"(...) Muad'Dib learned rapidly because his first training was in how to learn. And the first lesson of all was the basic trust that he could learn. It's shocking to find how many people do not believe they can learn, and how many more believe learning to be difficult. Muad'Dib knew that every experience carries its lesson."

Frank Herbert, *Dune*

CMPUT 365
Introduction to RL

Marlos C. Machado

Class 20/35

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.

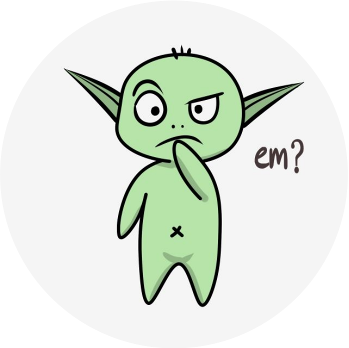
At the end of the term, I **will not port grades** from the public session in Coursera.

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

Reminders and Notes

- Midterm is next Friday. Chapters it will cover: 5, 6, and 8.
- What I plan to do today:
  - Wrap up TD Learning for Control (Second half of Chapter 6 of the textbook).
  - I probably won't finish my slides today, but we'll see.
- These next days will be busy
  - Programming assignment is due today.
  - Quiz on Planning, Learning & Acting is due on Monday.
  - Programming assignment on Planning, Learning & Acting is due on Wednesday.

Please, interrupt me at any time!



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

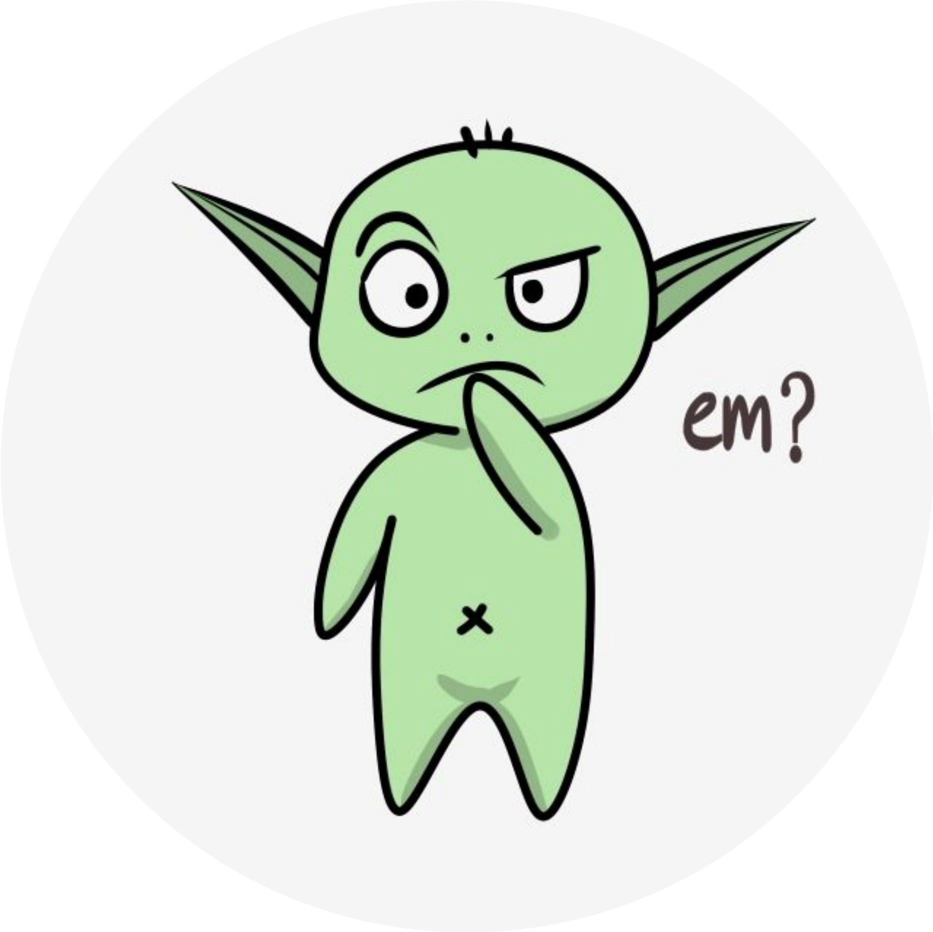
Last class: Sarsa and Q-Learning

- Sarsa:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\Big]$$

- Q-Learning:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\Big]$$



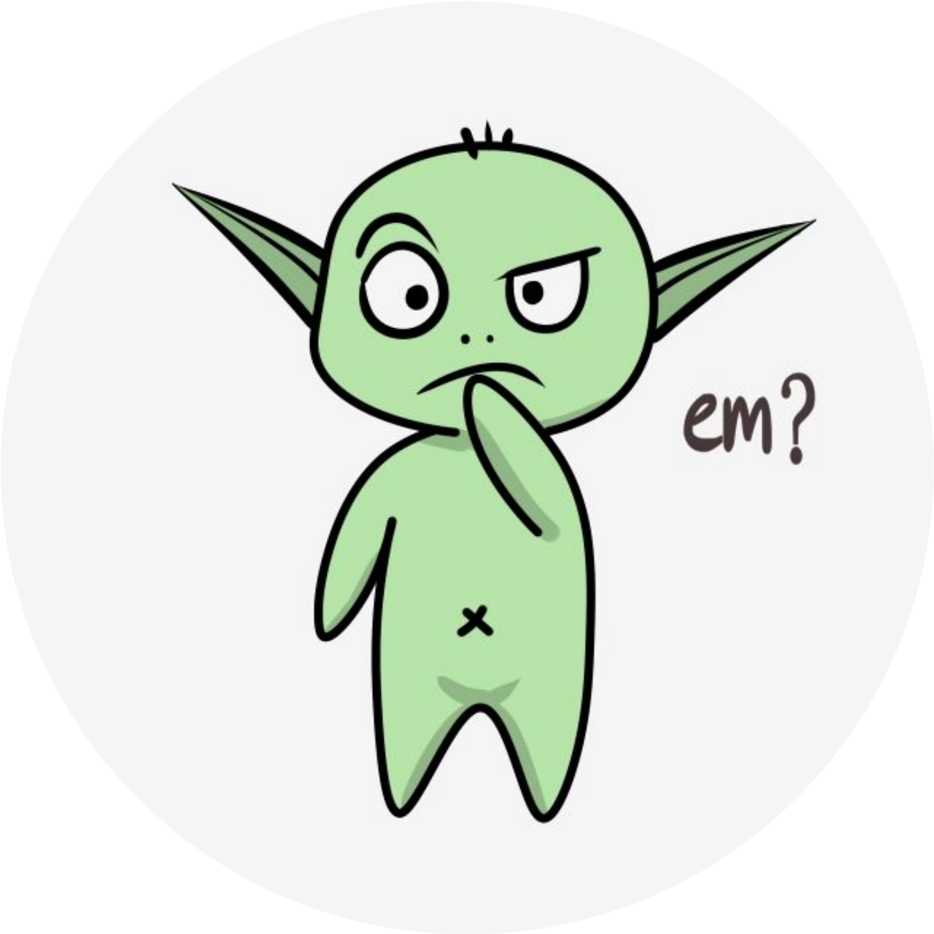
https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Discussion

Exercise 6.12. Suppose action selection is greedy. Is Q-learning then exactly the same algorithm as Sarsa? Will they make exactly the same action selections and weight updates?

$$Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma Q(S', A') - Q(S, A)\big]$$

$$Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$$



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

Expected Sarsa

What if instead of the maximum over next state-action pairs we used the expected value, taking into account how likely each action is under the current policy?

$$
\begin{aligned}
Q(S_t, A_t) &\leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \mathbb{E}_\pi[Q(S_{t+1}, A_{t+1}) \mid S_{t+1}] - Q(S_t, A_t)\Big] \\
&= Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t)\Big],
\end{aligned}
$$

Expected Sarsa

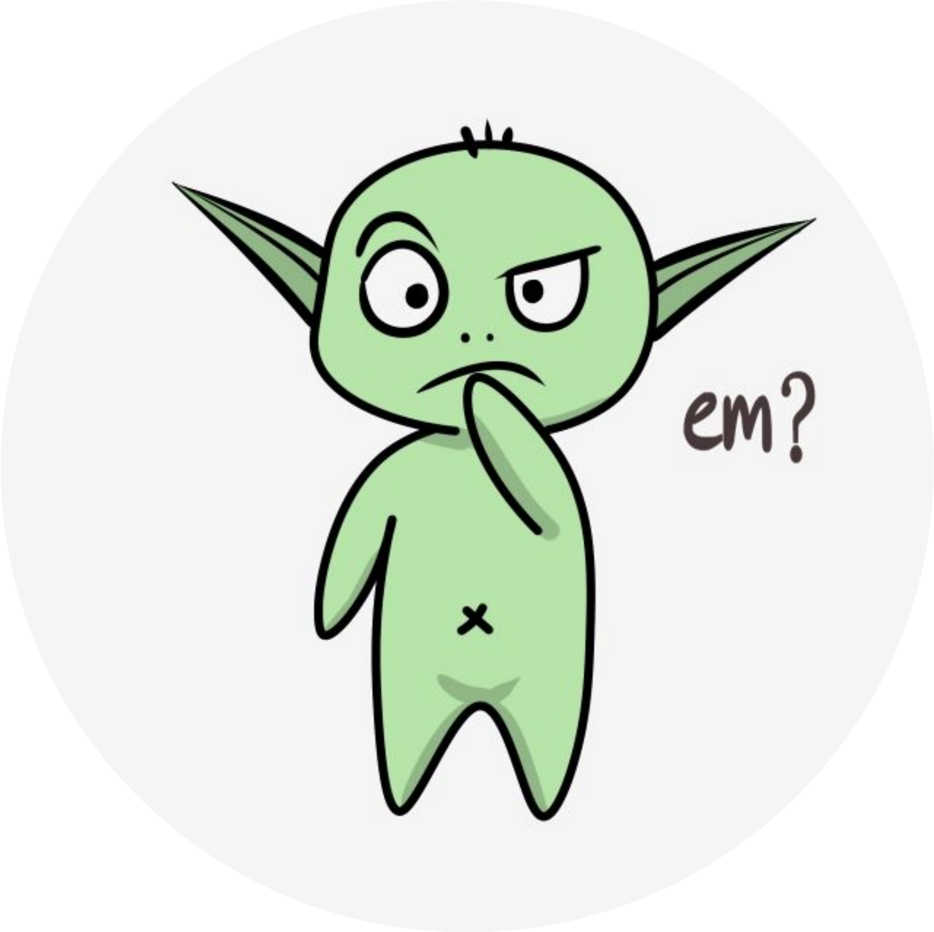
What if instead of the maximum over next state-action pairs we used the expected value, taking into account how likely each action is under the current policy?

$$\begin{aligned} Q(S_t, A_t) &\leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \mathbb{E}_\pi[Q(S_{t+1}, A_{t+1}) \mid S_{t+1}] - Q(S_t, A_t)\Big] \\ &= Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t)\Big], \end{aligned}$$

Expected Sarsa is more computationally expensive than Sarsa but, in return, it eliminates the variance due to the random selection of A_{t+1}.

Is Expected Sarsa on-policy or off-policy?

Expected can use a policy different from the target policy π to generate behavior (thus, it can be off-policy; although one can use it on-policy as well).



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

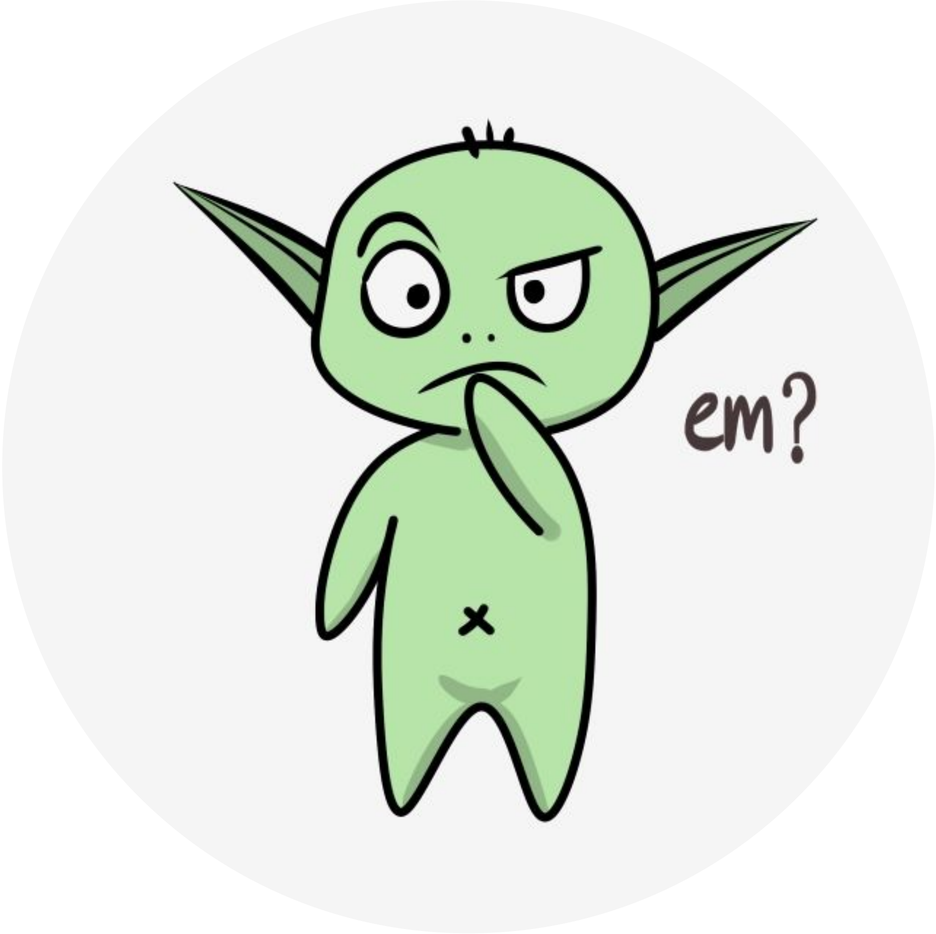
Maximization Bias

- The control algorithms we discussed so far use a maximization to get their target policies (either a max/greedy policy or an ε-greedy policy).
- *Maximization bias:* A maximum over estimated values is used implicitly as an estimate of the maximum value, which can lead to a significant positive bias.

Double Learning

- The issue is that we use the same samples to determine the maximizing action and to estimate its value.
- In Bandits:
  - Split the data, learn $Q_1(a)$ and $Q_2(a)$ to estimate $q(a)$.
  - Choose actions according to one estimate and get estimate from the other:
    $A^* = \text{argmax}_a\, Q_1(a)$ $\qquad$ $Q_2(A^*) = Q_2(\text{argmax}_a\, Q_1(a))$
  - This leads to unbiased estimates, that is: $\mathbb{E}[Q_2(A^*)] = q(A^*)$

$$Q_1(S_t, A_t) \leftarrow Q_1(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma Q_2\big(S_{t+1}, \arg\max_a Q_1(S_{t+1}, a)\big) - Q_1(S_t, A_t)\Big]$$



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

Double Q-Learning

Double Q-learning, for estimating $Q_1 \approx Q_2 \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q_1(s, a)$ and $Q_2(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, such that $Q(terminal, \cdot) = 0$

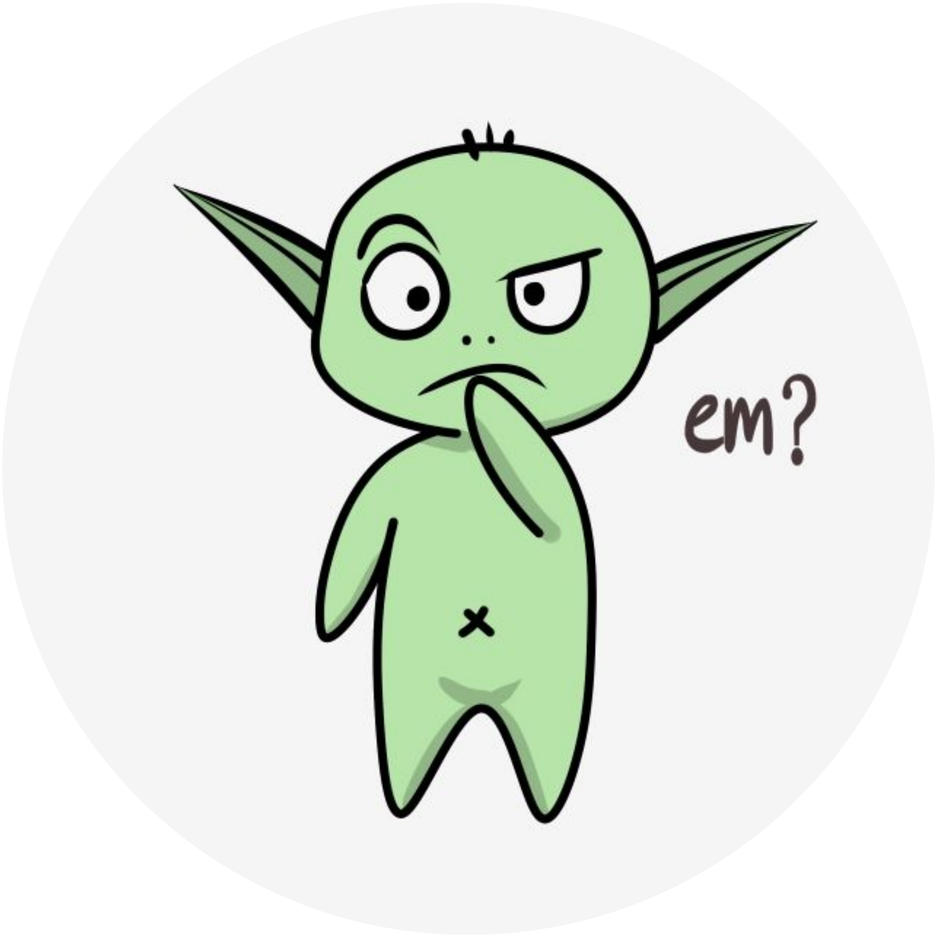
Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using the policy ε-greedy in $Q_1 + Q_2$
    Take action A, observe R, S'
    With 0.5 probabilility:

$$Q_1(S, A) \leftarrow Q_1(S, A) + \alpha\Big(R + \gamma Q_2\big(S', \arg\max_a Q_1(S', a)\big) - Q_1(S, A)\Big)$$

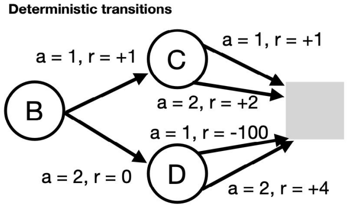
    else:

$$Q_2(S, A) \leftarrow Q_2(S, A) + \alpha\Big(R + \gamma Q_1\big(S', \arg\max_a Q_2(S', a)\big) - Q_2(S, A)\Big)$$

    $S \leftarrow S'$
  until S is terminal



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



Exercise

Consider the following MDP, with three states B, C, and D ($\mathscr{S}$ = {B, C, D}), and 2 actions ($\mathscr{A}$ = {1, 2}), with γ = 1.0. Assume the action values are initialized Q(s, a) = 0 ∀ s ∈ $\mathscr{S}$. The agent takes actions according to an ε-greedy policy with ε = 0.1.

1. What is the optimal policy for this MDP and what are the action-values corresponding to the optimal policy: q*(s, a)?
2. Imagine the agent experienced a single episode, and the following experience: S_0 = B, A_0 = 2, R_1 = 0, S_1 = D, A_1 = 2, R_2 = 4. What are the Sarsa updates during this episode, assuming α = 0.1? Start with state B, and perform the Sarsa update, then update the value of state D.
3. Using the sample episode above, compute the updates Q-learning would make, with α = 0.1? Again, start with state B, and then state D.
4. Let's consider one more episode: S_0 = B, A_0 = 2, R_1 = 0, S_1 = D, A_1 = 1, R_2 = -100. What would the Sarsa updates be? And what would the Q-learning updates be?
5. Assume you see one more episode, and it's the same on as in 4. Once more update the action values, for Sarsa and Q-learning. What do you notice?