"To succeed, planning alone is insufficient.
One must improvise as well."

Isaac Asimov, *Foundation*

CMPUT 365
Introduction to RL

Marlos C. Machado

Coursera 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.

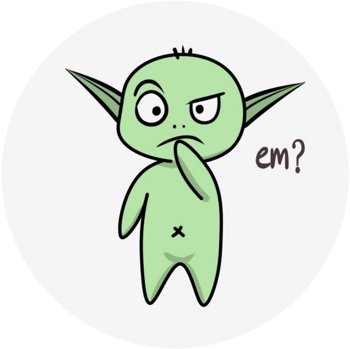
At the end of the term, I **will not port grades** from the public session in Coursera.

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

Reminders and Notes

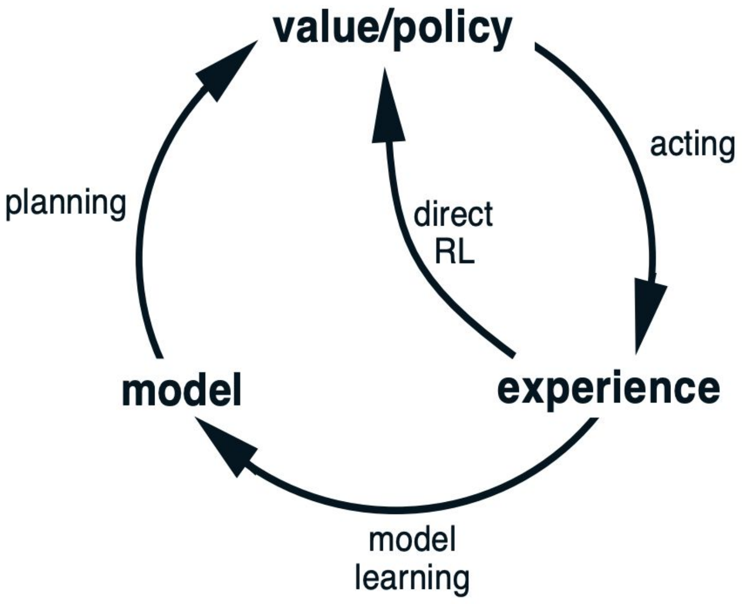
- What I plan to do today:
  - Wrap up Planning and Learning with Tabular Methods (Chapter 8).
  - An exercise.
- Programming assignment is due today.
- Midterm 2 is Friday.
- Monday is a new week, starting the 3*rd* module of Coursera (33 pending invites)
  Prediction and Control with Function Approximation
  On-policy Prediction with Approximation.

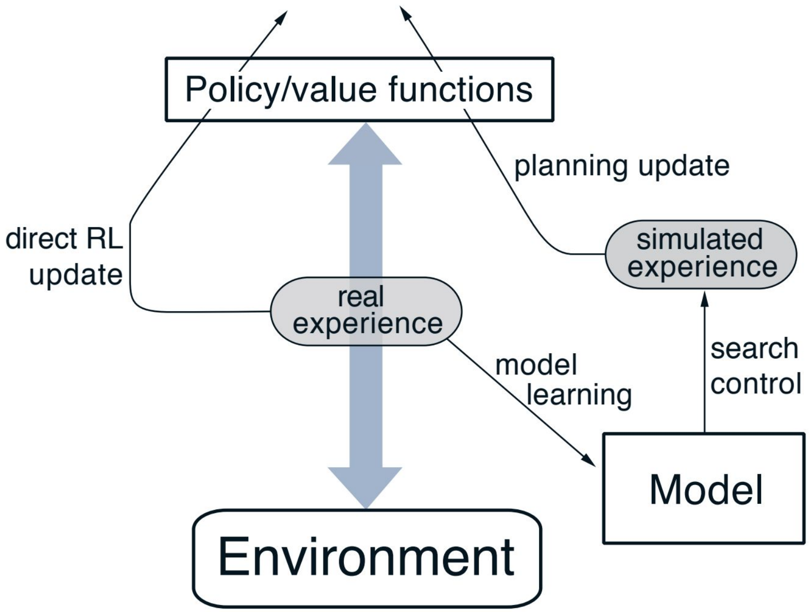
Please, interrupt me at any time!

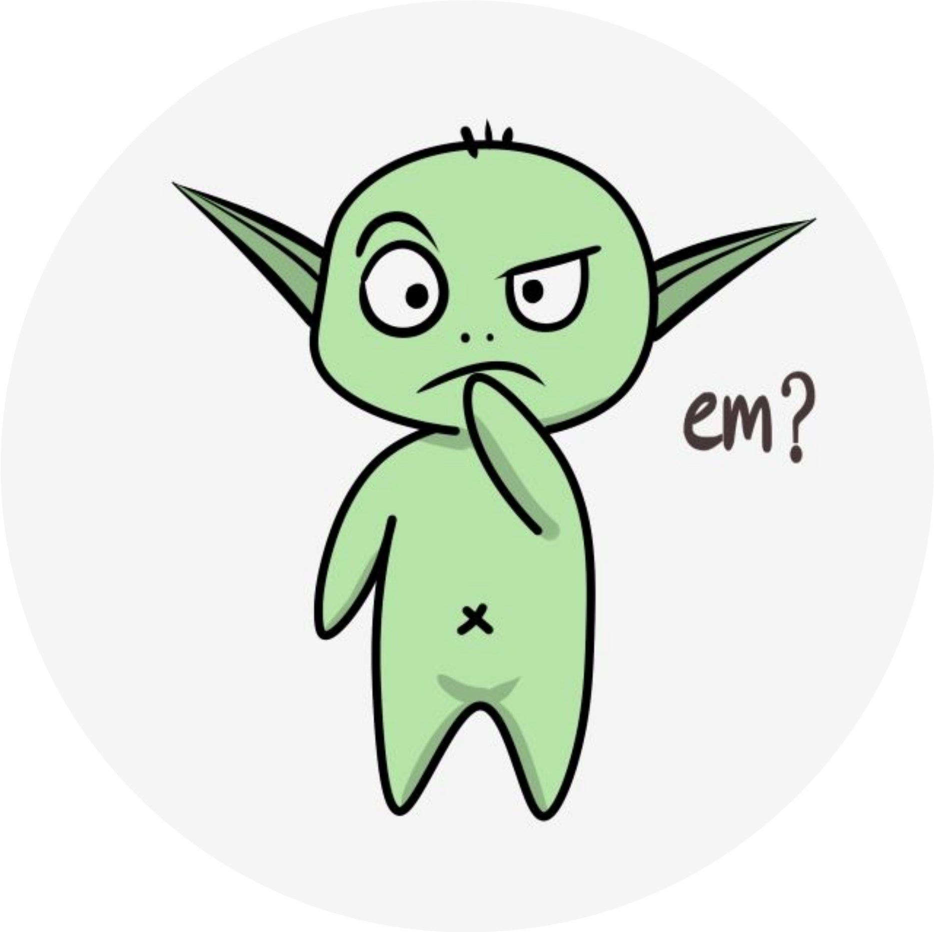


https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Last Class


value/policy
acting
planning
direct
RL
model
experience
model
learning


Policy/value functions
planning update
direct RL
update
simulated
experience
real
experience
model
learning
search
control
Model
Environment

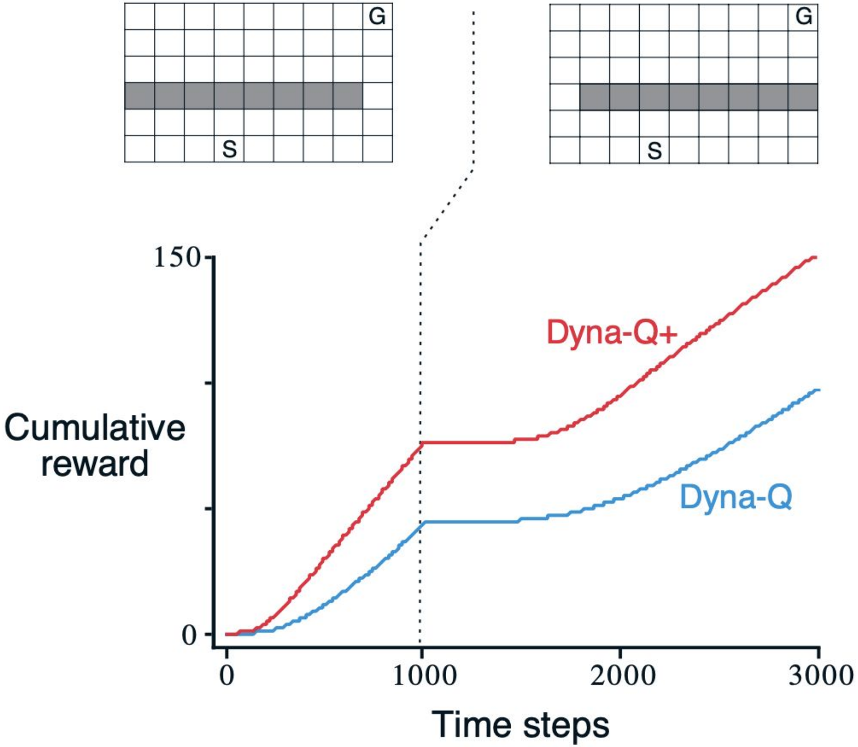


https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

When the Model Is Wrong

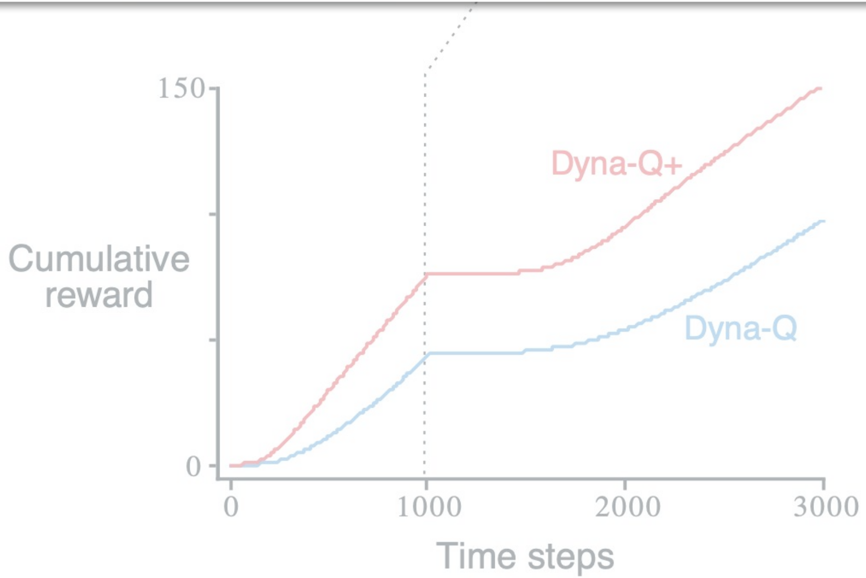
- A model can be wrong for all sorts of reasons (e.g., stochastic environment, function approximation, non-stationarity in the environment).
- An incorrect model often leads to suboptimal policies.
- One needs to constantly explore to refine the learned model.
    - Exploration: take actions that improve the model.
    - Exploitation: behaving in the optimal way given the current model.
- Dyna-Q+: Provides “bonus rewards” for long-untried actions.
  Specifically, consider the reward $r + \kappa\sqrt{\tau}$, where τ is the number of time steps since that transition was tried for the last time.

Dyna-Q+ Sometimes Works ¯_(ツ)_/¯

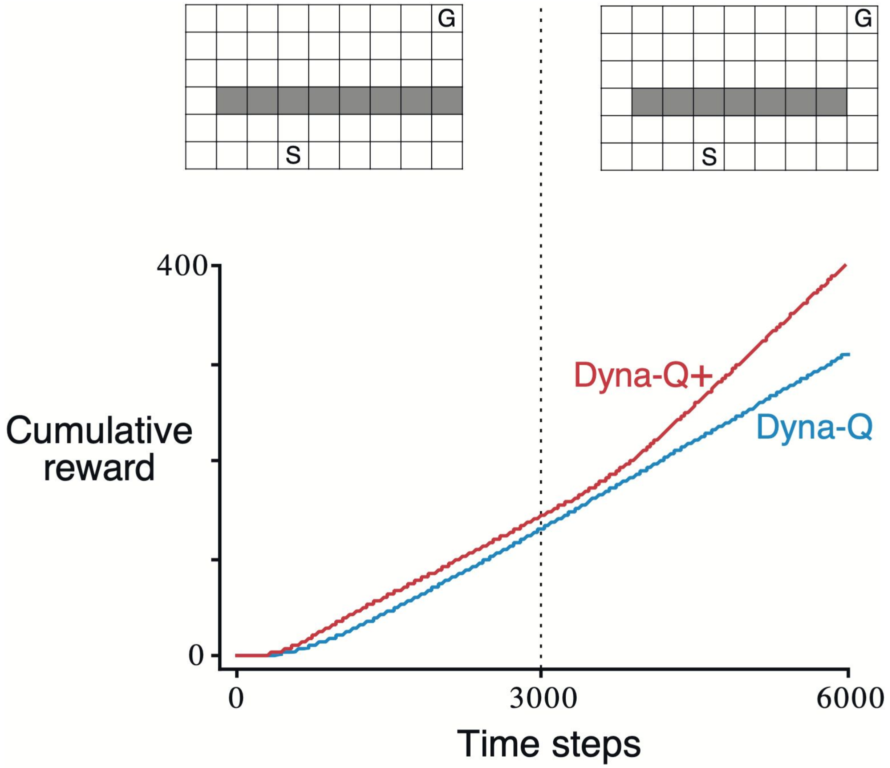


Dyna-Q+ Sometimes Works ¯_(ツ)_/¯

Exercise 8.2 Why did the Dyna agent with exploration bonus, Dyna-Q+, perform better in the first phase as well as in the second phase of the blocking and shortcut experiments? □

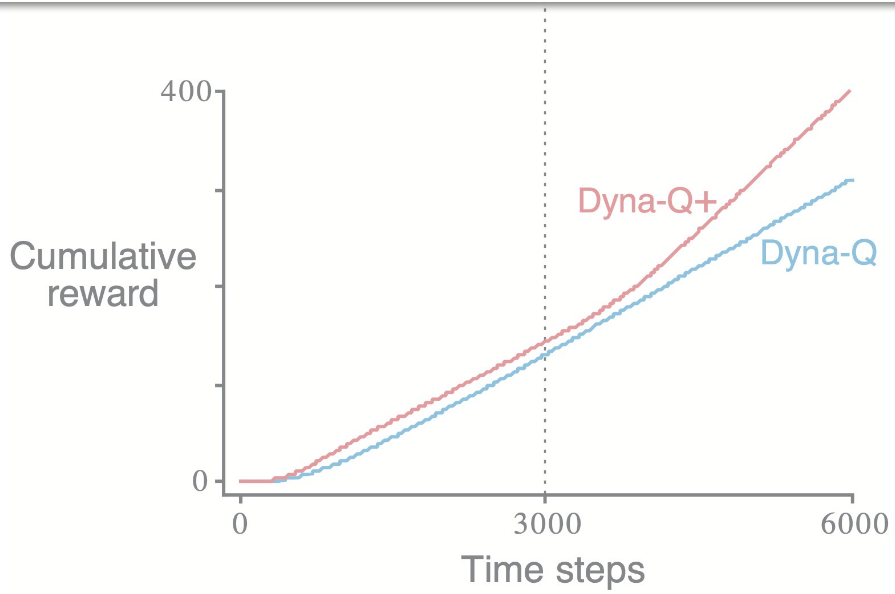


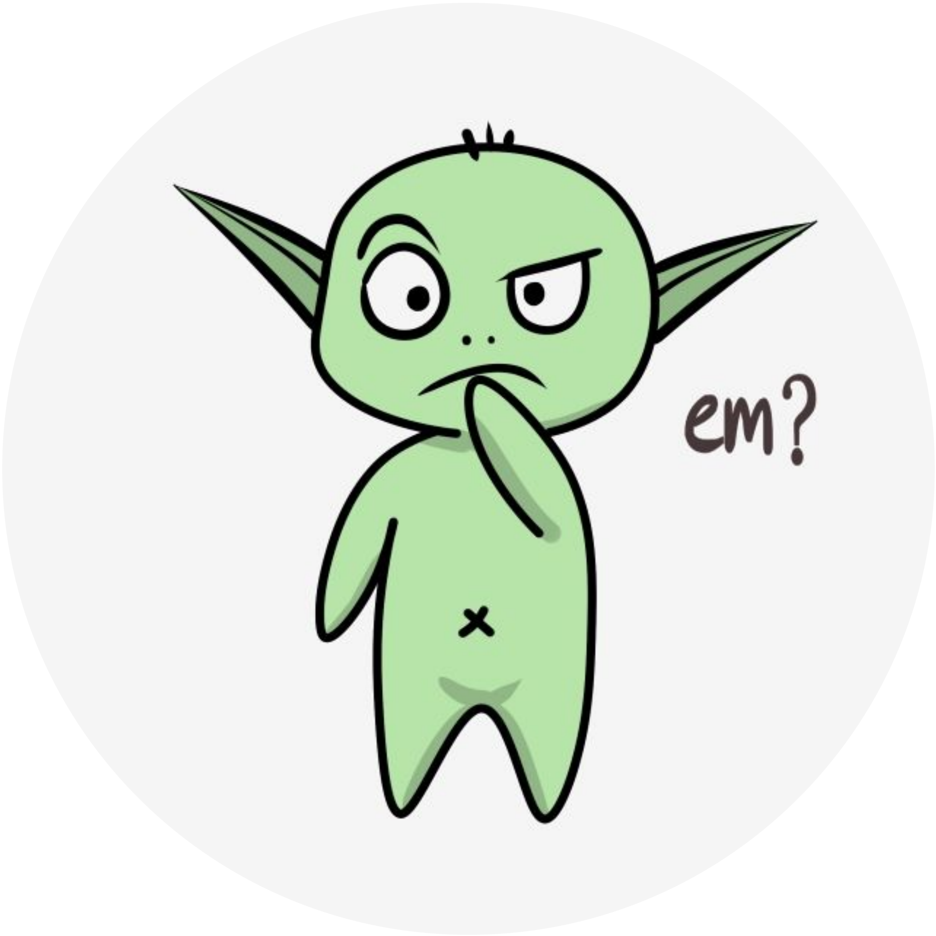
Sometimes it is Much Harder



Sometimes it is Much Harder

Exercise 8.3 Careful inspection of Figure 8.5 reveals that the difference between Dyna-Q+ and Dyna-Q narrowed slightly over the first part of the experiment. What is the reason for this? □

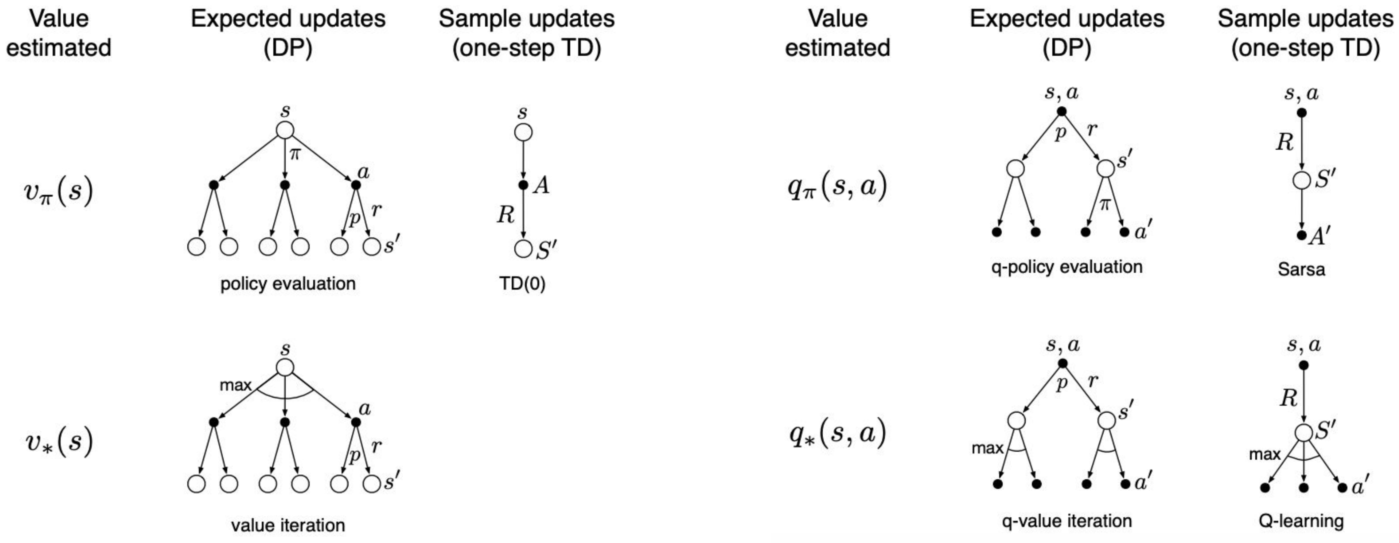


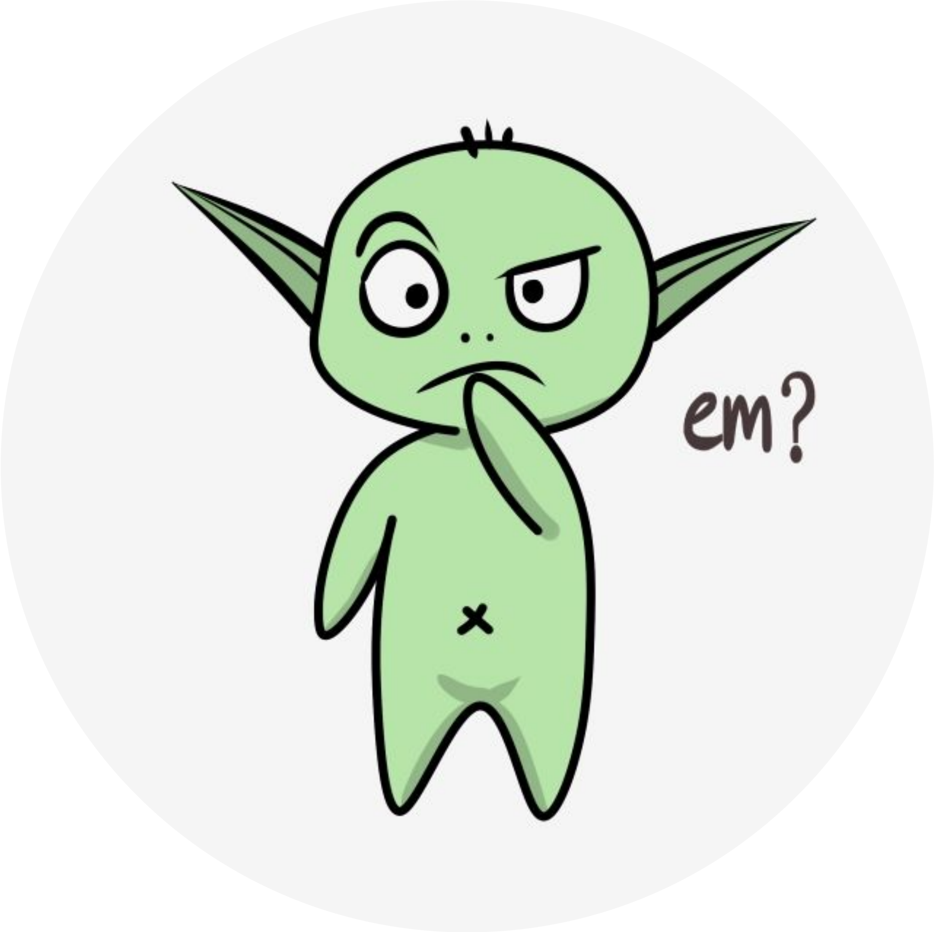


https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Expected vs. Sample Updates

- There are three dimensions in the updates one can do:
  - Should we use state values or action values?
  - Should we estimate the value for the optimal policy or for an arbitrary given policy?
  - Should we use expected or sample updates?





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

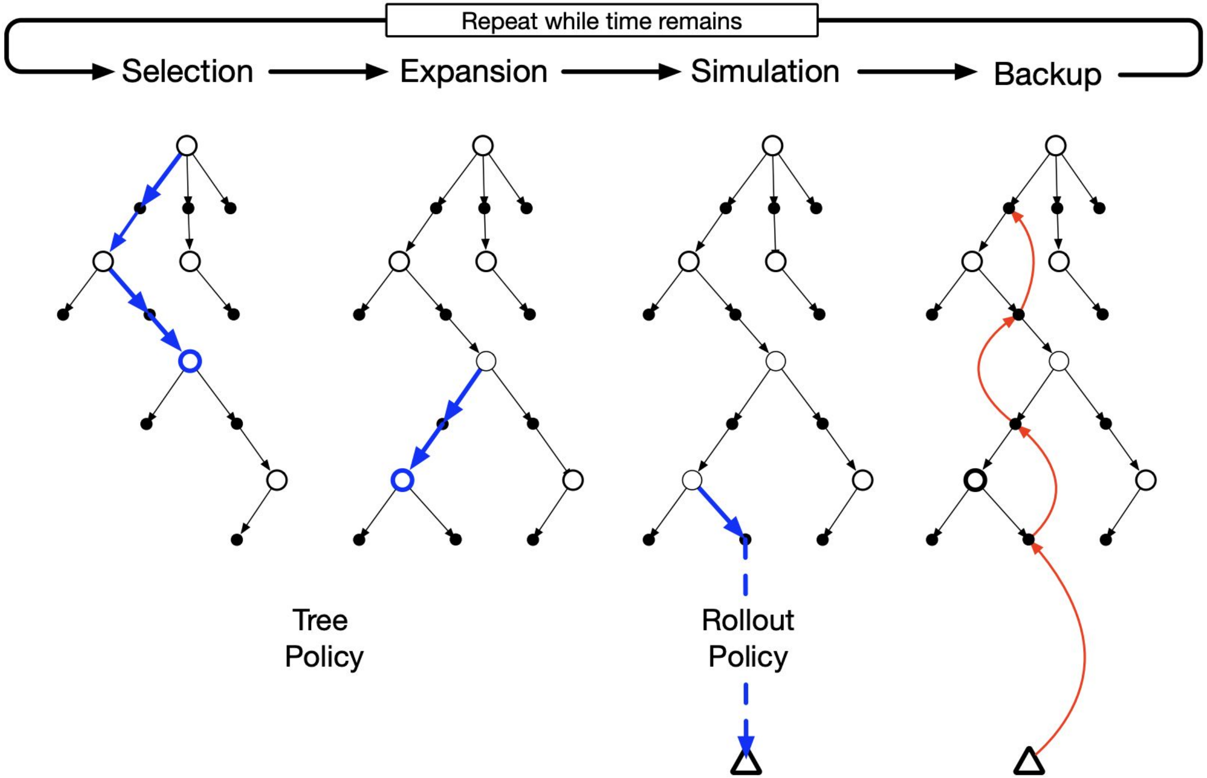
Rollout Algorithms

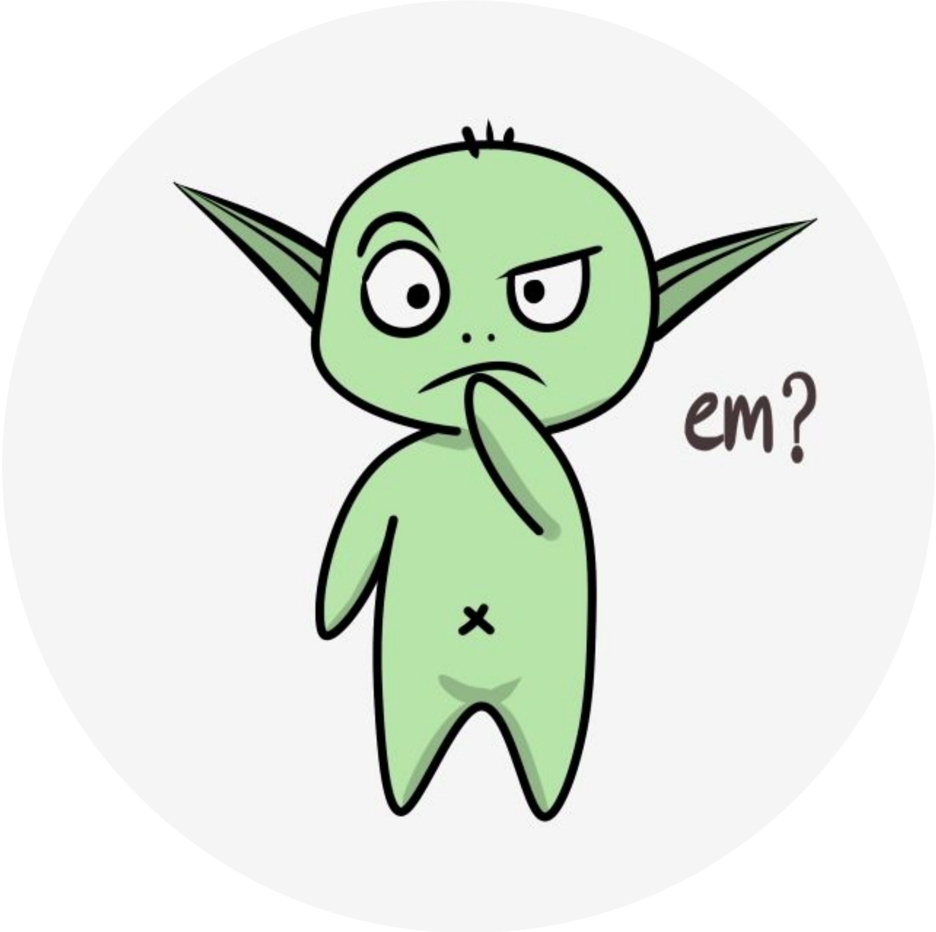
- Rollout algorithms are decision-time planning algorithms based on MC control applied to simulated trajectories that all begin at the current environment state.
  - They estimate action values for a given policy by averaging the returns of many simulated trajectories that start with each possible action and then follow the given policy.
- Unlike the Monte Carlo control algorithms previously described, the goal of a rollout algorithm is not to estimate a complete optimal action-value function, q_*, or a complete action-value function, q_π, for a given policy π.
  - They produce Monte Carlo estimates of action values only for each current state and for a given policy usually called the rollout policy.
- They are not learning algorithms *per se*, but they do leverage the RL toolkit.

Monte Carlo Tree Search (MCTS)

- MCTS is a great example of a rollout, decision-time planning algorithm.
  - But enhanced by the addition of a means for accumulating value estimates obtained from the MC simulations in order to successively direct simulations toward more highly-rewarding trajectories.
- The core idea of MCTS is to successively focus multiple simulations starting at the current state by extending the initial portions of trajectories that have received high evaluations from earlier simulations.
  - Monte Carlo value estimates are maintained only for the subset of state–action pairs that are most likely to be reached in a few steps, which form a tree rooted at the current state.

Monte Carlo Tree Search (MCTS)



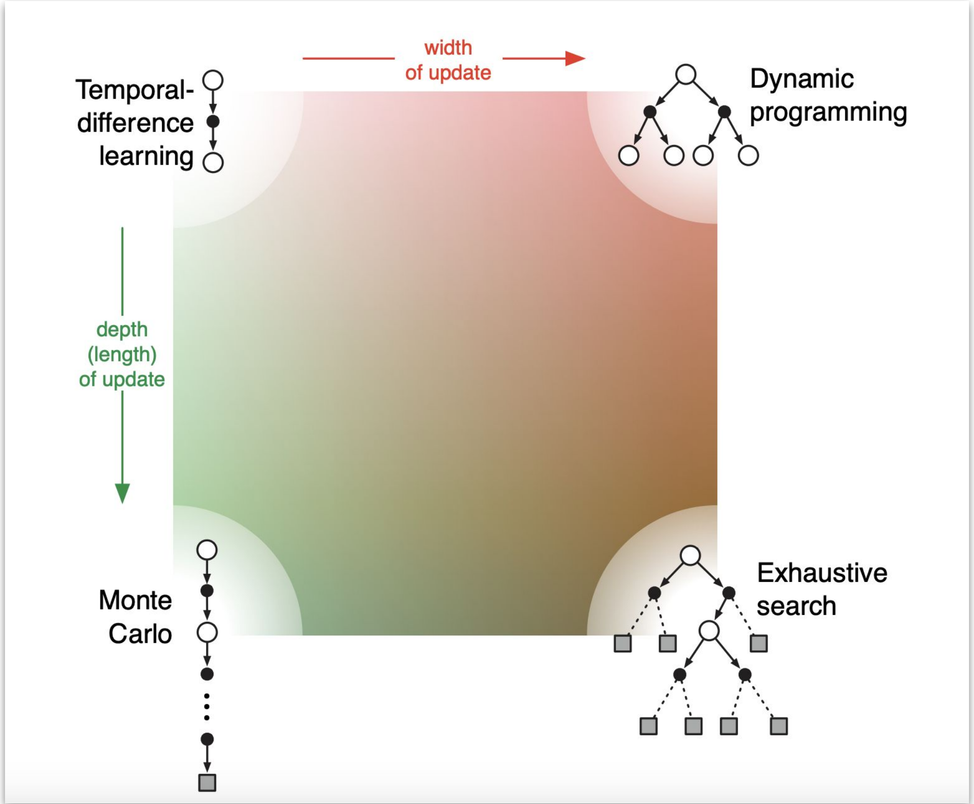


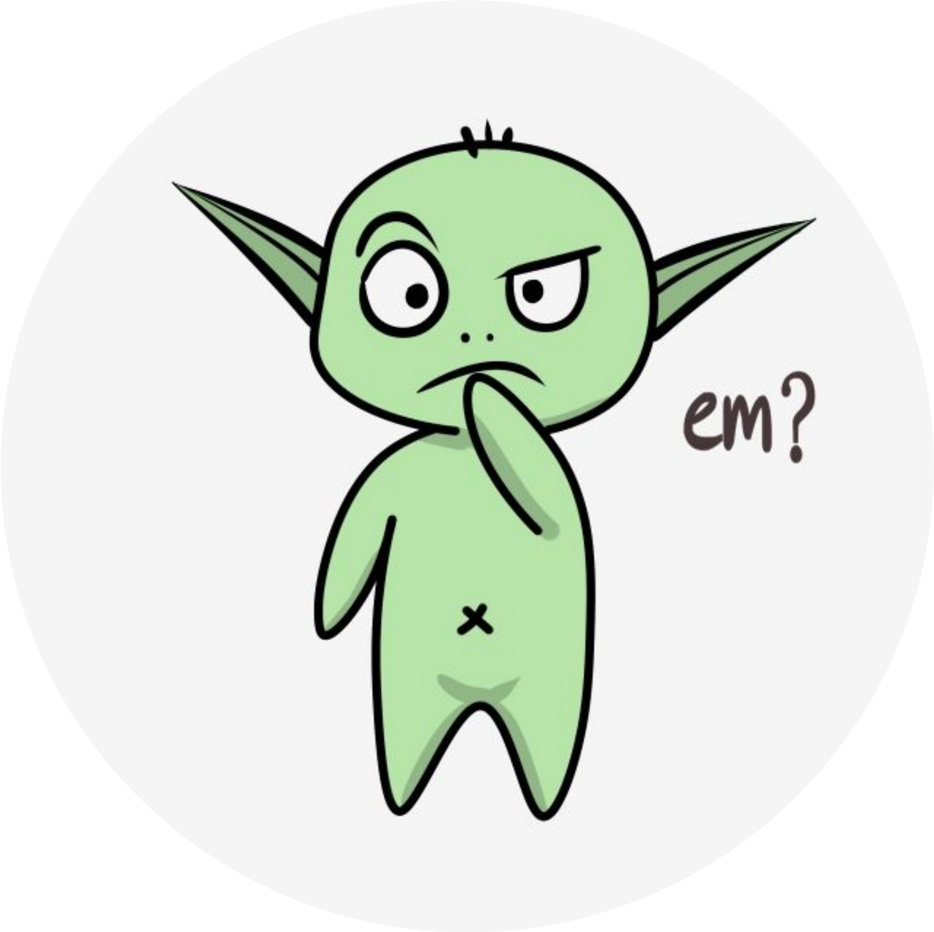
https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Wrapping Up

- We have finished Part I of the textbook, Tabular Solution Methods.
- Reinforcement learning can be seen as being more than a collection of individual methods, but a coherent set of ideas cutting across methods.
    - They all seek to estimate value functions.
    - They all operate by backing up values along actual or possible state trajectories.
    - They all follow the general strategy of generalized policy iteration (GPI).

Wrapping Up


width
of update
Temporal-
difference
learning
Dynamic
programming
depth
(length)
of update
Monte
Carlo
Exhaustive
search



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

Exercise

Consider an MDP with three states $\mathscr{S} = \{1, 2, 3\}$, where each state has two possible actions $\mathscr{A} = \{1, 2\}$ and a discount rate $\gamma = 0.5$. Suppose estimates of Q(S, A) are initialized to 0 and you observed the following episode according to an unknown behaviour policy where S_3 is the terminal state.

$$S_0 = 1, A_0 = 1, R_1 = -7, S_1 = 3, A_1 = 2, R_2 = 5, S_2 = 1, A_2 = 1, R_3 = 10$$

(a) Suppose you used Q-learning with the above trajectory to estimate Q(S, A), what are your new estimates for Q(S = 1, A = 1) using $\alpha = 0.1$?

(b) What is one possible model for this environment? Is the model stochastic or deterministic?

(c) Suppose in the planning loop, after search control, we would like to update Q(S = 1, A = 1) with Q-planning. What are the possible outputs of Model(S = 1, A = 1)?

(d) If your model outputs $R = R_3$ and $S' = S_3$, what is Q(S = 1, A = 1) after one Q-planning update? Use the estimates of Q(S, A) from before.