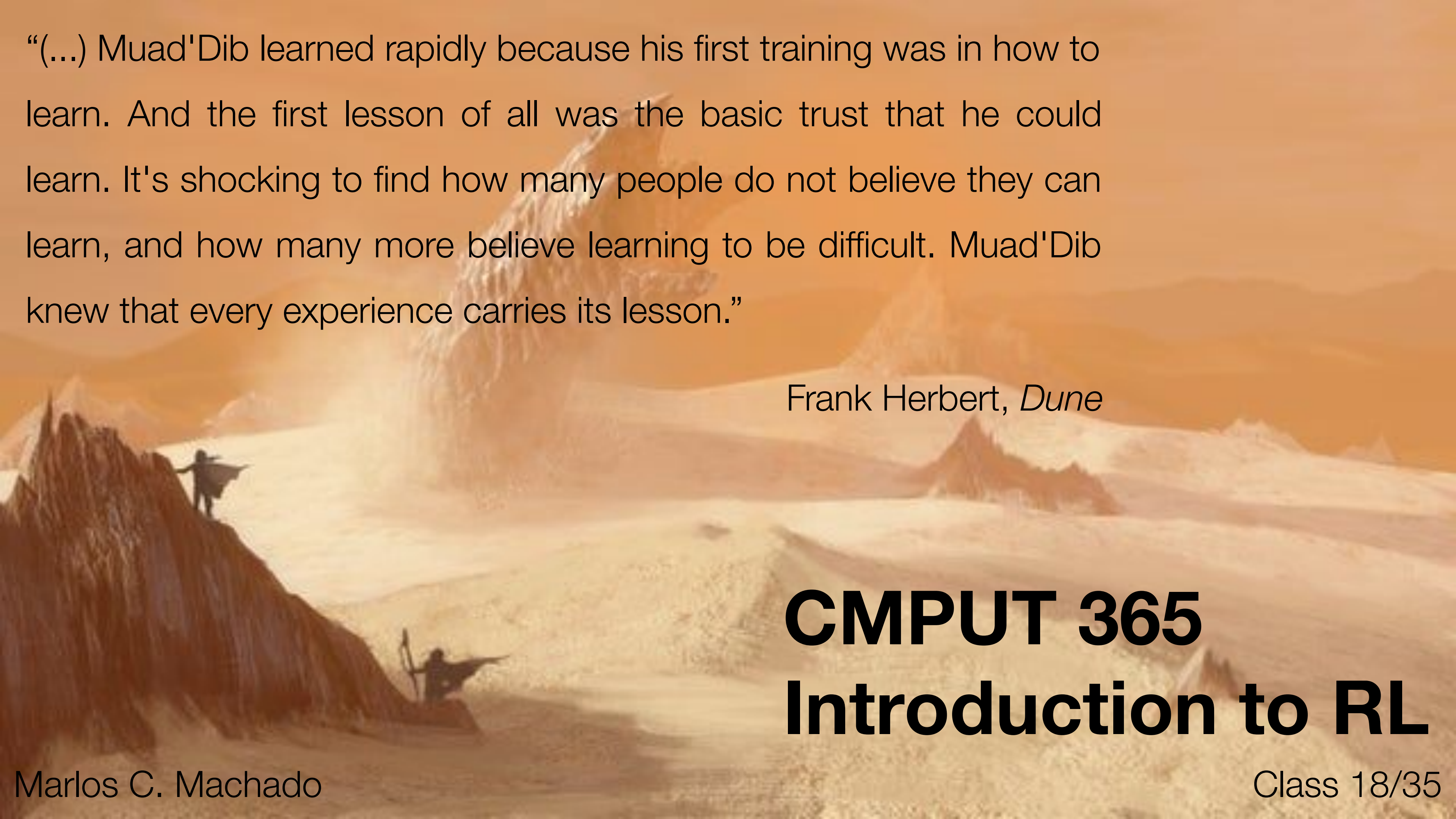


> “(...) Muad'Dib learned rapidly because his first training was in how to learn. And the first lesson of all was the basic trust that he could learn. It's shocking to find how many people do not believe they can learn, and how many more believe learning to be difficult. Muad'Dib knew that every experience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

CMPUT 365 Introduction to RL

Marlos C. Machado

Class 18/35

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.

There is still **1 pending invitation** for a student who is still enrolled in the course.

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`.

Reminder II

- Midterm marks are now available on eClass.
  - Average: 10.4; Median: 11; Std. Dev.: 3.2.
  - Exam viewing on Tuesday and Wednesday next week.

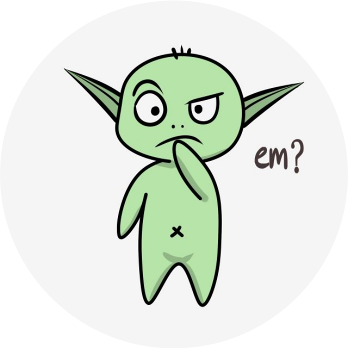
- **I want your feedback!**
  - Mid-term Course and Instruction Feedback online evaluation opened today.
  - It will close today.
  - 17 of 90 students responded so far 😭

Plan / Reminder III

- What **I** plan to do today:
  - Talk about Temporal-Difference Learning for Prediction (Beginning of Chapter 6 of the textbook).

- What I recommend **YOU** to do for Monday:
  - Read Chapter 6 up to Section 6.3.
  - Programming assignment (Policy evaluation with TD learning).

Please, interrupt me at any time!



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

Chapter 6

Temporal-Difference Learning

Prediction

Temporal-difference learning – Why?

“If one had to identify one idea as central and novel to reinforcement learning, it would undoubtedly be temporal-difference (TD) learning.”

TD Prediction

A simple every-visit Monte Carlo method is:

$$V(S_t) \leftarrow V(S_t) + \alpha\Big[\underline{G_t} - V(S_t)\Big]$$

What if we don't want to wait until we have a full return (end of episode)!

$$NewEstimate \leftarrow OldEstimate \; + \; StepSize \; \Big[Target - OldEstimate\Big]$$

TD Prediction

A simple every-visit Monte Carlo method is:

$$V(S_t) \leftarrow V(S_t) + \alpha\Big[\underbrace{G_t}_{\text{Target}} - V(S_t)\Big]$$

Temporal-Difference Learning:

$$V(S_t) \leftarrow V(S_t) + \alpha\Big[\underbrace{R_{t+1} + \gamma V(S_{t+1})}_{\text{Target}} - V(S_t)\Big]$$

TD Prediction

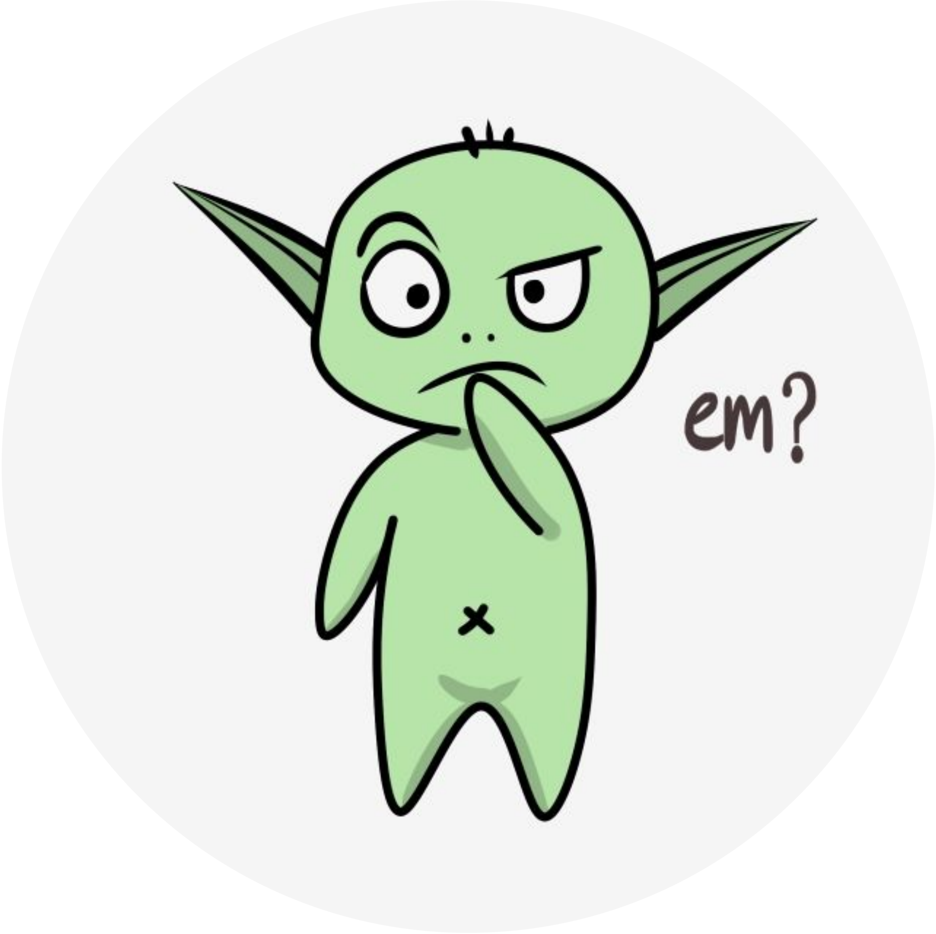
A simple every-visit Monte Carlo method is:

$$V(S_t) \leftarrow V(S_t) + \alpha\Big[G_t - V(S_t)\Big]$$

Temporal-Difference Learning (specifically, **one-step TD**, or **TD(0)**):

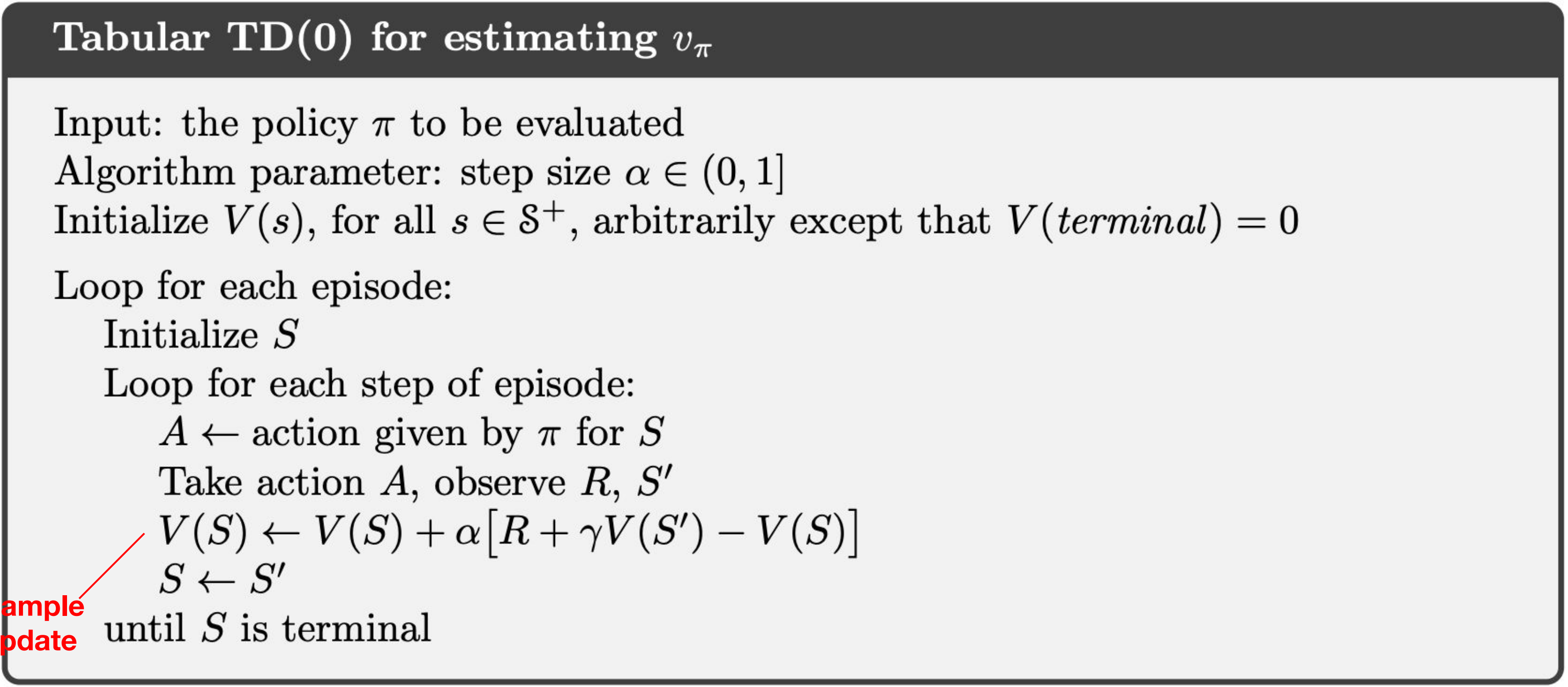
$$V(S_t) \leftarrow V(S_t) + \alpha\Big[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\Big]$$

These are estimates all the way down…



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

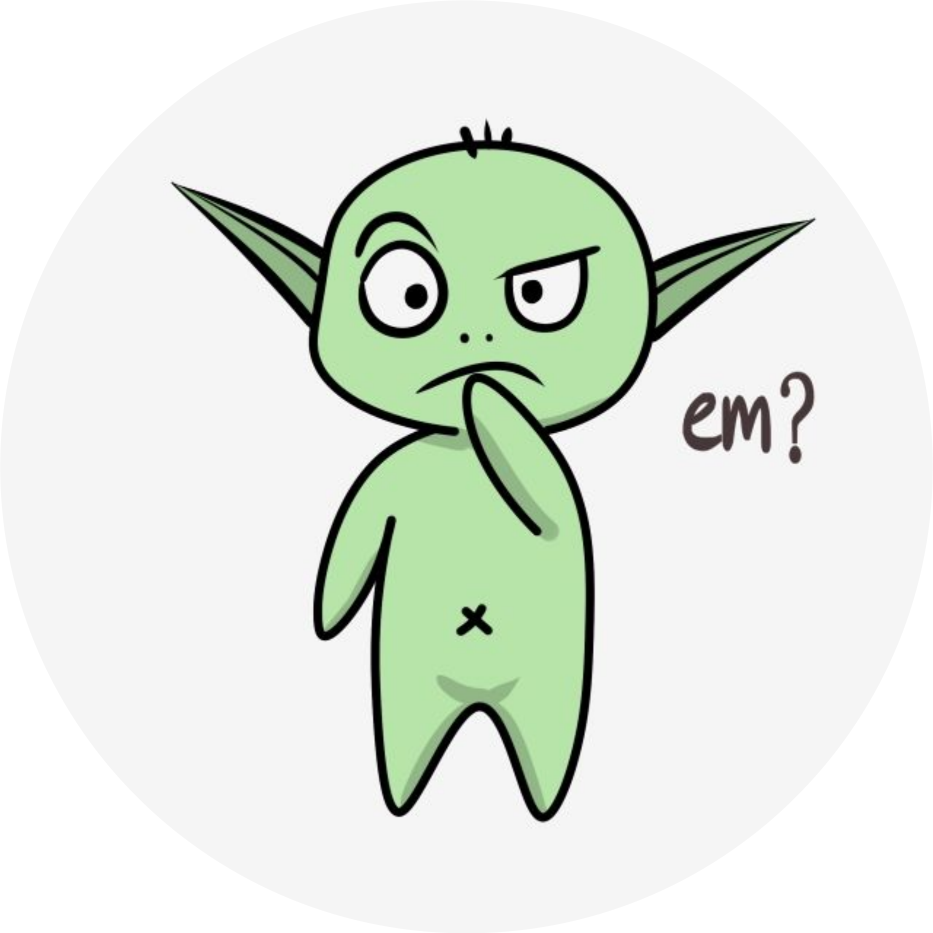
Tabular TD(0)



Tabular TD(0) for estimating v_π

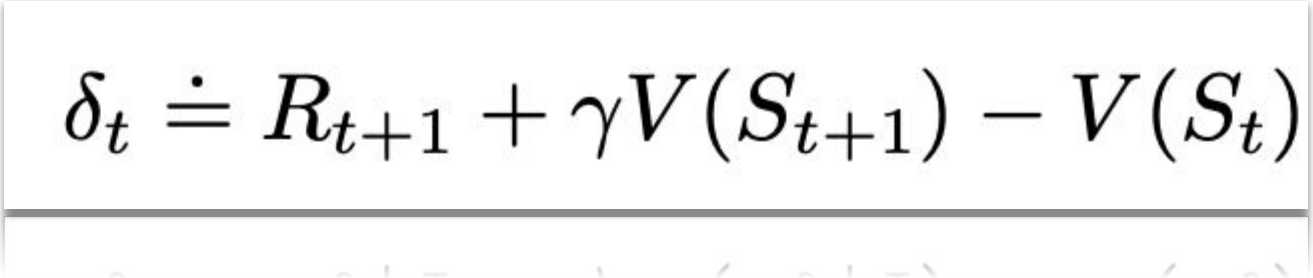
```
Input: the policy $\pi$ to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(\textit{terminal}) = 0$

Loop for each episode:
    Initialize $S$
    Loop for each step of episode:
        $A \leftarrow$ action given by $\pi$ for $S$
        Take action $A$, observe $R$, $S'$
        $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
        $S \leftarrow S'$
    until $S$ is terminal
```

Sample update



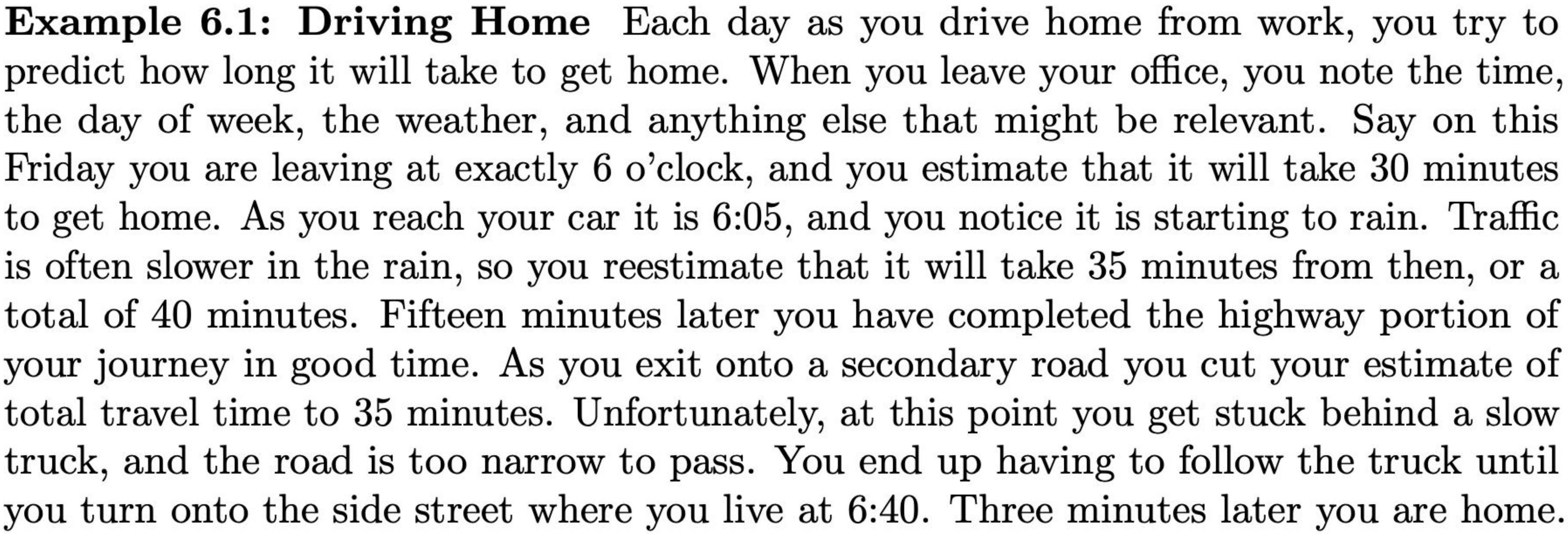
https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Temporal-Difference Error



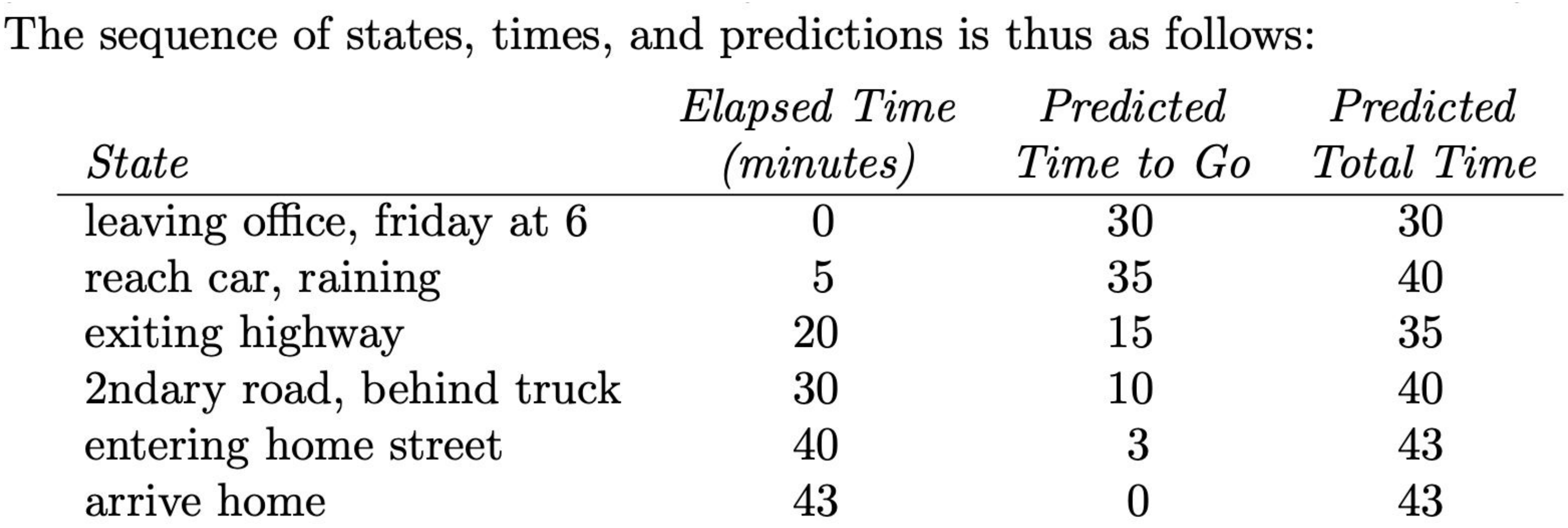
$$\delta_t \doteq R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$$

Example – Driving Home



Example 6.1: Driving Home Each day as you drive home from work, you try to predict how long it will take to get home. When you leave your office, you note the time, the day of week, the weather, and anything else that might be relevant. Say on this Friday you are leaving at exactly 6 o'clock, and you estimate that it will take 30 minutes to get home. As you reach your car it is 6:05, and you notice it is starting to rain. Traffic is often slower in the rain, so you reestimate that it will take 35 minutes from then, or a total of 40 minutes. Fifteen minutes later you have completed the highway portion of your journey in good time. As you exit onto a secondary road you cut your estimate of total travel time to 35 minutes. Unfortunately, at this point you get stuck behind a slow truck, and the road is too narrow to pass. You end up having to follow the truck until you turn onto the side street where you live at 6:40. Three minutes later you are home.

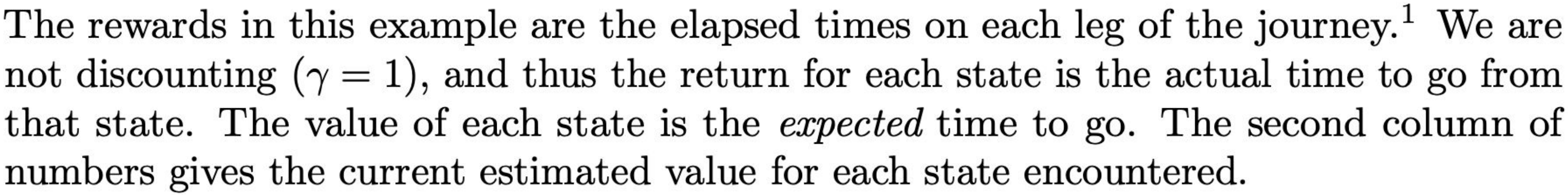
Example – Driving Home



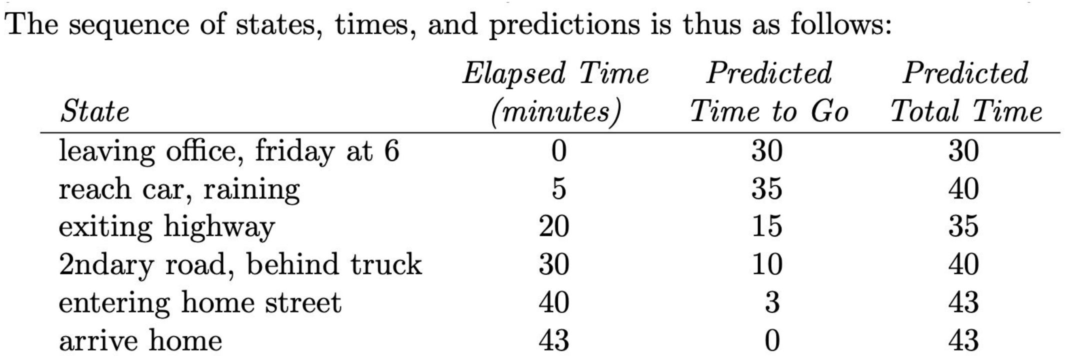
The sequence of states, times, and predictions is thus as follows:

State	*Elapsed Time (minutes)*	*Predicted Time to Go*	*Predicted Total Time*
leaving office, friday at 6	0	30	30
reach car, raining	5	35	40
exiting highway	20	15	35
2ndary road, behind truck	30	10	40
entering home street	40	3	43
arrive home	43	0	43

Example – Driving Home



The rewards in this example are the elapsed times on each leg of the journey.[1] We are not discounting ($\gamma = 1$), and thus the return for each state is the actual time to go from that state. The value of each state is the *expected* time to go. The second column of numbers gives the current estimated value for each state encountered.



The sequence of states, times, and predictions is thus as follows:

State	*Elapsed Time (minutes)*	*Predicted Time to Go*	*Predicted Total Time*
leaving office, friday at 6	0	30	30
reach car, raining	5	35	40
exiting highway	20	15	35
2ndary road, behind truck	30	10	40
entering home street	40	3	43
arrive home	43	0	43

Example – Driving Home

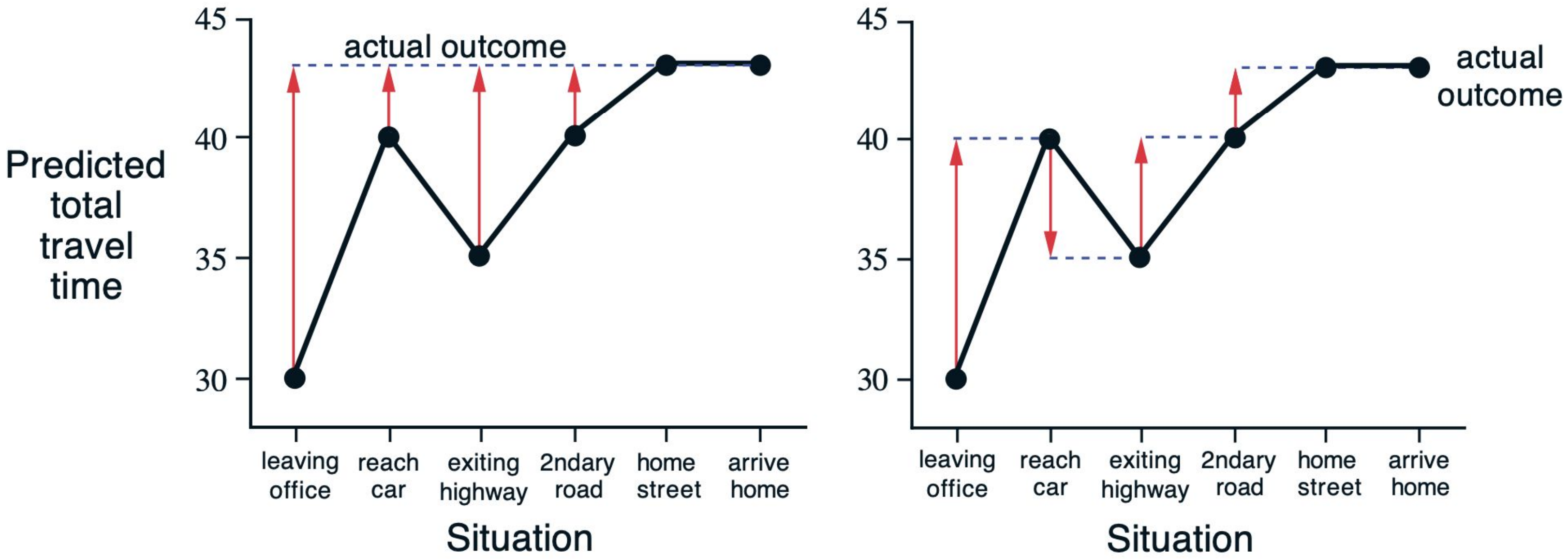
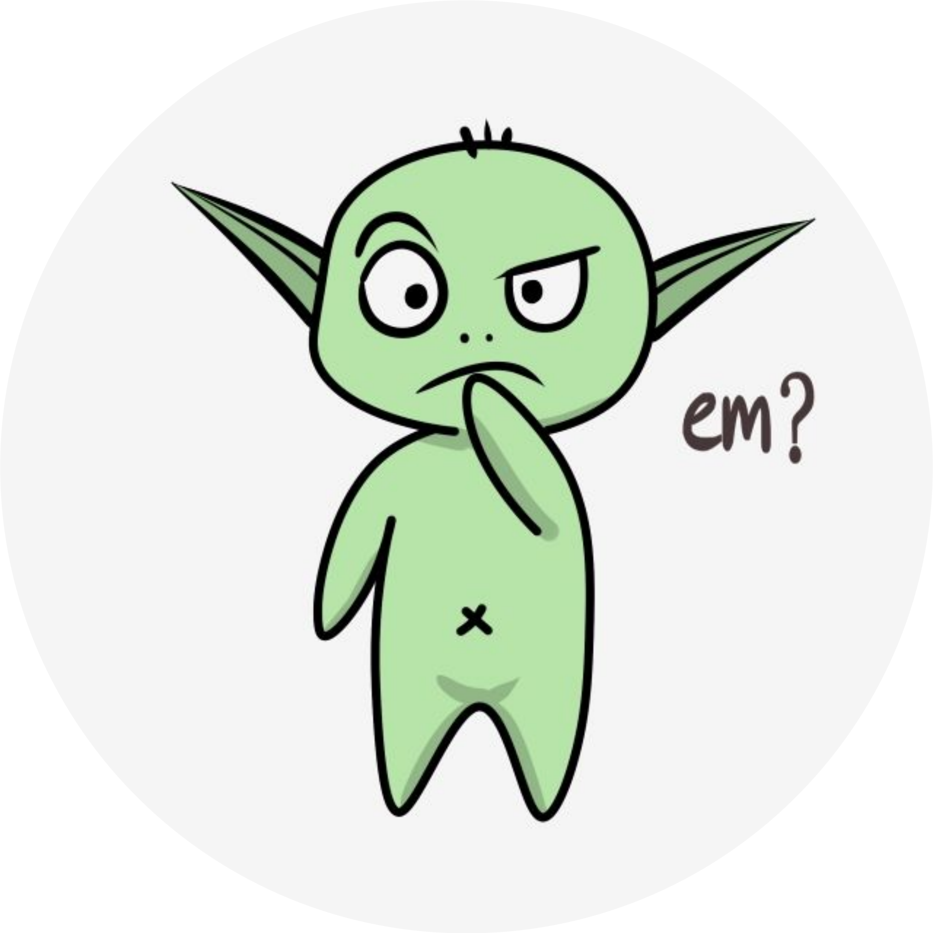


Figure 6.1: Changes recommended in the driving home example by Monte Carlo methods (left) and TD methods (right).



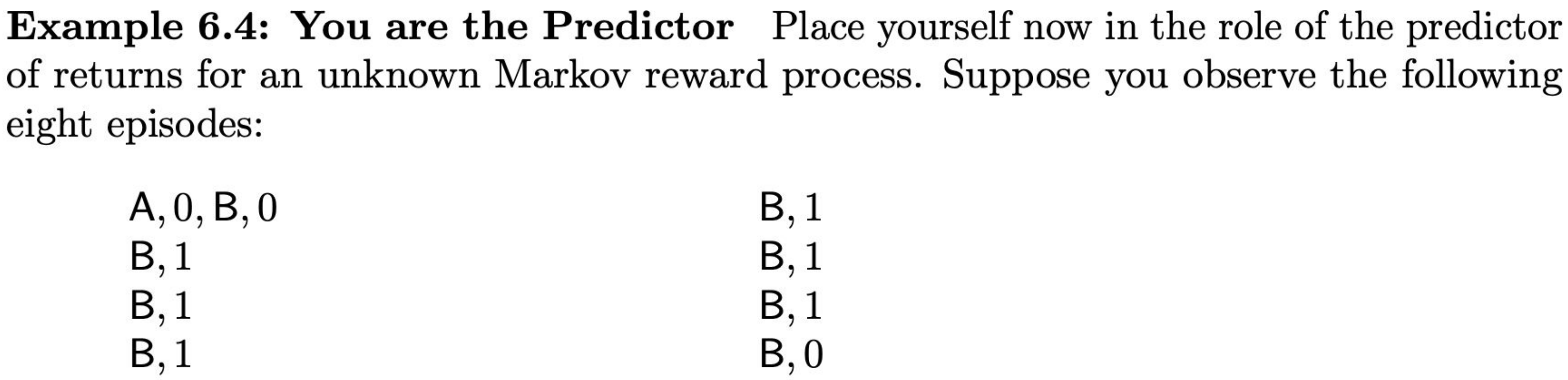
https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Optimality of TD(0)

- Under batch training, constant-α MC converges to values, V(s), that are sample averages of the actual returns experienced after visiting each state s. These are optimal estimates in the sense that they minimize the mean square error from the actual returns in the training set.

- Bath TD(0) gives us the answer that it is based on first modeling the Markov process and then computing the correct estimates given the model (the *certainty-equivalence estimate*).

Example

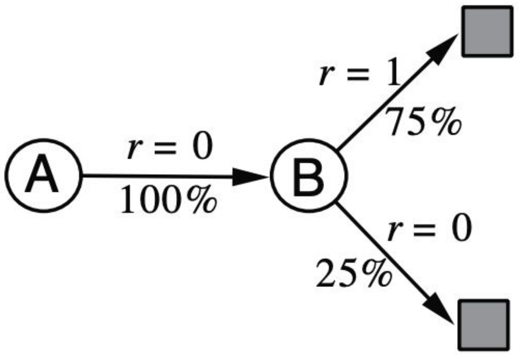


Example 6.4: You are the Predictor Place yourself now in the role of the predictor of returns for an unknown Markov reward process. Suppose you observe the following eight episodes:

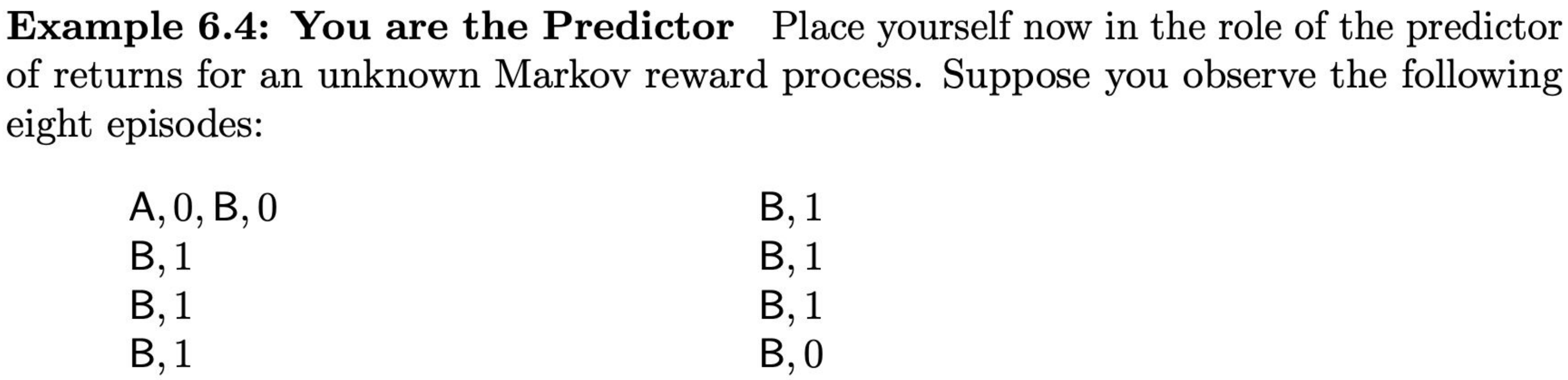
$\mathsf{A}, 0, \mathsf{B}, 0$
$\mathsf{B}, 1$
$\mathsf{B}, 1$
$\mathsf{B}, 1$
$\mathsf{B}, 1$
$\mathsf{B}, 1$
$\mathsf{B}, 1$
$\mathsf{B}, 0$

V(A) = ?

V(B) = ?



Example



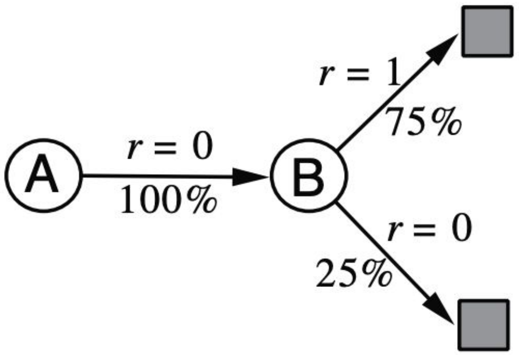
Example 6.4: You are the Predictor Place yourself now in the role of the predictor of returns for an unknown Markov reward process. Suppose you observe the following eight episodes:

A, 0, B, 0
B, 1
B, 1
B, 1
B, 1
B, 1
B, 1
B, 0

TD **MC**

V(A) = ? ¾ or 0?

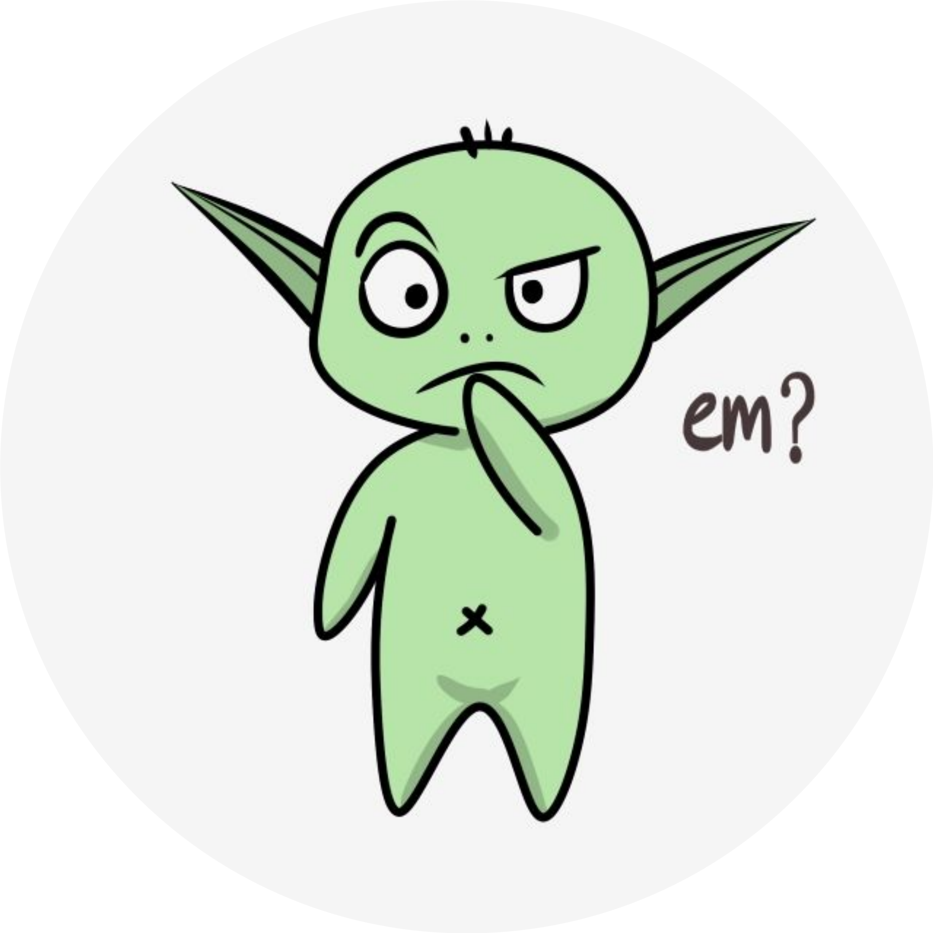
V(B) = ¾



TD vs Monte Carlo

“Batch Monte Carlo methods always find the estimates that minimize mean square error on the training set, whereas batch TD(0) always finds the estimates that would be exactly correct for the maximum-likelihood model of the Markov process.”

In general, the *maximum-likelihood estimate* of a parameter is the parameter value whose probability of generating the data is greatest.



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