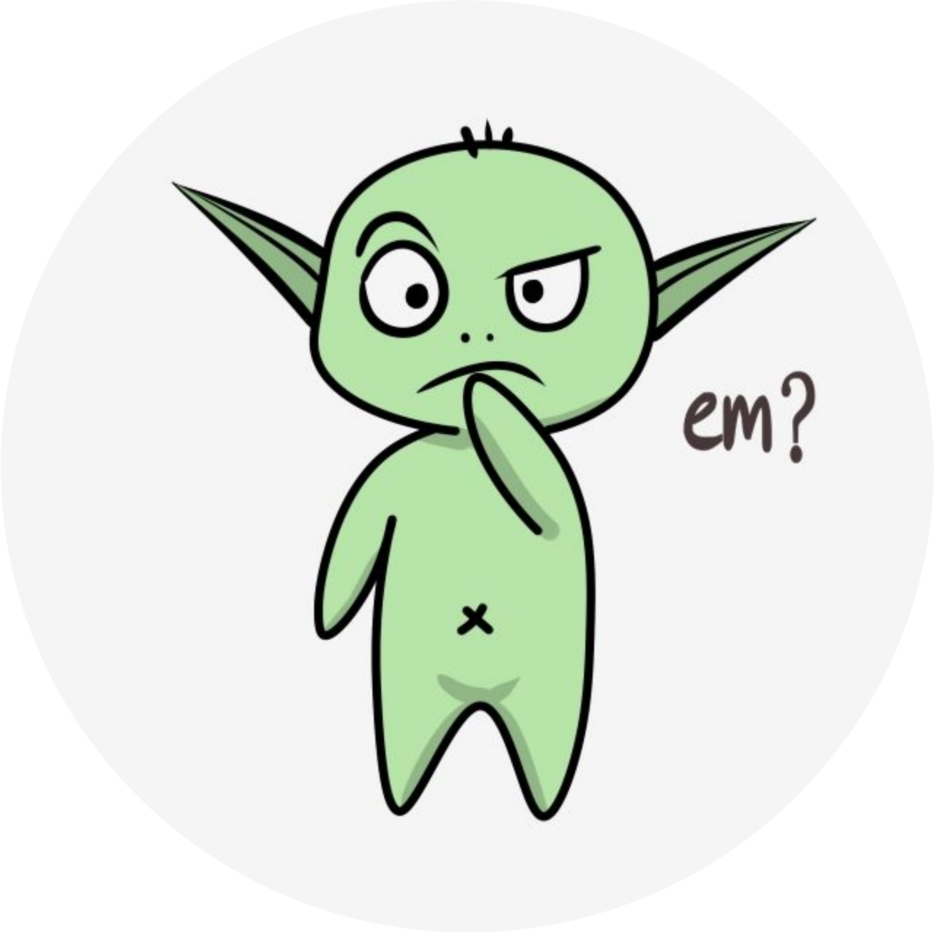
Value Functions Satisfy Recursive Relationships

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r | s, a) \Big[r + \gamma \mathbb{E}_\pi[G_{t+1} | S_{t+1} = s'] \Big] \\
&= \sum_a \pi(a|s) \sum_{s', r} p(s', r | s, a) \Big[r + \gamma v_\pi(s') \Big]
\end{aligned}
$$

This is a system of linear equations!



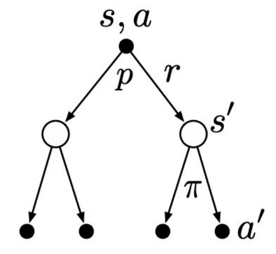
Backup diagram for v_π



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

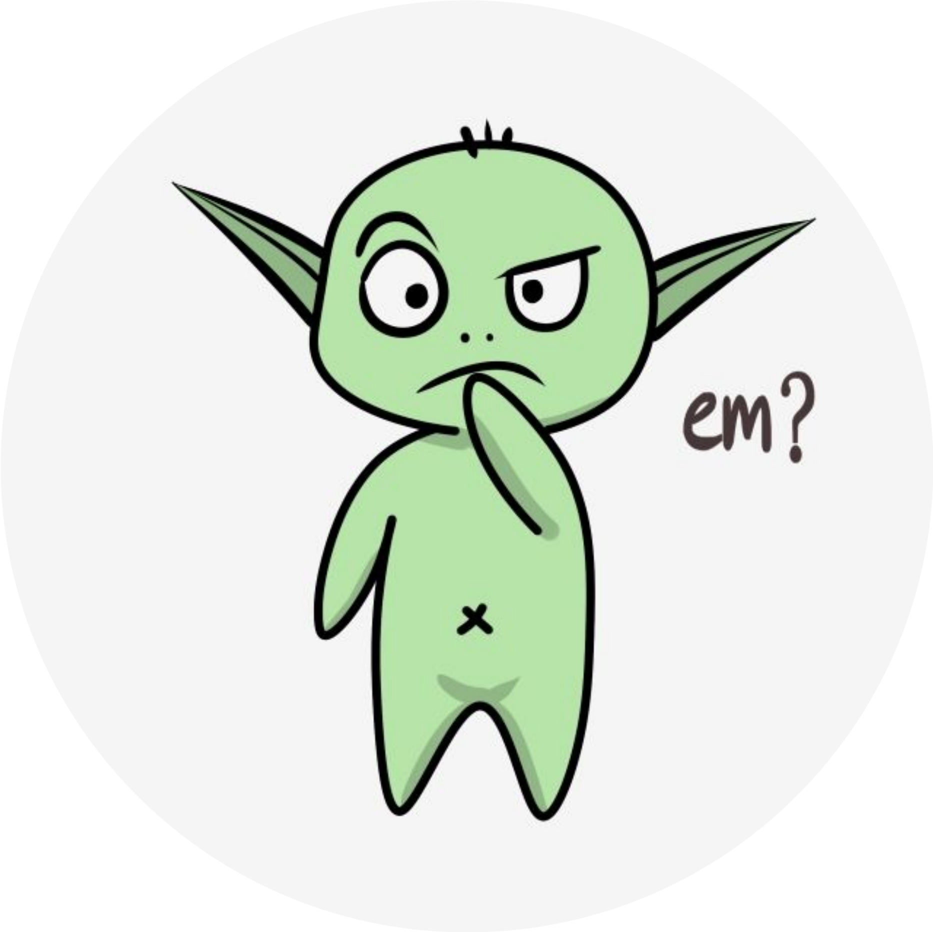
State-Action Value Functions Satisfy Recursive Relationships

Exercise 3.17 What is the Bellman equation for action values, that is, for q_π? It must give the action value $q_\pi(s, a)$ in terms of the action values, $q_\pi(s', a')$, of possible successors to the state–action pair (s, a). Hint: The backup diagram to the right corresponds to this equation. Show the sequence of equations analogous to (3.14), but for action values. □



q_π backup diagram

Marlos C. Machado



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

Optimal Policies and Optimal Value Functions

- Value functions define a partial ordering over policies.
  - $\pi \geq \pi'$ iff $v_\pi(s) \geq v_{\pi'}(s)$ for all $s \in \mathscr{S}$.
  - There is always at least one policy that is better than or equal to all other policies. The *optimal policy*.

$$v_*(s) \doteq \max_\pi v_\pi(s)$$

$$q_*(s,a) \doteq \max_\pi q_\pi(s,a)$$

$$q_*(s,a) = \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a]$$

Marlos C. Machado

Optimal Policies and Optimal Value Functions

- Because v_* is the value function for a policy, it must satisfy the self-consistency condition given by the Bellman equation for state values.

$$v_*(s) = \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s, a)$$

Optimal Policies and Optimal Value Functions

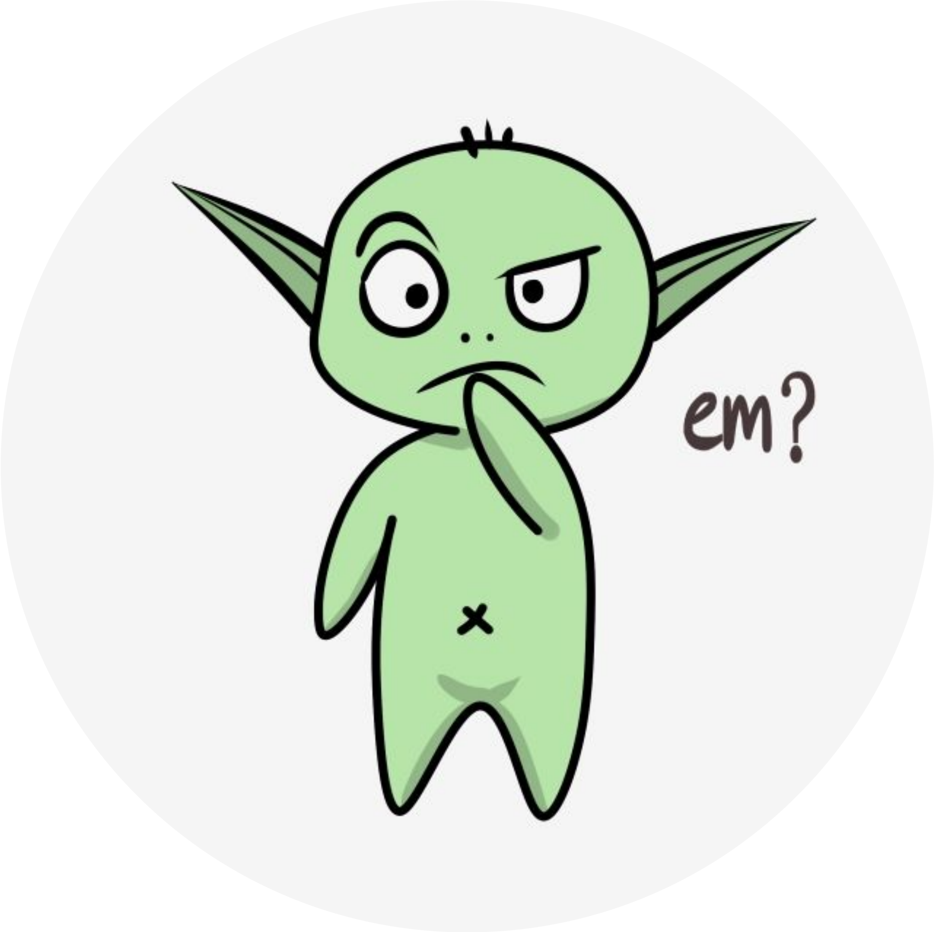
- Because v_* is the value function for a policy, it must satisfy the self-consistency condition given by the Bellman equation for state values.

$$\begin{aligned} v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s,a) \\ &= \max_a \mathbb{E}_{\pi_*}[G_t \mid S_t = s, A_t = a] \\ &= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a] \\ &= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma v_*(s')\big]. \end{aligned}$$

$$\begin{aligned} q_*(s,a) &= \mathbb{E}\Big[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1},a') \;\Big|\; S_t = s, A_t = a\Big] \\ &= \sum_{s',r} p(s',r|s,a)\Big[r + \gamma \max_{a'} q_*(s',a')\Big]. \end{aligned}$$

Also…

I have highlighted a couple of exercises during the class, but there are more.
Take a look at exercises in the Worksheet!



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

Reinforcement learning is very related to search algorithms

"Heuristic search methods can be viewed as expanding the right-hand side of the equation below several times, up to some depth, forming a "tree" of possibilities, and then using a heuristic evaluation function to approximate v_*, at the "leaf" nodes."

$$v_*(s) = \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma v_*(s')\big].$$

Yay! We solved sequential decision-making problems

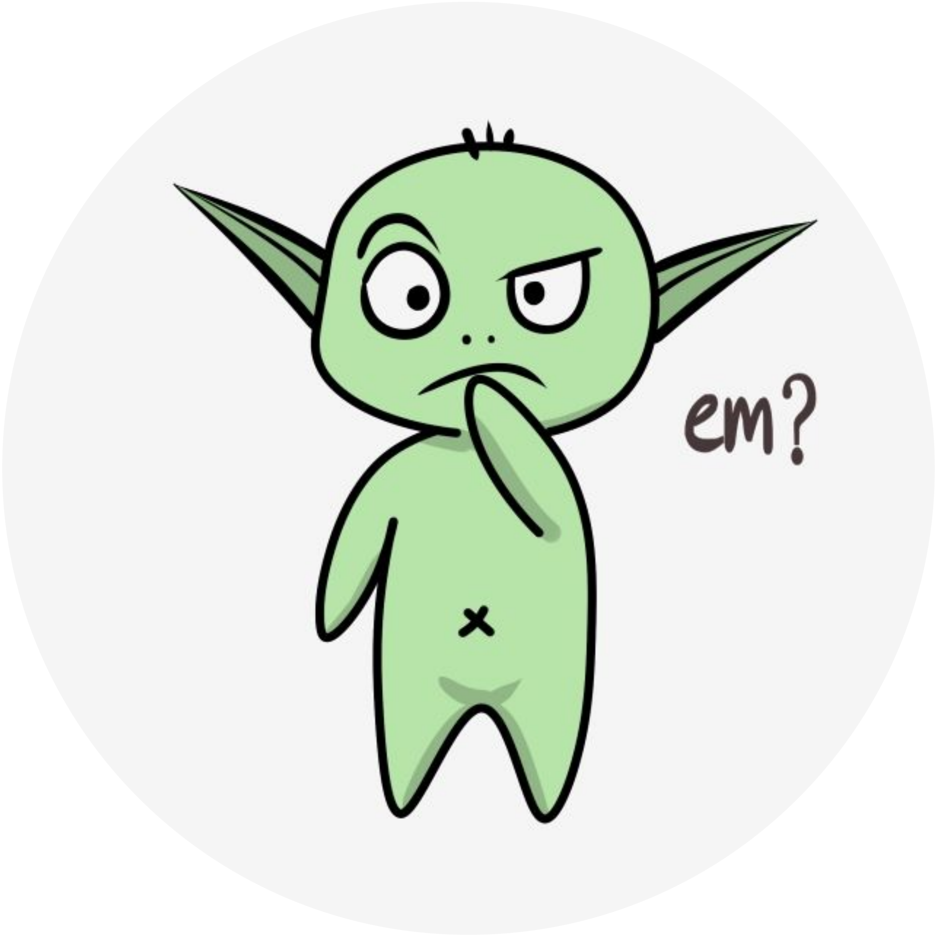
Except…

1.
2.
3.
4.

Yay! We solved sequential decision-making problems

Except…

1. we need to know the dynamics of the environment
2. we have enough computational resources to solve the system of linear eq.
3. actually, we need to solve a system of nonlinear equations (the optimality ones)
4. the Markov property

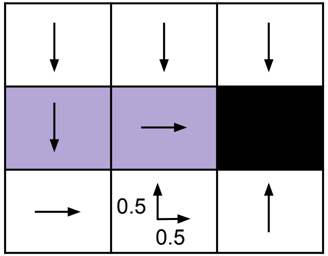


https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Example: Value Function Computation

Consider the 8-state MDP on the side. It has four actions available: {`up`, `down`, `left`, and `right`}. Its dynamics are deterministic, except at the purple states, where the agent can go up with 40% chance, regardless of the action taken, and 60% chance one goes to the intended direction. The reward is +1 upon entering state s_6, +2 upon entering the terminal state, and 0 otherwise. Let $\gamma = 0.8$. Consider the policy below:

s_1	s_2	s_3
s_4	s_5	
s_6	s_7	s_8



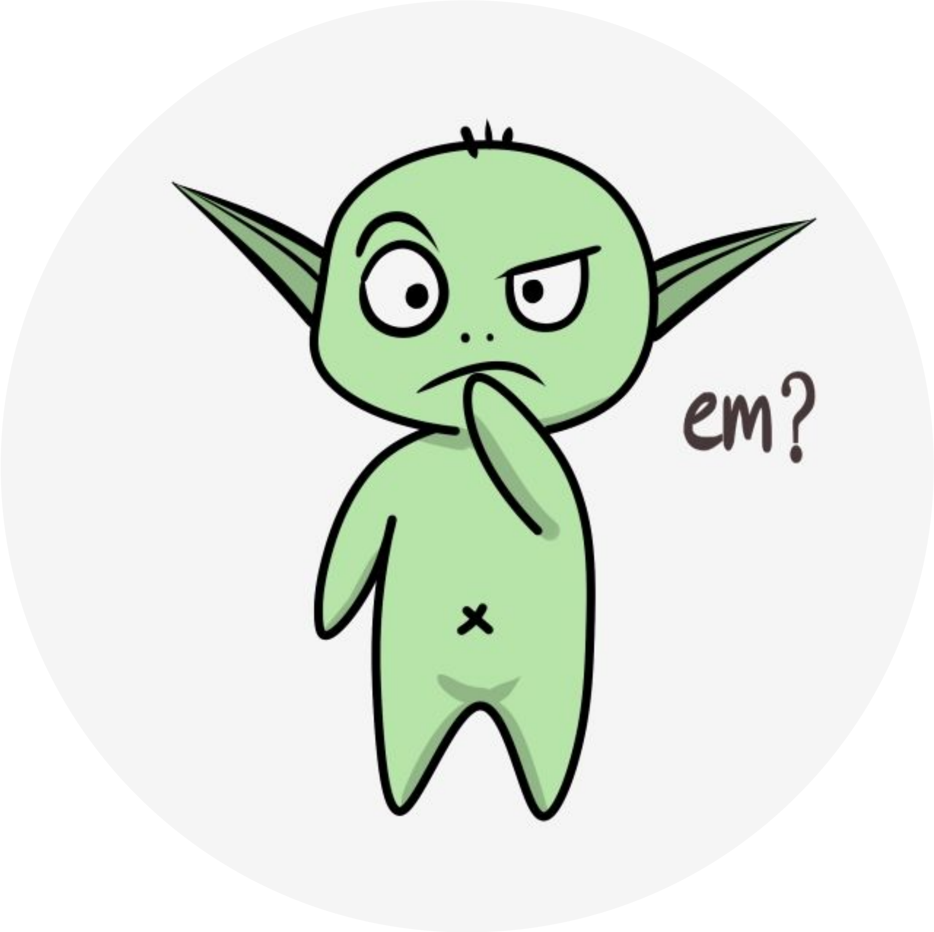
a) What's $v_\pi(s_4)$?

Recall

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_\pi(s')\Big], \quad \text{for all } s \in \mathcal{S}$$

Solution: Value Function Computation

Solution: Value Function Computation



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