“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

Class 10/ 35

Plan

- Value Functions and Bellman Equations
  - Non–comprehensive 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

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

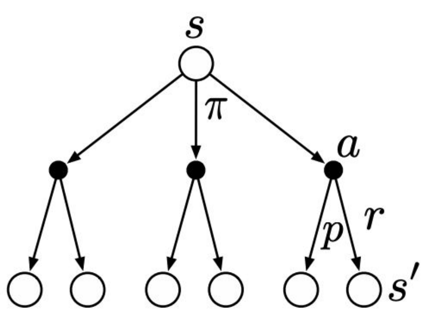
Value Functions Satisfy Recursive Relationships

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

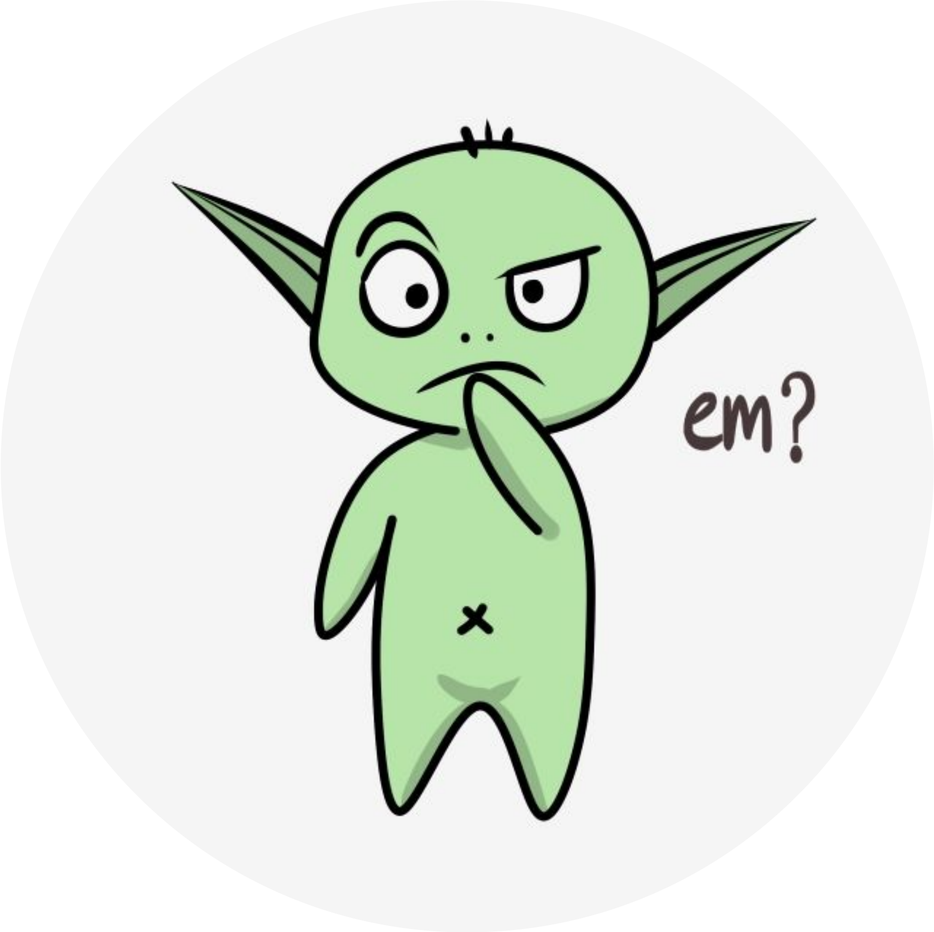
Value Functions Satisfy Recursive Relationships

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

This is a system of linear equations!



Backup diagram for v_π



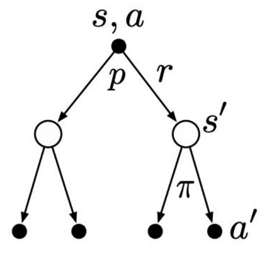
https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Example

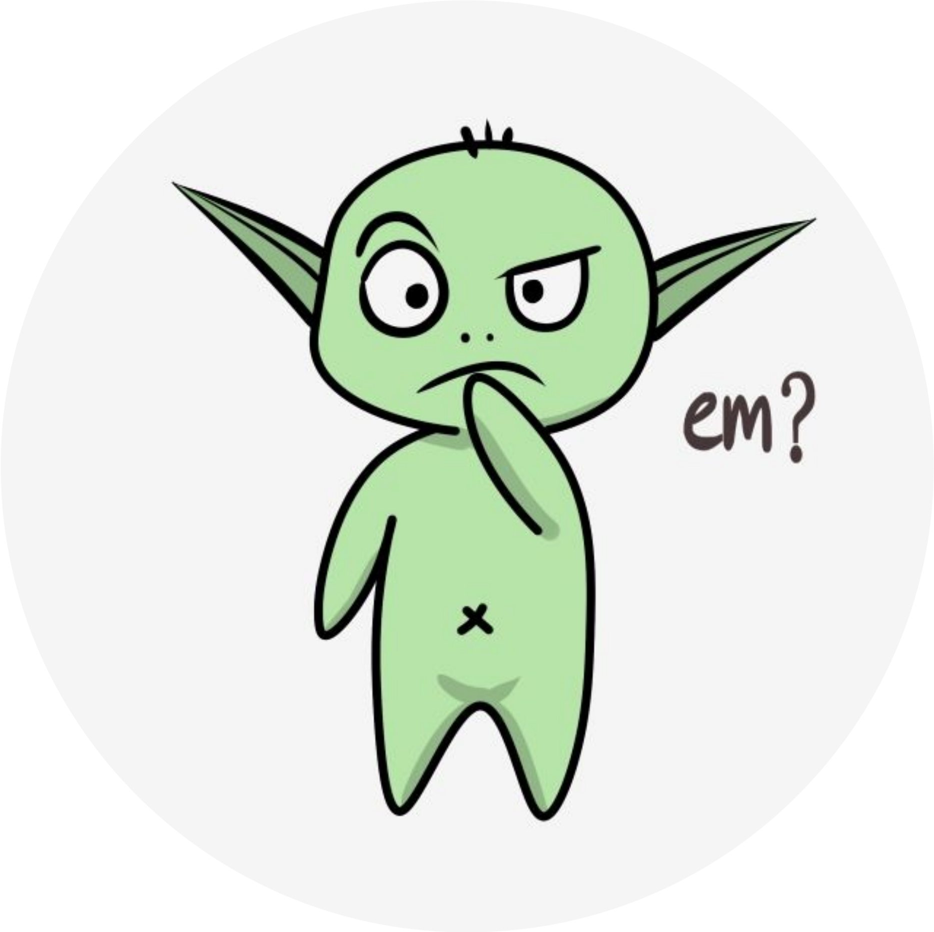
Whiteboard

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



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*.

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

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

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

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.

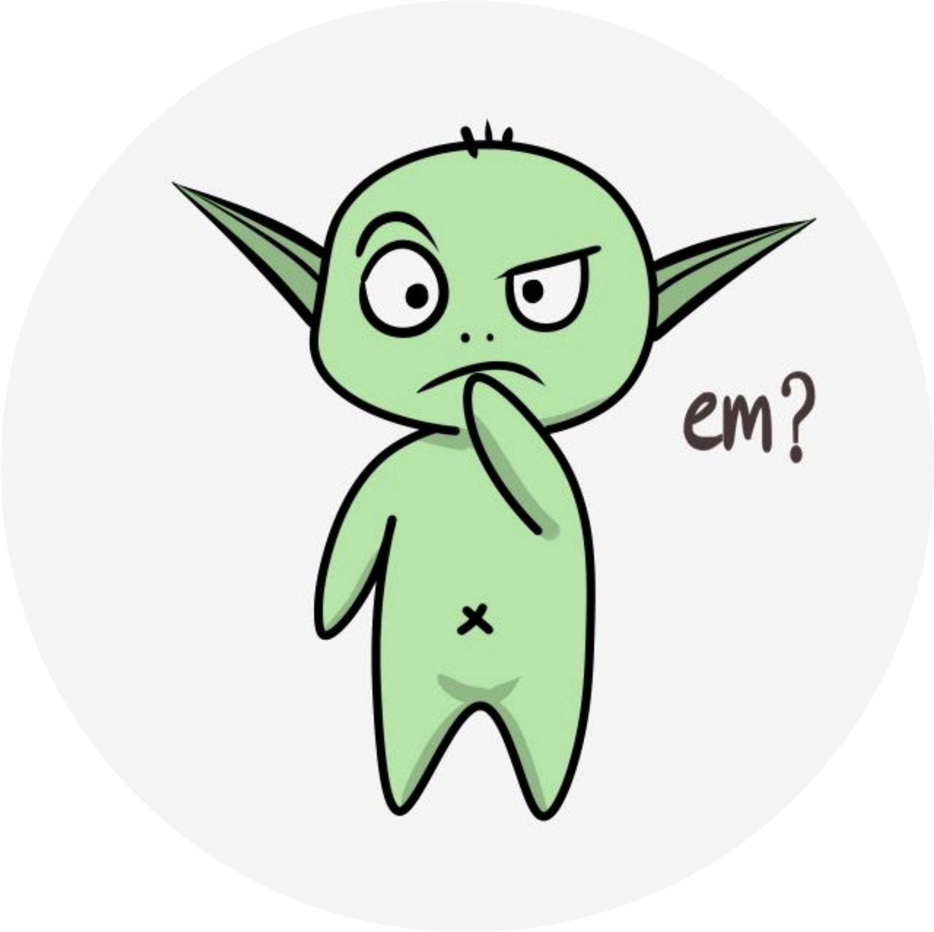
$$v_*(s) = \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s, a)$$

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.

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

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



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."

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

Yay! We solved sequential decision-making problems

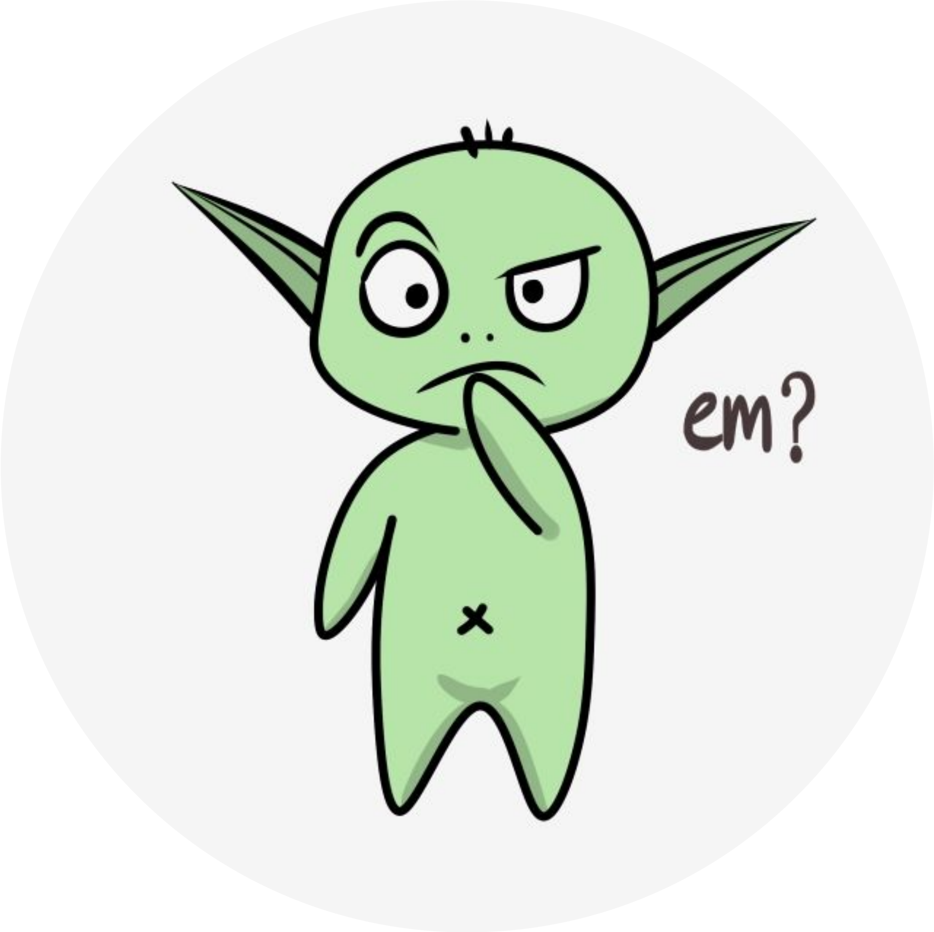
Except…

1.
2.
3.

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. the Markov property



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

Next class

- What I plan to do:
    - Talk about optimality.
    - Exercises and examples.