Reinforcement Learning (RL)

Chapter 5:

Temporal Difference (TD) Learning

Saeed Saeedvand, Ph.D.

Contents

In this Chapter:

- ✓ Temporal Differences (TD) learning algorithm
- ✓ Temporal Differences (1)
- ✓ Temporal Differences (0)
- ✓ Temporal Differences (λ)
- ✓ Introduction to SARSA

Aim of this chapter:

- ✓ Understand concepts of the Temporal Differences (TD) learning algorithms concept and the basic algorithms of TD predictions.

Temporal Differences (TD) learning algorithm

- ✓ **Temporal difference** (TD) learning is a class of model-free reinforcement learning.
- ✓ The **name TD** derives from its use of differences, or changes of value on different time steps
- ✓ The **differences** is to be **used in the learning process** (based on reward signal).

Temporal Differences (TD) learning algorithm

✓ TD learning is a **combination** of Monte Carlo and dynamic programming ideas:

- **Like Monte Carlo** methods, TD methods **can learn directly from raw experience without a model of the environment's** dynamics.
- **Like DP,** TD methods **update estimates based in part on other learned estimates**, without waiting for a final outcome (bootstrapping).

What is the relationship between TD, DP, and MC methods?

✓ We are using theme in the theory of reinforcement learning.

Temporal Differences (TD) learning algorithm

General differences between TD, and MC methods

Monte-Carlo

- ✓ MC learns from experience.
- ✓ MC must **wait until end of episode** with complete sequences (to get the return)
- ✓ MC **only works for episodic** environments.

Temporal Differences

- ✓ TD learns from experience (can be limited).
- ✓ TD can **learn before knowing the final outcome** (learn online).
- ✓ TD **can work in** environments with infinite horizon.

Temporal Differences (TD) learning algorithm

Temporal-Difference Learning algorithms

TD Prediction

- ✓ Temporal Differences (1) → TD(1)
- ✓ Temporal Differences (0) → TD(0)
- ✓ Temporal Differences (λ) → TD(λ)

TD Control

- ✓ SARSA: On-policy TD Control
- ✓ Q-learning: Off-policy TD Control
- ✓ Expected SARSA (both on-policy or off-policy TD Control)

Temporal Differences (TD) algorithm (Prediction)

Temporal Differences TD(1):

- ✓ TD(1) is for estimating V_π
- ✓ Makes an **update** to the values **at the end of an episode** but not limited to
- ✓ Therefore we can update if we stop for any reason or use it **in continuing tasks**
- ✓ **Similar to the Monte Carlo** with using the experience to solve the prediction problem but can be online with n-steps

Temporal Differences (TD) algorithm (Prediction)

Temporal Differences TD(1):

- ✓ TD(1) is known as a way of MC implementation
- ✓ It is similar to the MC approaches because of it uses G_t

$$G_t = r_t + \gamma^1 r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{T-1} r_T$$

The return following time t

Discounted sum of all the rewards seen in the episode (T steps).

Note: we're only trying to predict the values for the non-terminal states

Temporal Differences (TD) algorithm (Prediction)

Temporal Differences TD(1):

$$G_t = r_t + \gamma^1 r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{T-1} r_T$$

✓ Use the sum of discounted rewards G_t, (from whole episode) and subtract from the prior estimate (We call it the **TD Error**):

Learning rate (also known as step-size, or Error update that can be constant or variable)

TD error

$$V(s_t) = V(s_t) + \alpha(G_t - V(s_t))$$

updated estimate

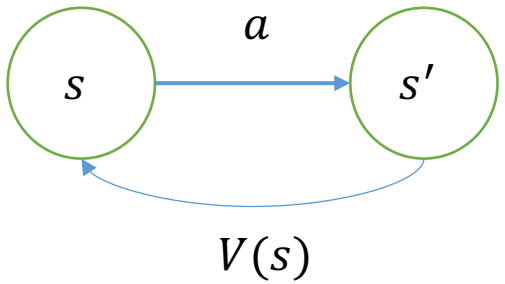
previous estimate

Temporal Differences (TD) algorithm (Prediction)

Temporal Differences TD(0):

Idea:

- ✓ TD(0) is for estimating V_π
- ✓ In TD (0) algorithm in **we do not need to wait until the end of the episode** to update Q-value (in contrast to the Monte Carlo approaches)
- ✓ In TD (0) algorithm we can apply an action and move to the next step then update the value



Update value of the previous state

Note: TD(0) also is known as **one-step TD**

Temporal Differences (TD) algorithm (Prediction)

Temporal Differences TD(0):

$$\text{TD(1)} \rightarrow V(s_t) = V(s_t) + \alpha(G_t - V(s_t))$$

✓ The update rule for value at Temporal Differences (TD (0)):

$$V(s) = V(s) + \alpha[r(s') + \gamma V(s') - V(s)]$$

Difference:
only 1 step ahead

Immediately on transition to s' or s_{t+1} and receiving $r(s')$ or r_{t+1}.

Temporal Differences (TD) algorithm (Prediction)

Temporal Differences TD(0):

$$V(s) = V(s) + \alpha[r(s') + \gamma V(s') - V(s)]$$

Input: the policy π that we want to evaluate
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize arbitrarily $V(s)$ for all states $s \in S$, except that $V(terminal) = 0$
Loop for each episode:
    Initialize s
    Loop for each step of episode:
        a = action given by policy π at s
        Take action a, and observe $r,\ s_{t+1}$
        $V(s) = V(s) + \alpha[r(s_{t+1}) + \gamma V(s_{t+1}) - V(s)]$
        $s = s_{t+1}$
    until s is terminal

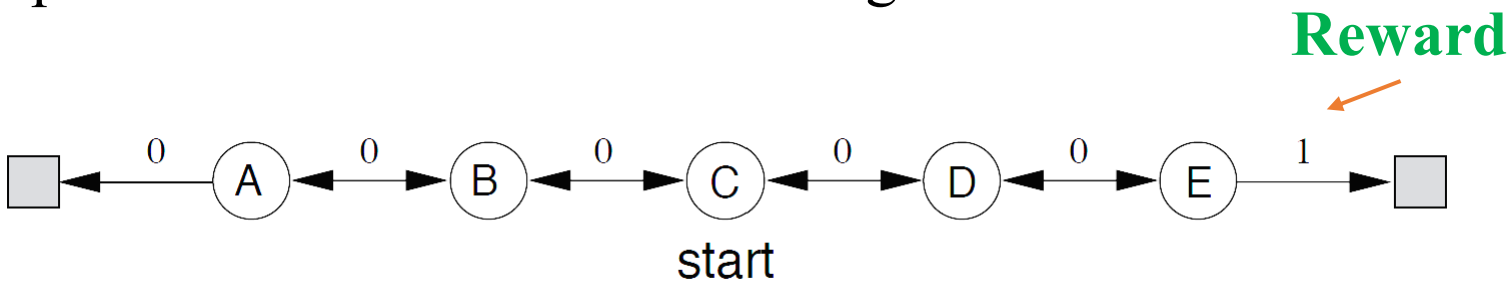
Temporal Differences (TD) algorithm (Prediction)

Temporal Differences TD(0):

Example

❖ Consider Random Walk example:

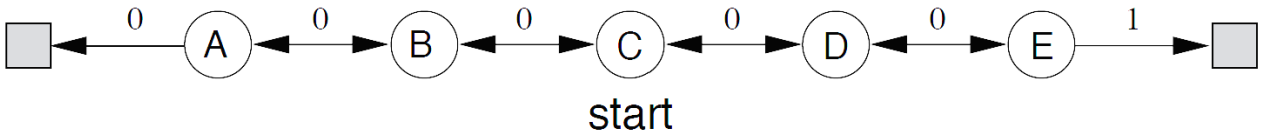
- We have two actions left or right (with equal probability)
- Episodes terminate in the left or right ends



We want to see empirical comparison of the prediction abilities of TD(0) and MC.

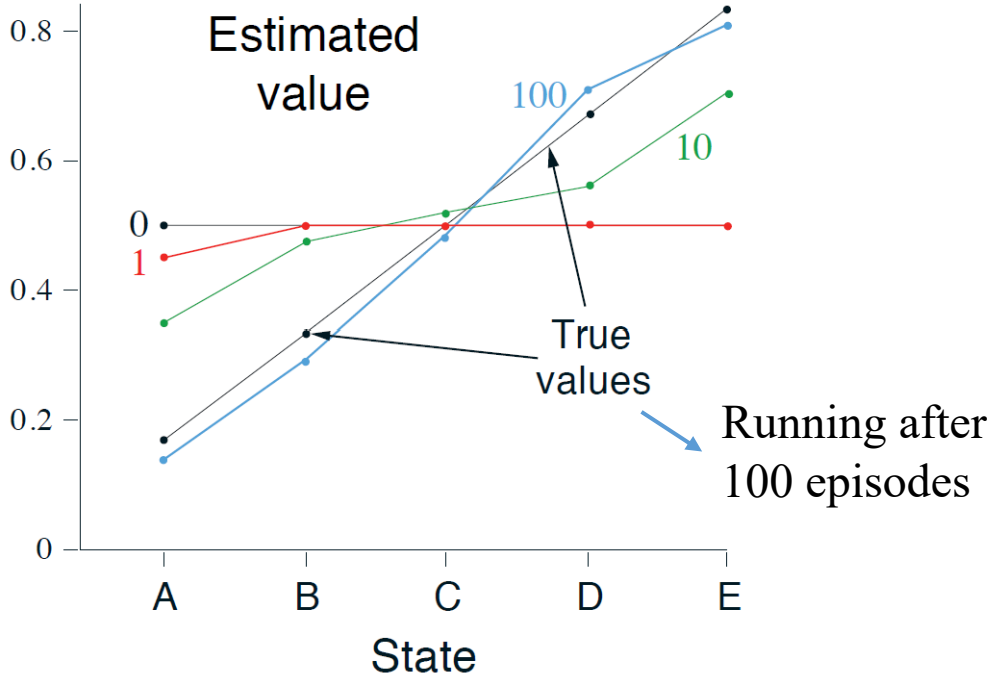
Temporal Differences (TD) algorithm (Prediction)

Temporal Differences TD(0):



Example

Values V(s) learned after running **numbers of episodes** of TD(0).



constant step-size ($\alpha = 0.1$)

Temporal Differences (TD) algorithm (Prediction)

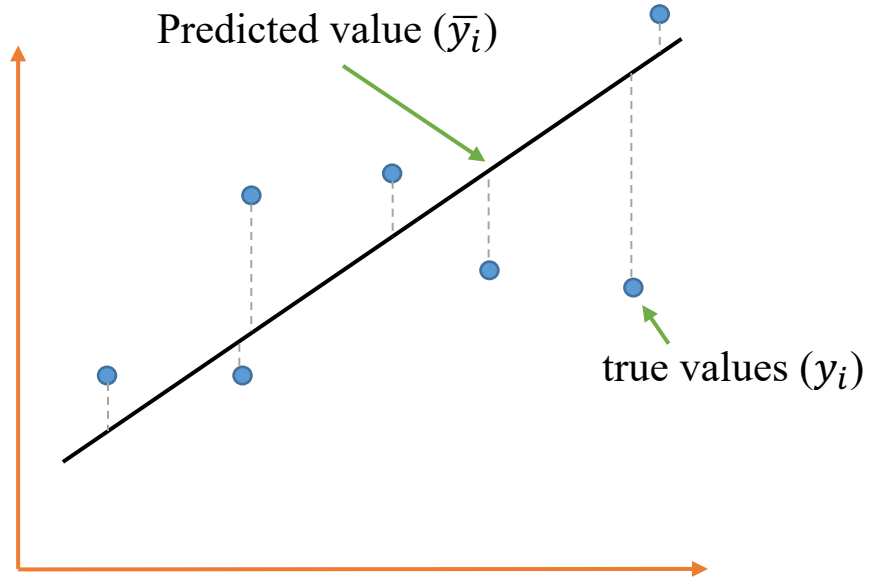
Root-mean-square Error (RMSE)

RMSE is a standard metric to **measure the error** of a model in predicting quantitative data.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\bar{y}_i - y_i)}{n}}$$

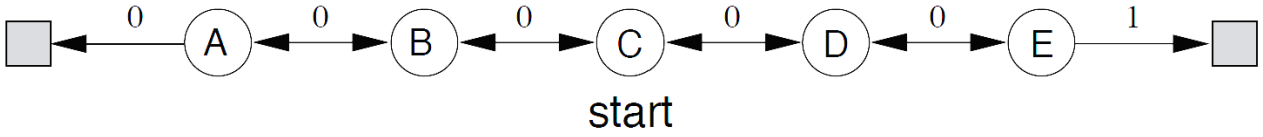
What is the Function Approximation

- ✓ Linear or non-linear
- ✓ Linear regression, ANN, or …



Temporal Differences (TD) algorithm (Prediction)

Temporal Differences TD(0):

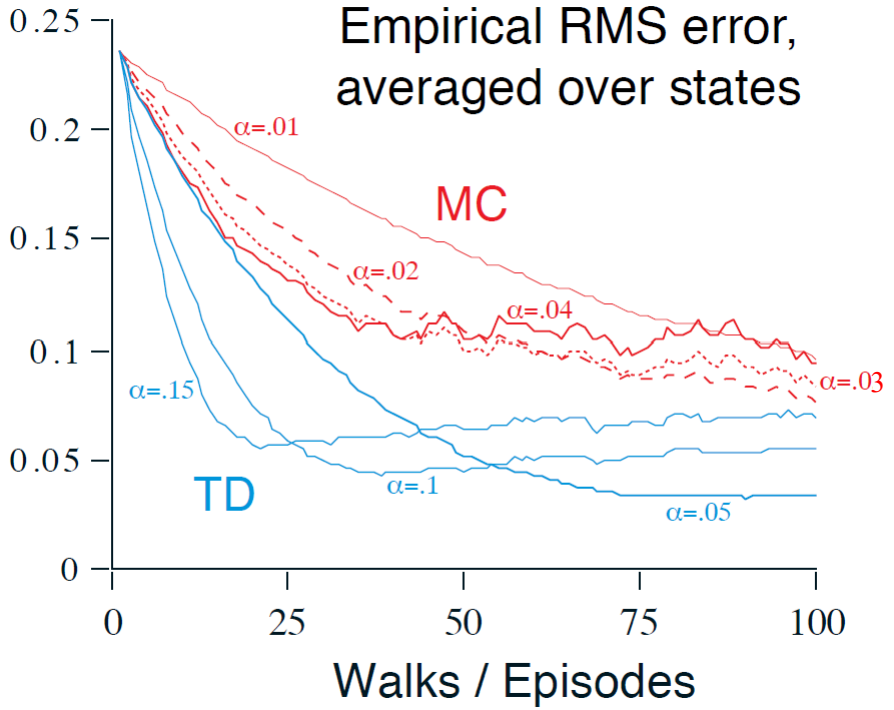


Example

✓ Learning curves for the two methods (TD(0) and MC) for various values of α.

$$V(s) = V(s) + \alpha[r(s') + \gamma V(s') - V(s)]$$

✓ RMSE Here between the learned value function ($\bar{y}_i$) and the true value function (y_i)



Temporal Differences (TD) algorithm (Prediction)

Temporal Differences TD("λ"):

- ✓ If we want to update value before the end of an episode (TD(1)) and use more than a 1 step ahead (TD(0)) for our estimation *(change more states by some affections)*

- ✓ Two types of implementations for TD(λ):
  - ➢ Forward View (is not directly implementable)
  - ➢ **Backward View** (simple conceptually)

Both they have same performance (see in Chapter 12, Richard S. for proof)

Temporal Differences (TD) algorithm (Prediction)

TD("λ") backward view:

- ✓ The backward view of TD(λ) updates values at each step:
  - ➢ After each step in the episode it makes **updates to all prior steps.**

Then how we can weight or assign credit to all prior steps appropriately?

By using Eligibility Traces (ET)

Note: So λ refers to the use of an eligibility trace

Temporal Differences (TD) algorithm (Prediction)

TD("λ") backward view:

Eligibility Traces (ET)

- ✓ Eligibility Traces (ET) increases the frequency of entering a state s at time t denoted as $e_t(s)$
- ✓ An additional memory variable associated with each state (ET)
- ✓ On each step, the eligibility traces for all states decay by lambda (λ) and gamma (γ) is as follows:

$$e_t(s) = \begin{cases} \gamma\lambda e_{t-1}(s) & if \;\; s \neq S_t \\ \gamma\lambda e_{t-1}(s) + 1 & if \;\; s = S_t \end{cases}$$

γ: *discount rate*
λ: *trace-decay parameter*

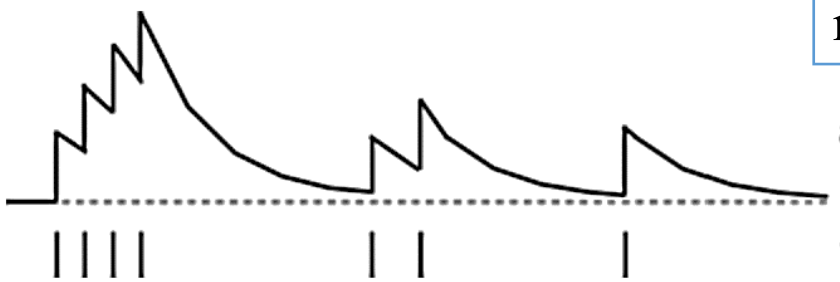
Temporal Differences (TD) algorithm (Prediction)

TD("λ") backward view:

Eligibility Traces (ET)

✓ This kind of ET is called an ***accumulating trace*** because:

- It accumulates each time the state is visited
- Then fades away gradually when the state is not visited



record which states have recently ($\lambda\gamma$) been visited

The traces rate indicates the degree to which each state is *eligible* for undergoing learning changes should a reinforcing event occur.

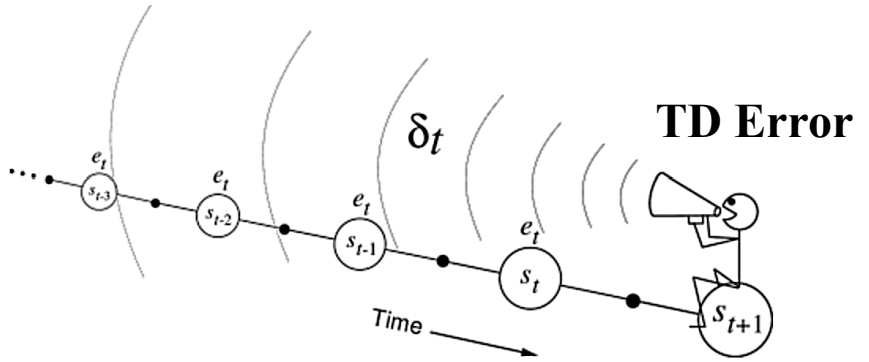
Temporal Differences (TD) algorithm (Prediction)

TD("λ") backward view:

Eligibility Traces (ET)

- ✓ ET assigns credit to states that are both **visited frequently** as well as **visited recently** with respect to the terminal state.

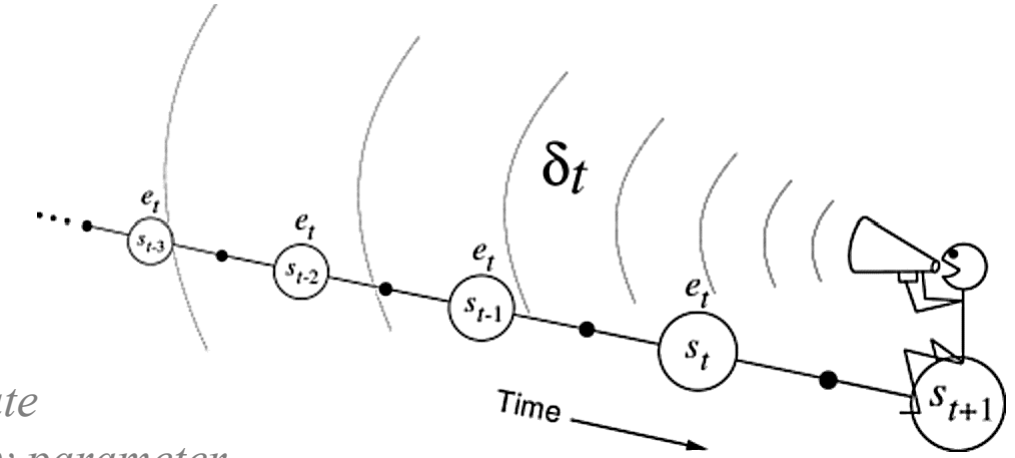
- ✓ At each moment we **look at the current TD error** and assign it backward to each prior state according to the state's eligibility trace at that time.



Temporal Differences (TD) algorithm (Prediction)

TD("λ") backward view:

Eligibility Traces (ET)



Initialize V(s) arbitrarily and e(s_t) = 0, for all $s \in S$
Repeat (for each episode):
  Initialize s_t
  Repeat (for each step of episode):
    a_t = action given by π for s_t
    Take the action a_t, observe reward r, and next state s_{t+1}
    $\delta = r + \gamma V(s_{t+1}) - V(s_t)$
    $e(s_t) = e(s_t) + 1$
    For all states S:
      $V(s_t) = V(s_t) + \alpha\delta e(s_t)$
      $e(s_t) = \gamma\lambda e(s_t)$
    $s_t = s_{t+\_1}$
    until s is terminal

γ: *discount rate*
λ: *trace-decay parameter*
δ: TD error

Propagate current information in past states

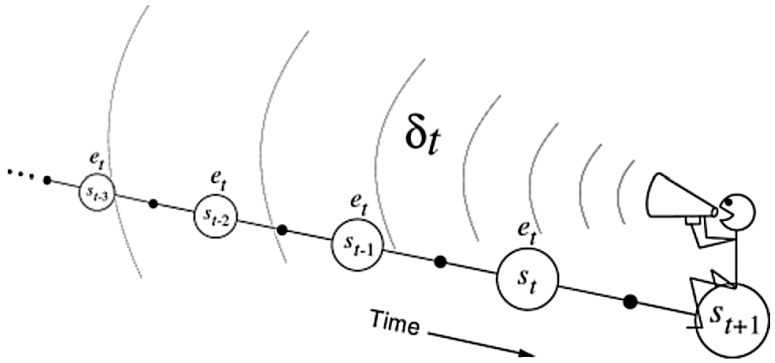
What happens if value of λ is 0?
Then we have TD(0)

What happens if value of λ is 1?
Then we have Monte Carlo similar behavior; TD(1)

Temporal Differences (TD) algorithm (Prediction)

Eligibility Traces (ET)

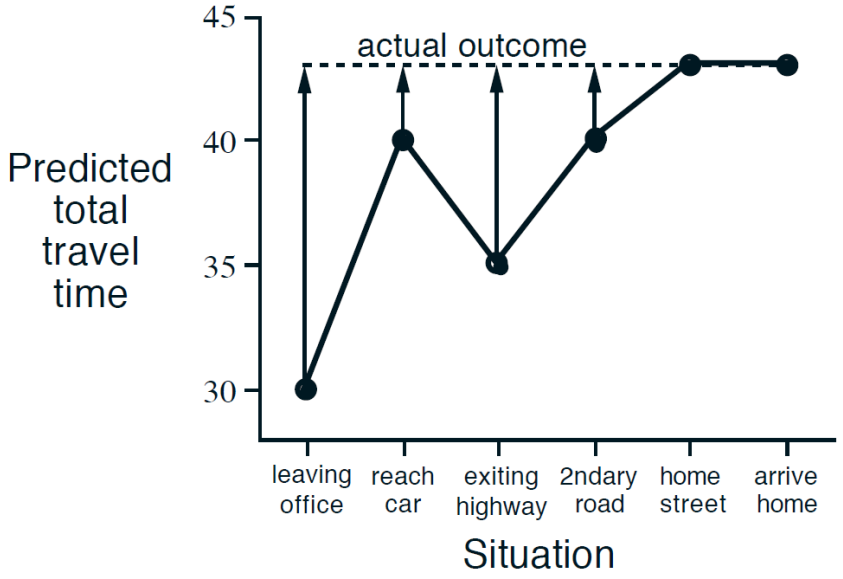
- ✓ Almost any TD method, **can be combined with ET** to obtain a more general method that **may** learn more efficiently..
- ✓ **For example:** Q-learning or SARSA, …



Temporal Differences (TD) learning algorithm

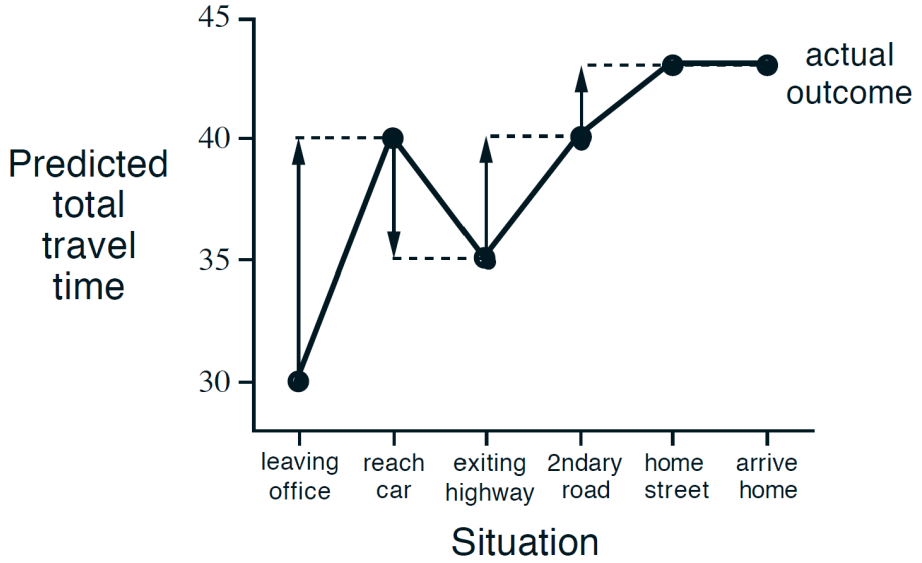
Changes recommended by TD, and MC methods

Monte-Carlo



Changes recommended by Monte Carlo methods

Temporal Differences



Changes recommended by TD methods

Temporal Differences (TD) algorithm (Control)

SARSA as a Temporal Difference (TD) method

- ✓ State–action–reward–state–action (SARSA) combines both Monte Carlo and dynamic programming methods
- ✓ The update equation has the similar form of Monte Carlo's update equation with some differences
- ✓ SARSA replaces the actual return G_t from the data with (**bellman equation**)

$$r_t + \gamma Q(s_{t+1}, a_{t+1})$$

We will discuss SARSA in next chapter in more details

Temporal Differences (TD) learning algorithm

Bootstrapping:

When you estimate value based on another estimated value e.g. Q-value here

$$V(s_t) = V(s_t) + \alpha(R_{t+1} + Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$$

SARSA (next chapter)

Sampling:

When you estimate value does not use any another estimate

$$V(s_t) = V(s_t) + \alpha(G_t - V(s_t))$$

Monte-Carlo

Temporal Differences

Dynamic Programming

Why? $V_\pi(s) = r(s) + \gamma \sum_{s' \in S} p(s', r|s, \pi(s)) V_\pi(s')$

Summery

- ✓ Temporal Differences (TD) learning algorithm
- ✓ Temporal Differences (1) → $V(s_t) = V(s_t) + \alpha(G_t - V(s_t))$
- ✓ Temporal Differences (0) → $V(s) = V(s) + \alpha[r(s') + \gamma V(s') - V(s)]$
- ✓ Temporal Differences (λ) →

$$\delta = r + \gamma V(s_{t+1}) - V(s_t)$$
$$e(s_t) = e(s_t) + 1$$
For all states S:
$$V(s_t) = V(s_t) + \alpha\delta e(s_t)$$
$$e(s_t) = \gamma\lambda e(s_t)$$

- ✓ Bootstrapping and sampling