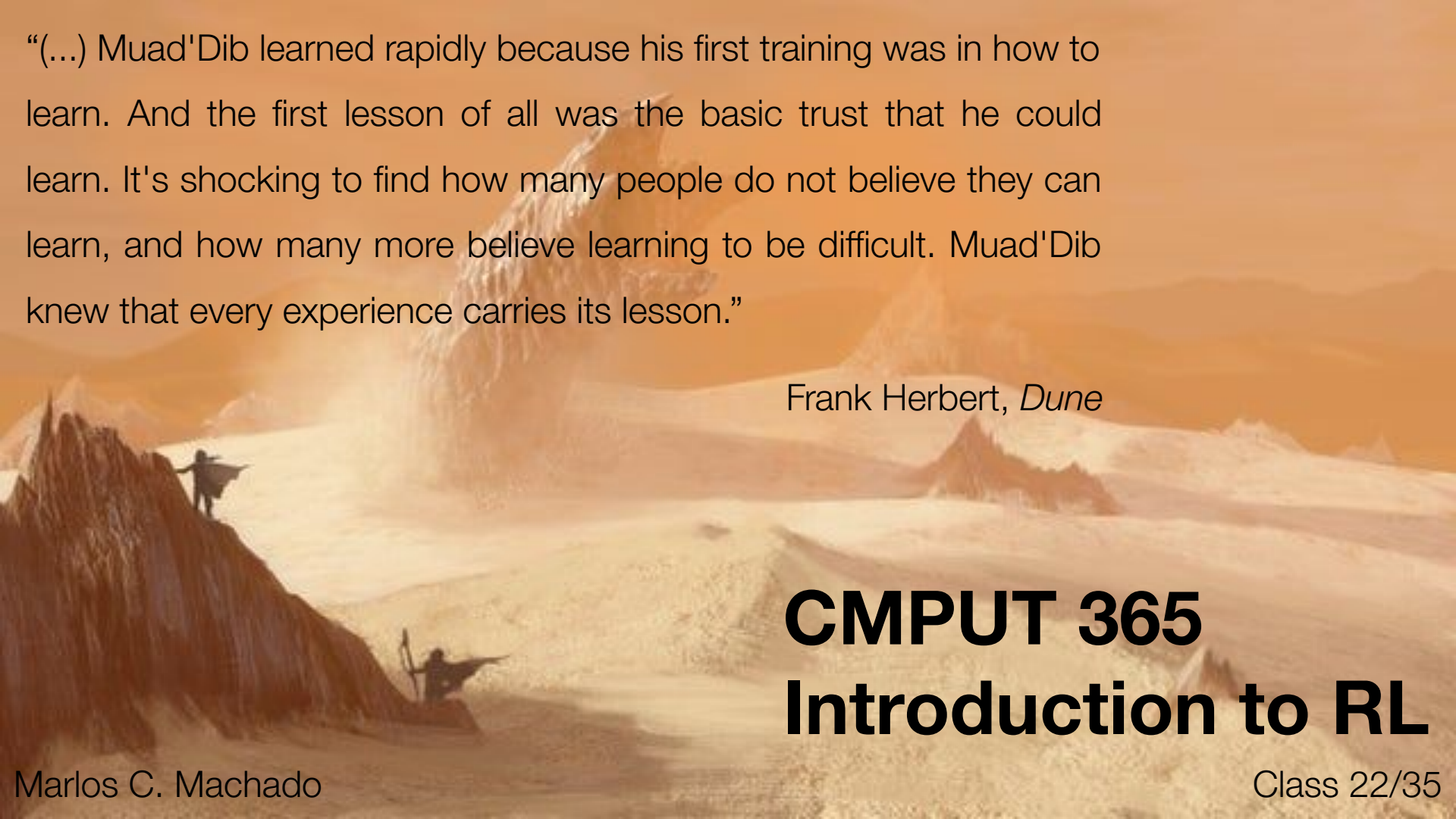

“(...) Muad'Dib learned rapidly because his first training was in how to learn. And the first lesson of all was the basic trust that he could learn. It's shocking to find how many people do not believe they can learn, and how many more believe learning to be difficult. Muad'Dib knew that every experience carries its lesson.”
Frank Herbert, *Dune*
CMPUT 365
Introduction to RL
Marlos C. Machado
Class 22/35

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.

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`.

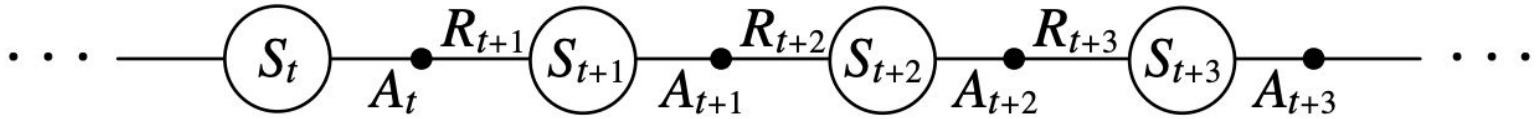
Plan / Reminder II

- What I plan to do today:
  - (Wrap up?) TD Learning for Control (Second half of Chapter 6 of the textbook).
- Programming assignment is due today.

Please, interrupt me at any time!

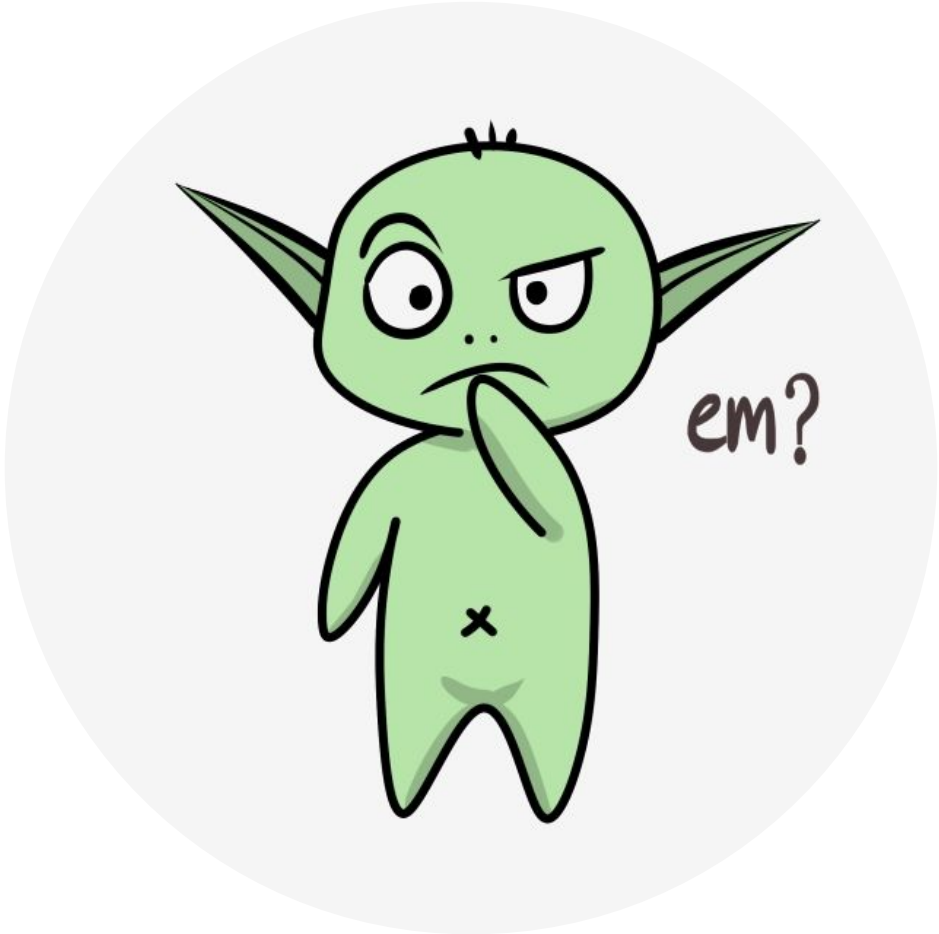
https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Last Class: Sarsa and Q-Learning



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

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



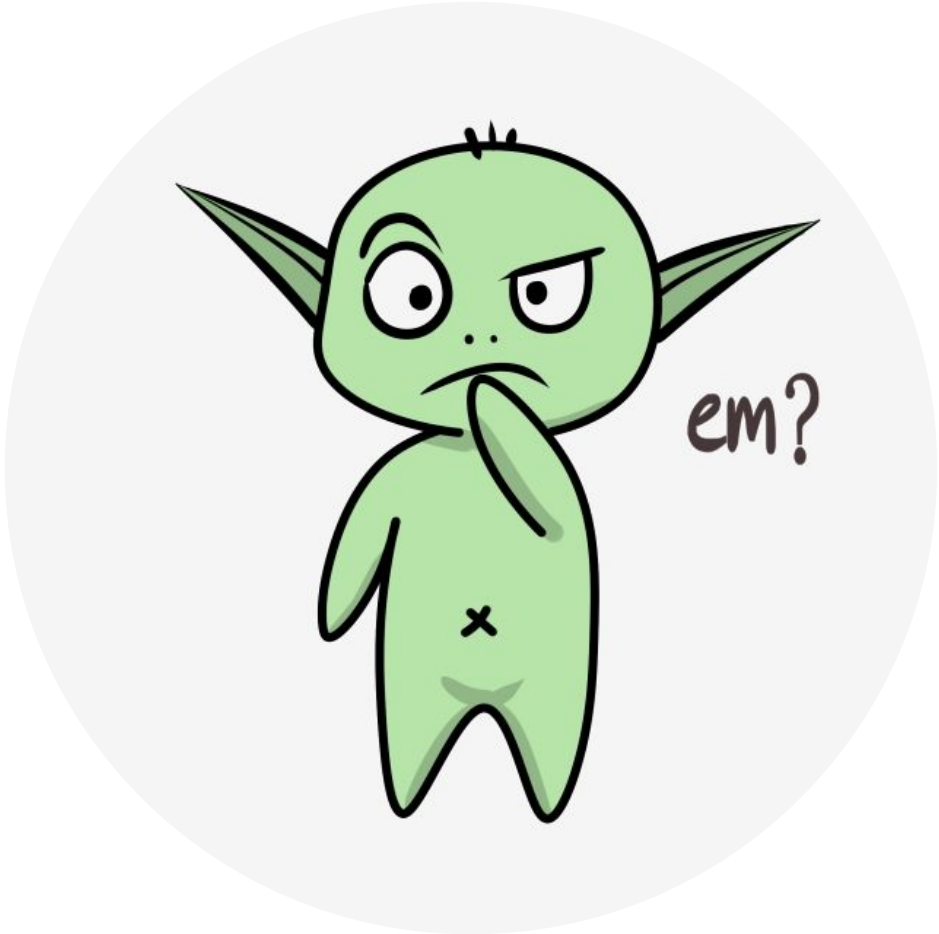
https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

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

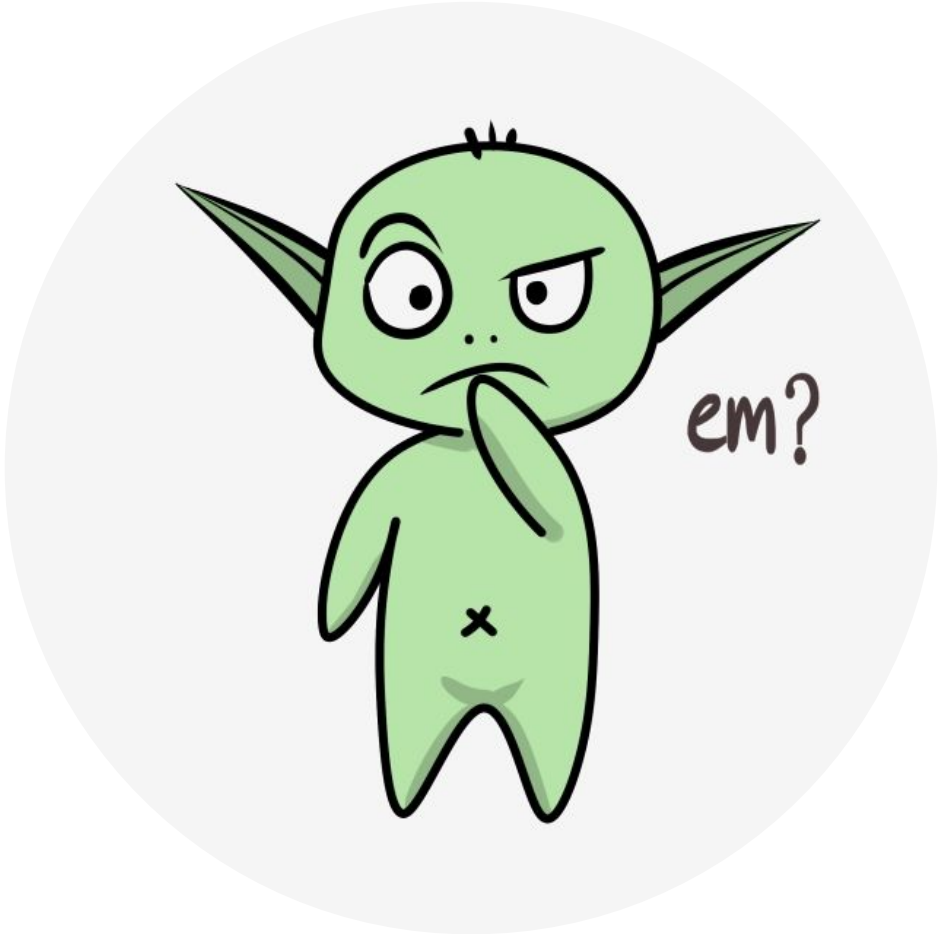
What if instead of the maximum over next state-action pairs we used the expected value, taking into account how likely each action is under the current policy?

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

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

Exercise

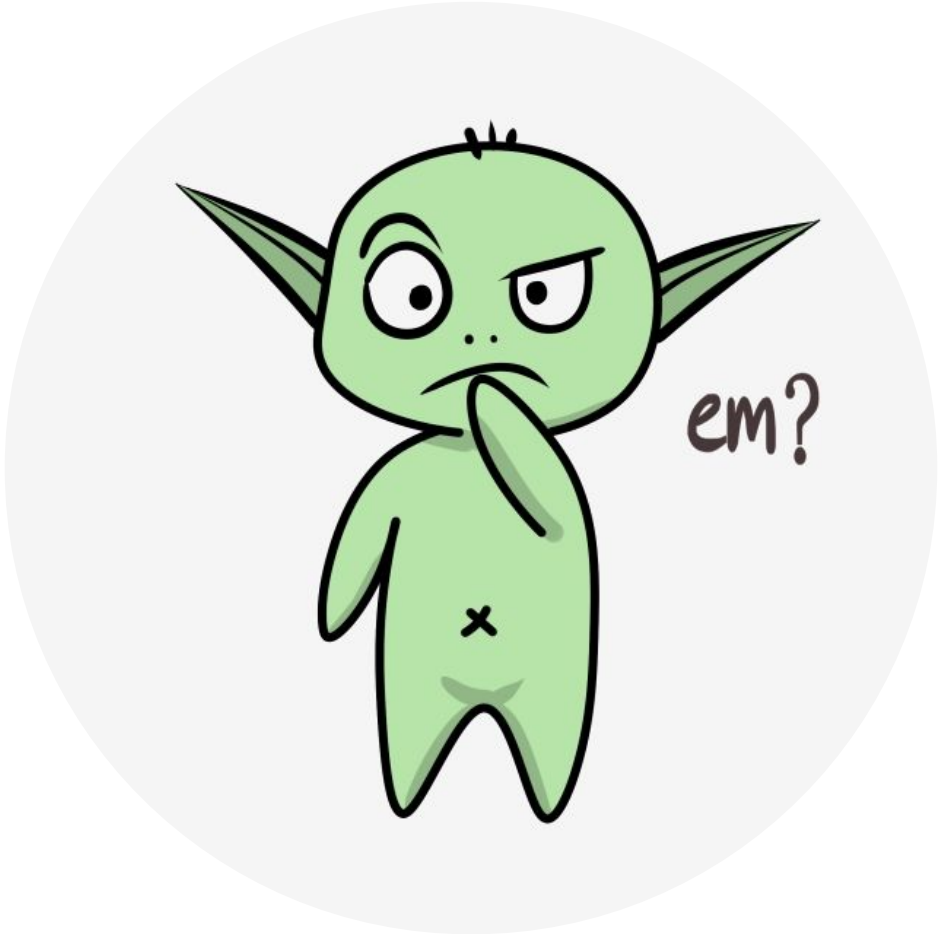
The update rule for Expected Sarsa is on-policy,

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \mathbb{E}_\pi [Q(S_{t+1}, A_{t+1}) | S_{t+1}] - Q(S_t, A_t) \right]$$

when when the target policy π is the same as the behavior policy b that generated $\{S_t, A_t, R_{t+1}, S_{t+1}\}$. However, Expected Sarsa can be made to learn off-policy with different target policies π and behavior policies b.

To learn off-policy in Monte Carlo learning, we need to correct the expectation using importance sampling (IS). Why we not need IS in Expected Sarsa, despite there being an expectation term?

Exercise



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

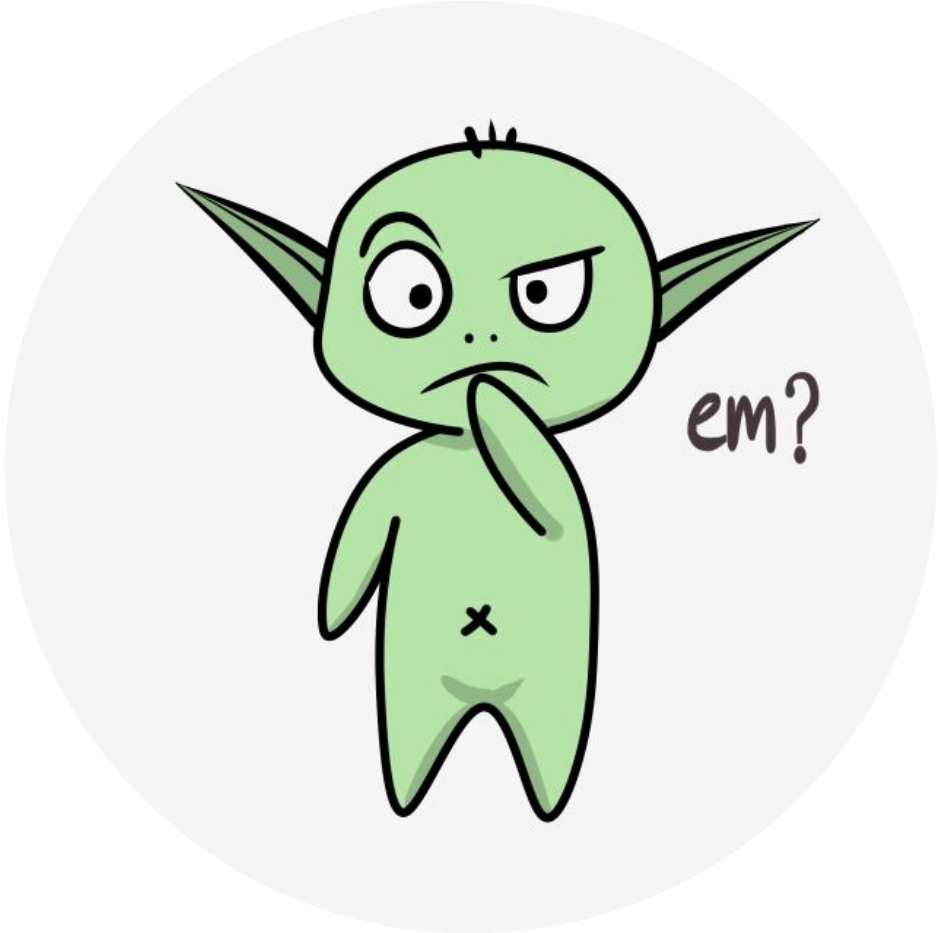
Maximization Bias

- 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:
    $A^* = \text{argmax}_a\ Q_1(a)$ $\quad\quad$ $Q_2(A^*) = Q_2(\text{argmax}_a\ Q_1(a))$
  - This leads to unbiased estimates, that is: $\mathbb{E}[Q_2(A^*)] = q(A^*)$

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



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 $Q(\textit{terminal}, \cdot) = 0$

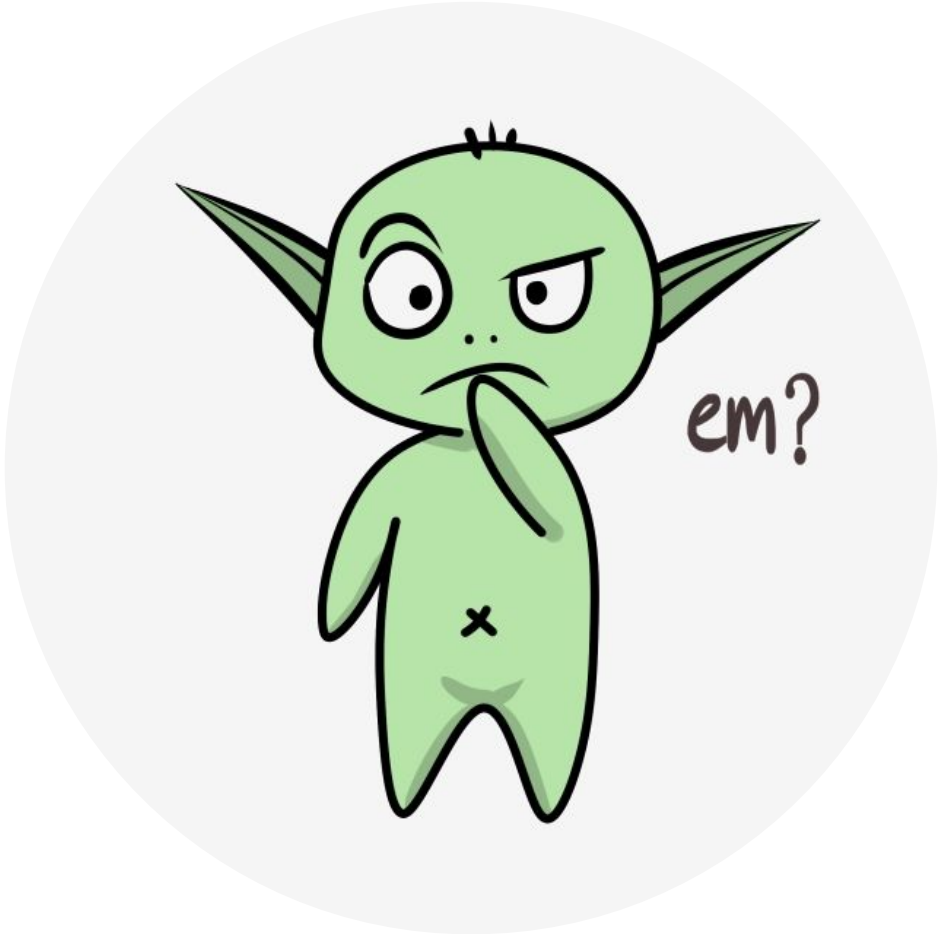
Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using the policy ε-greedy in $Q_1 + Q_2$
    Take action A, observe R, S'
    With 0.5 probabilility:

$$Q_1(S, A) \leftarrow Q_1(S, A) + \alpha\Big(R + \gamma Q_2\big(S', \operatorname{argmax}_a Q_1(S', a)\big) - Q_1(S, A)\Big)$$

    else:

$$Q_2(S, A) \leftarrow Q_2(S, A) + \alpha\Big(R + \gamma Q_1\big(S', \operatorname{argmax}_a Q_2(S', a)\big) - Q_2(S, A)\Big)$$

    $S \leftarrow S'$
  until S is terminal



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

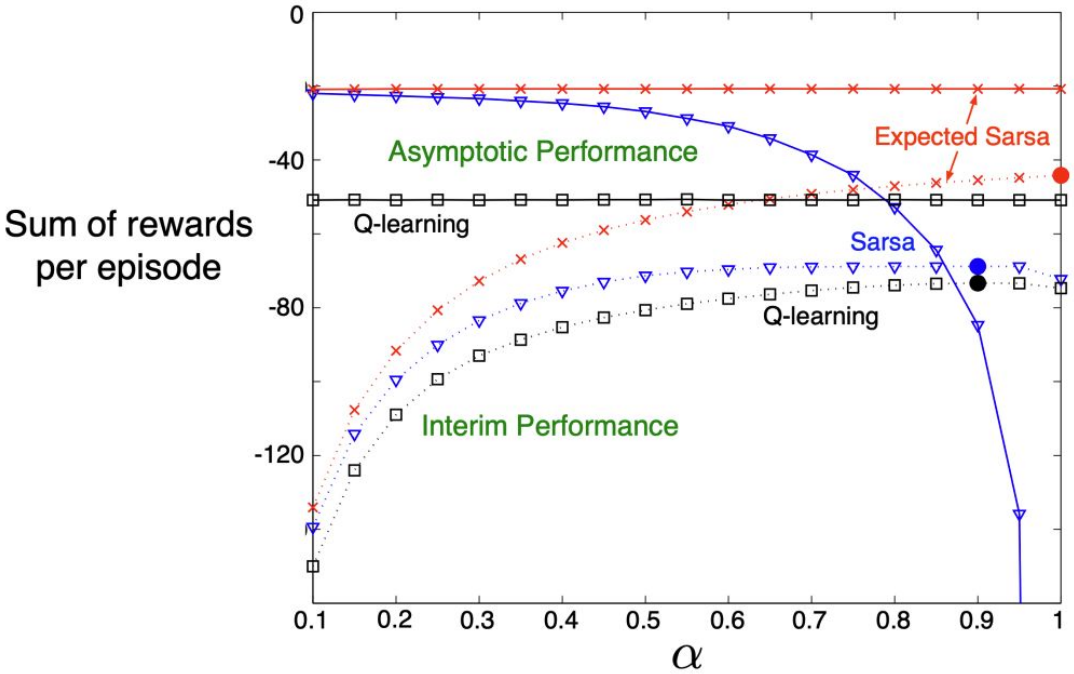
Choosing to Choose your RL Algorithm?

- What's the nature of your task, episodic or continuing?
- Do you have access to the model, p(s', r | s, a)?
- Is changing the policy during an episode important for good performance?
- Will my initial estimate of the value function, Q_0, be really terrible?
- Do I need the optimal policy or near-optimal is good enough?
- *Are we measuring performance online or offline?*

The Devil is in the Details

- Let’s say we decide to go with Q-Learning.
- How should we:
  - Decide on the target policy?
  - Decide on the behaviour policy?
  - How should we initialize Q_0?
  - Set ε? Set α? Should they change over time?
- What do we care about? The area under the curve or just the final policy?
- How long will you run it for?

The Devil is in the Details

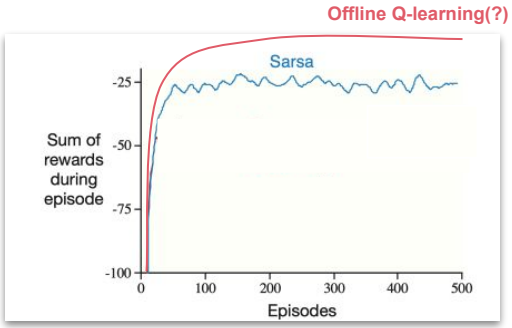


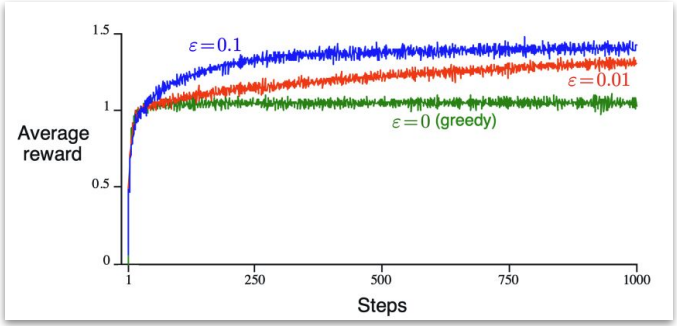
Each point on this plot represents a different choice of:

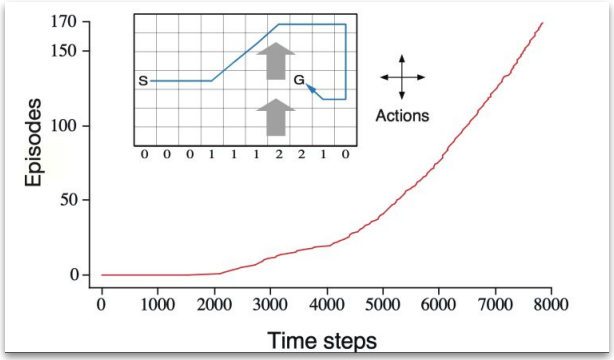
1. **Target policy.**
2. **α.**
3. **How long it was run for.**

Online vs Offline Performance

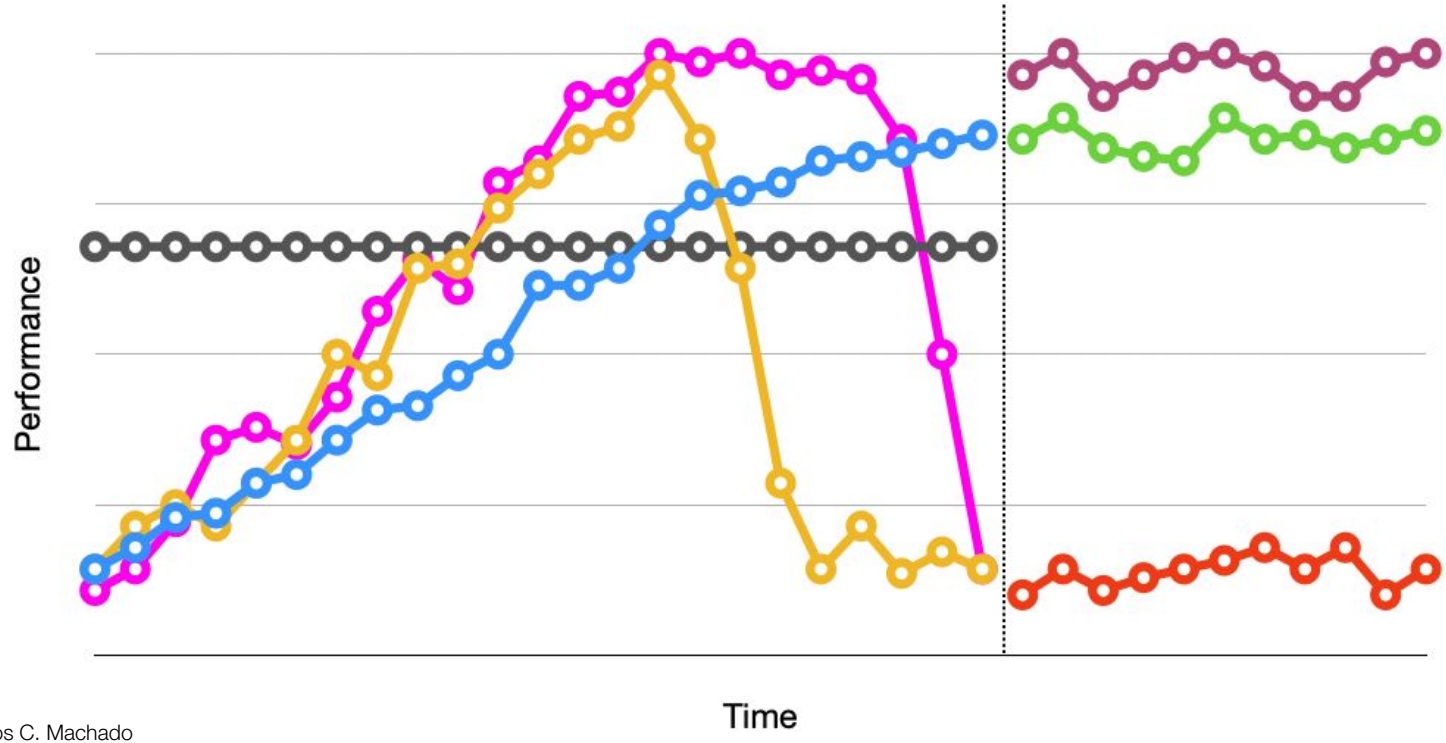
- Online is what we have been doing all along.
- In offline evaluation we have two phases:
  - **Learning** (updating the value function) and taking actions, with no performance evaluation.
  - **Testing**, when learning is disabled and we evaluate the current policy π.



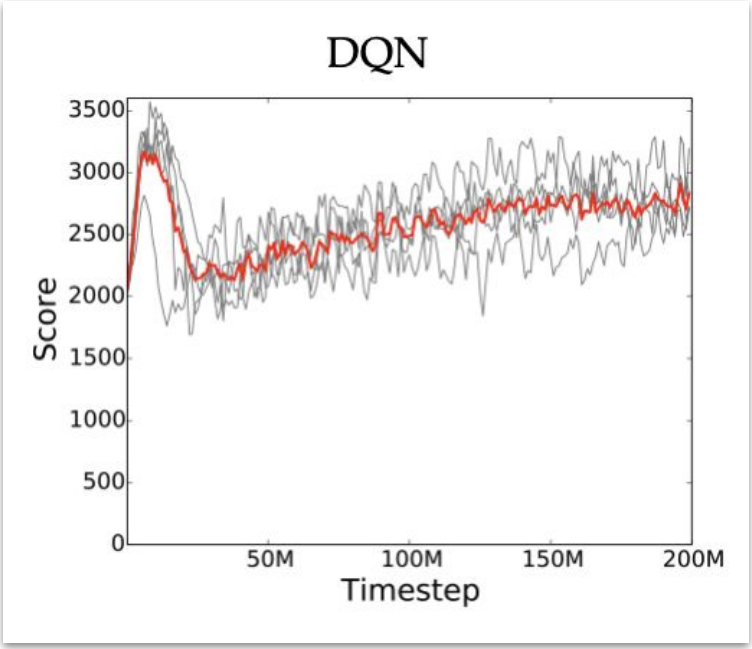




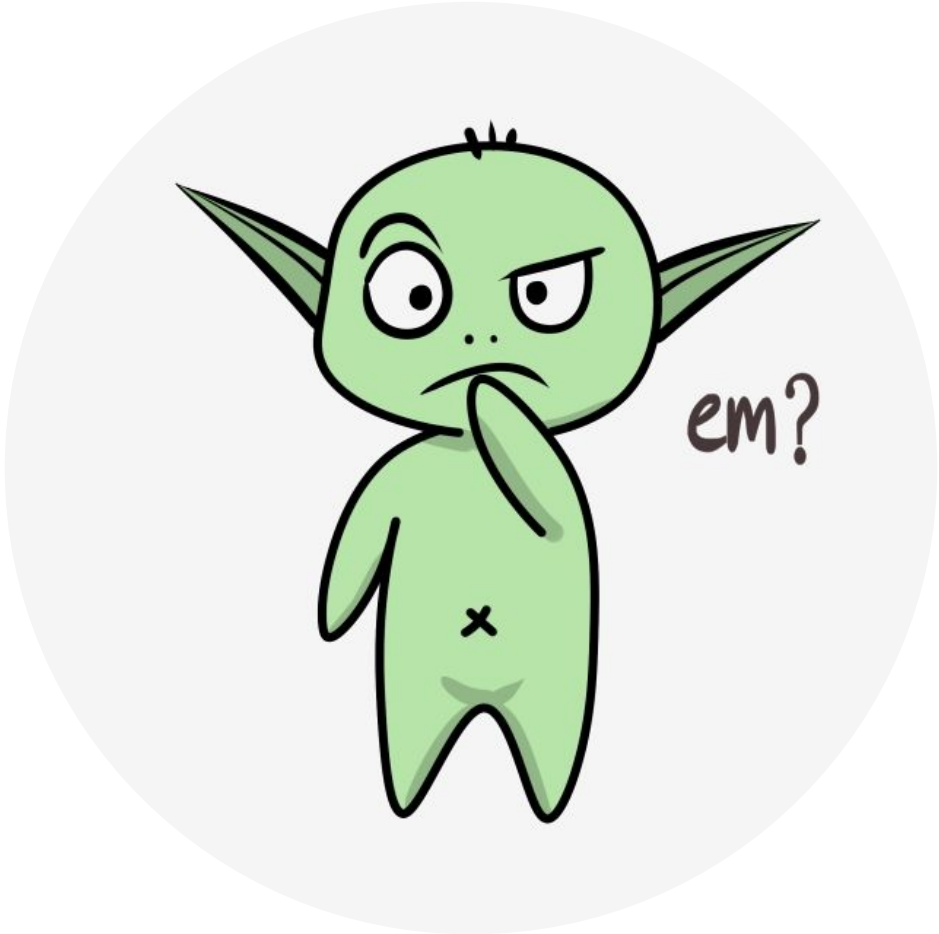
Which one do you prefer?



It does happen!



[Machado et al., 2018]



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