

“(...) 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**

# 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`.

# Reminder II

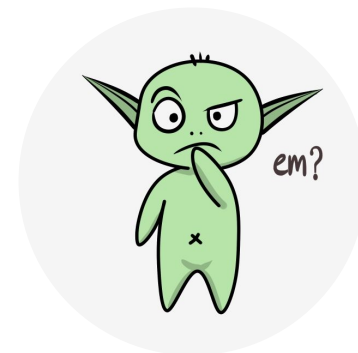
- Exam viewing on Tuesday and Wednesday
  - Tuesday: 2pm - 5pm in ATH 3-28
  - Wednesday: 3pm - 5pm in ATH 3-32

**Re-evaluation of midterm exams:** Students will have access to their midterm exam during an exam viewing period. A student who has concerns about how specific questions of their midterm exam were marked can submit a request to the instructor, via email, within two weeks the date they received their marked exam. The request should specify (1) which question is to be re-evaluated, (2) the rationale for such a request, and (3) the proposed marks. Importantly, once a request for re-evaluation is submitted, it is up to the instructor's discretion to adjust the marks. *Students won't be allowed to take their midterm exams with them, nor to take pictures of them, so in case of concerns the student is advised to take notes during the exam viewing period. The TAs are not authorized to weigh in on the midterm exams, this is something only the instructor can do. Notice marks can also go down once a question is re-evaluated.*

## Plan / Reminder III

- What **I** plan to do today:
  - Continue talking about TD Learning for Prediction (Beginning of Chapter 6 of the textbook).
- **Due today** (Monday):
  - Programming assignment (Policy evaluation with TD learning).
- I still recommend you to read Chapter 6 of the textbook.

# Please, interrupt me at any time!



## Last Class: Temporal-Difference Error

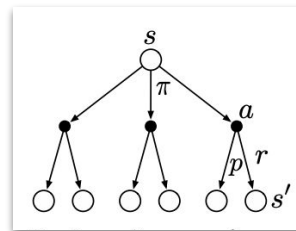
$$V(S_t) \leftarrow V(S_t) + \alpha \left[ R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right]$$

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

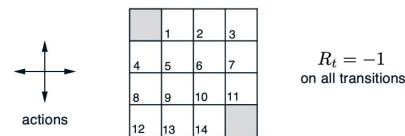
# TD is a sample update with bootstrapping

- Dynamic programming update:

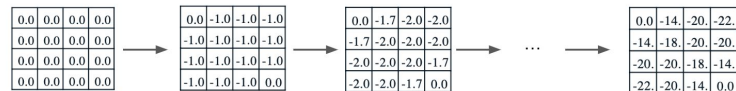
$$\begin{aligned}
 v_{k+1}(s) &\doteq \mathbb{E}_{\pi}[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\
 &= \sum_a \pi(a|s) \sum_{s', r} p(s', r | s, a) [r + \gamma v_k(s')]
 \end{aligned}$$



## Policy Evaluation – Example



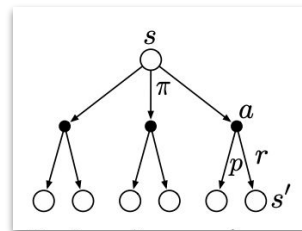
$v_k$  for the random policy



# TD is a sample update with bootstrapping

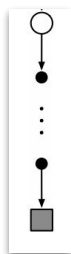
- Dynamic programming update:

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



- Monte Carlo update:

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

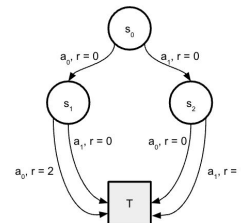


16

CMPUT 365 – Class 17/35

## Practice Exercise 1

Consider the three-state MDP below with terminal state T and  $\gamma = 1$ . Suppose you observe three episodes:  $\{s_0, s_1, T\}$  with a return of 2,  $\{s_0, s_1, T\}$  with a return of 2,  $\{s_0, s_2, T\}$  with a return of 1. What is the (every-visit) Monte-Carlo estimator of the value for each of the states,  $s_0, s_1, s_2$ ? How would the Monte-Carlo estimates change if  $r(s_0, a_1, s_2) = 1$ ?



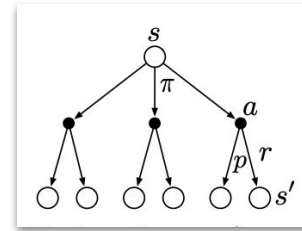
Marlos C. Machado



# TD is a sample update with bootstrapping

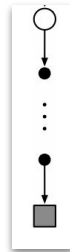
- Dynamic programming update:

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



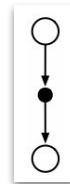
- Monte Carlo update:

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



- TD update:

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





## 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:

<i>State</i>	<i>Elapsed Time (minutes)</i>	<i>Predicted Time to Go</i>	<i>Predicted Total Time</i>
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

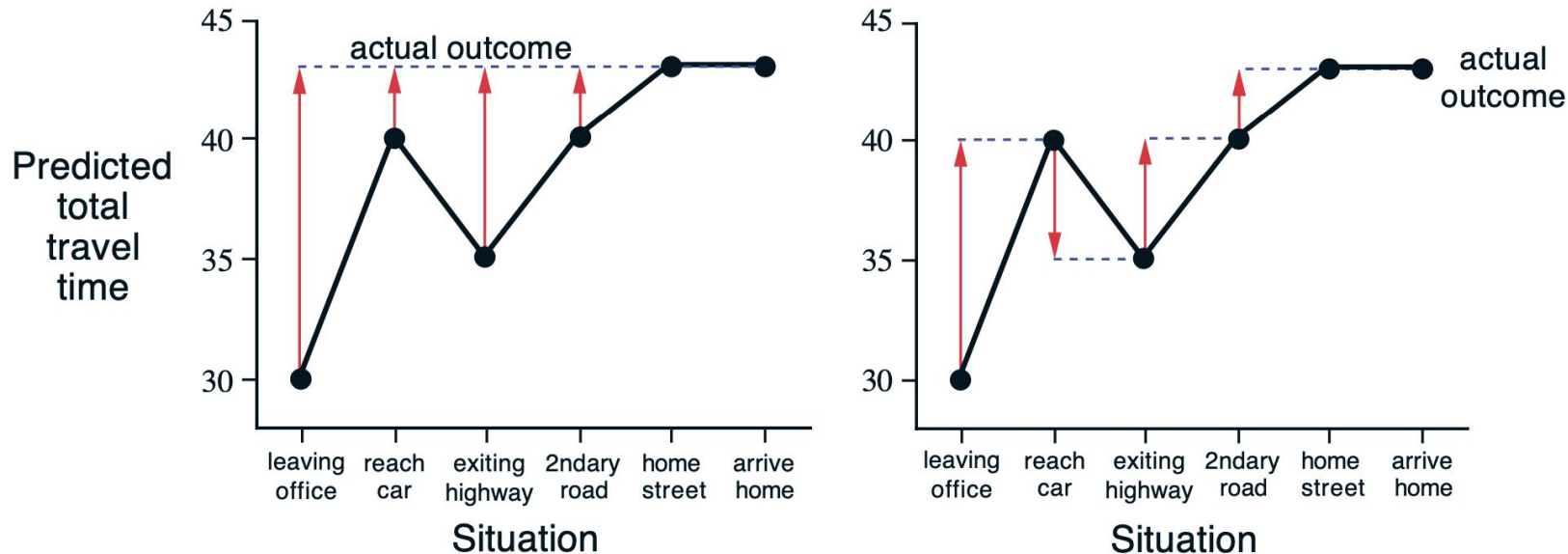
## Example – Driving Home

The rewards in this example are the elapsed times on each leg of the journey.<sup>1</sup> 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.

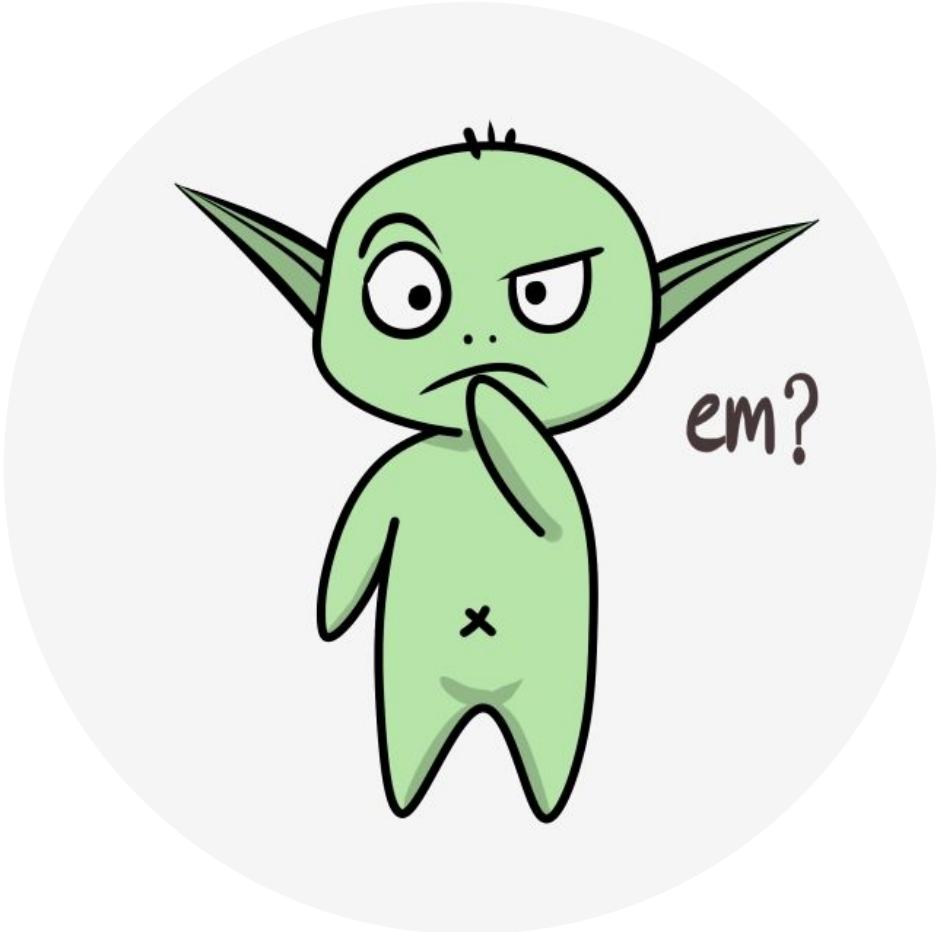
The sequence of states, times, and predictions is thus as follows:

<i>State</i>	<i>Elapsed Time (minutes)</i>	<i>Predicted Time to Go</i>	<i>Predicted Total Time</i>
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).



# Advantages of TD Prediction Methods

- TD methods do not require a model of the environment.
- TD methods are implemented in an online, fully incremental fashion.
- TD(0) is biased towards the initialization while MC methods have higher variance.
- When processing a fixed batch of data, they converge to different points.



## Optimality of TD(0)

- Under batch training, constant- $\alpha$  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:

A, 0, B, 0

B, 1

B, 1

B, 1

B, 1

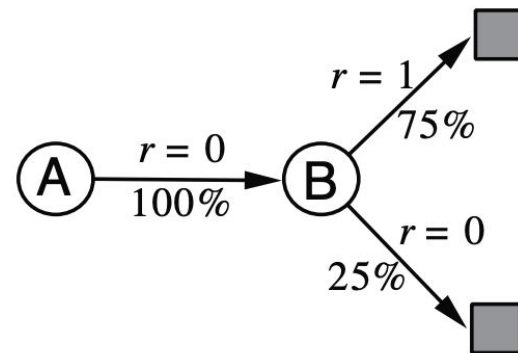
B, 1

B, 1

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

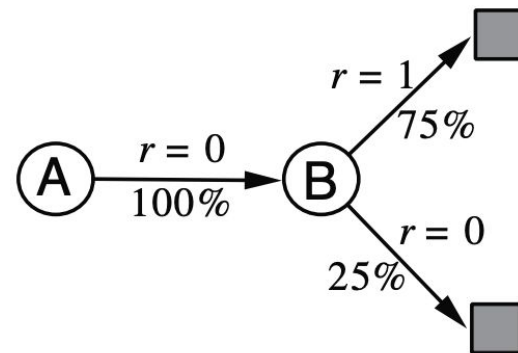
B, 1

B, 1

B, 0

$V(A) = ?$     **TD**    **MC**  
                    $\frac{3}{4}$  or 0?

$V(B) = \frac{3}{4}$



## 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.**

