“For even the very wise cannot see all ends.”

J.R.R. Tolkien, The Fellowship of the Ring

CMPUT 365 Introduction to RL

Marlos C. Machado

Class 28/35

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.

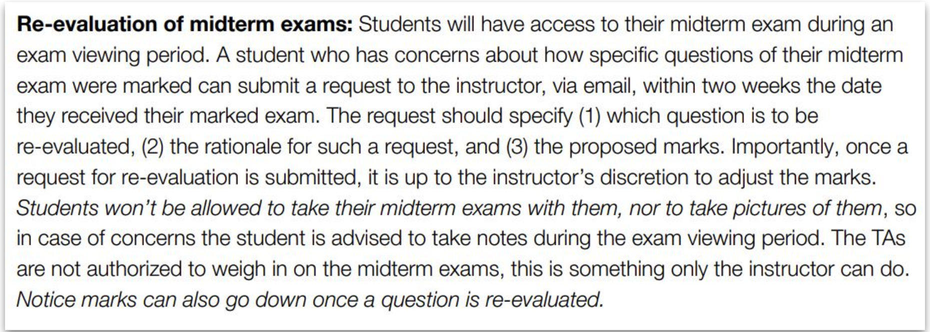
You **need** to **check**, **every time**, if you are in the private session and if you are submitting quizzes and assignments to the private section.

The deadlines in the public session **do not align** with the deadlines in Coursera.

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

Reminder II / Updates

- The programming assignment is due today.
- I need to be away on Friday, something just came up.
  - I'll record a lecture and upload it until Thursday.
- Exam viewing will be on Friday.
  - ATH 3-28 from 1pm to 5pm.
- Rich Sutton will give a guest lecture Dec 1st (next next Friday).



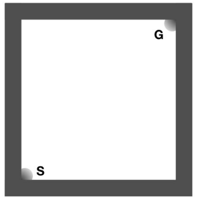
Re-evaluation of midterm exams: Students will have access to their midterm exam during an exam viewing period. A student who has concerns about how specific questions of their midterm exam were marked can submit a request to the instructor, via email, within two weeks the date they received their marked 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.*

Please, interrupt me at any time!

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

Last Class: Function Approximation

- In the tabular case, $\mathbf{v}_\pi \in \mathbb{R}^{|\mathcal{S}|}$.
- Instead, we will approximate $\mathbf{v}_\pi$ using a function parameterized by some weights $\mathbf{w} \in \mathbb{R}^d$ where $d \ll |\mathcal{S}|$. We will write $\hat{v}(s,\mathbf{w}) \approx v_\pi(s)$.
- An example:



$$\mathbf{s} = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} \qquad \mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} \qquad \hat{v}(\mathbf{s}, \mathbf{w}) = \mathbf{s}^\top \mathbf{w}$$

- Extending RL to function approximation also makes it applicable to partially observable problems, in which the full state is not available to the agent.
  I may use **o** to denote the agent's observation (instead of **s**).

Last Class: The Prediction Objective (A Notion of Accuracy)

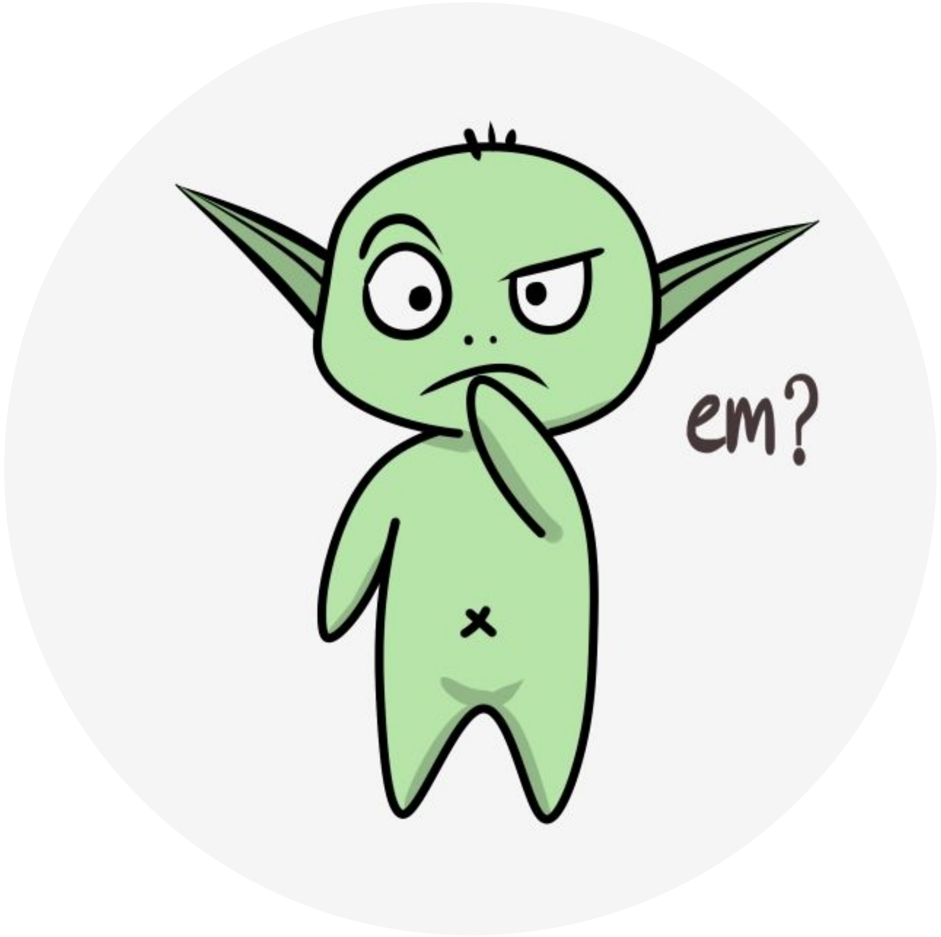
- In the tabular case we can have equality, but with FA, not anymore.
  - Making one state's estimate more accurate invariably means making others' less accurate.
- Mean Squared Error:

How much do we care about the error in each state s.

$$\overline{\text{VE}}(\mathbf{w}) \doteq \sum_{s \in \mathcal{S}} \mu(s) \Big[v_\pi(s) - \hat{v}(s, \mathbf{w}) \Big]^2$$

Usually, the fraction of time spent in s.
On-policy distribution.

- When doing nonlinear function approximation, we lose pretty much every guarantee we had (often, even convergence guarantees).



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

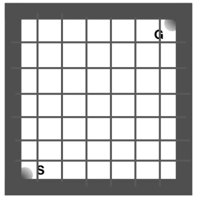
Recipe for Deriving a Concrete Algorithm for SGD

1. Specify a function approximation architecture (parametric form of v_π).
2. Write down your objective function.
3. Take the derivative of the objective function with respect to the weights.
4. Simplify the general gradient expression for your parametric form.
5. Make a weight update rule:

   $w = w - \alpha$ `GRAD`

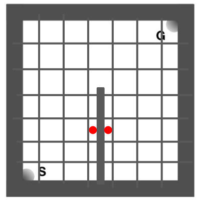
1. Specify a FA architecture (parametric form of v_π)

- We will use *state aggregation with linear function approximation*



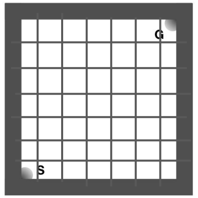
1. Specify a FA architecture (parametric form of v_π)

- We will use *state aggregation* with *linear function approximation*
- State aggregation
  - The features are always binary with only a single active feature that is not zero



State aggregation is far from perfect!

1. Specify a FA architecture (parametric form of v_π)

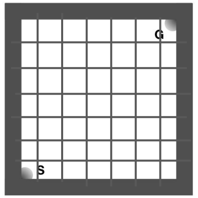


- We will use *state aggregation* with *linear function approximation*
- State aggregation
    - The features are always binary with only a single active feature that is not zero
- Value function
    - Linear function

$$v_\pi(s) \approx \hat{v}(s, \mathbf{w}) \doteq \mathbf{w}^\top \mathbf{x}(s) \doteq \sum_{i=1}^{d} w_i \cdot x_i(s)$$

2. Write down your objective function

- We will use the *value error*



$$\overline{\mathrm{VE}}(\mathbf{w}) \doteq \sum_{s\in\mathcal{S}} \mu(s)\Big[v_\pi(s) - \hat{v}(s,\mathbf{w})\Big]^2$$

$$= \sum_{s\in\mathcal{S}} \mu(s)\Big[v_\pi(s) - \mathbf{w}^\top \mathbf{x}(s)\Big]^2$$

3. Take the derivative of the obj. function w.r.t. the weights

$$
\begin{aligned}
\nabla\overline{\mathrm{VE}}(\mathbf{w}) &= \nabla \sum_{s\in\mathcal{S}} \mu(s)\Big[v_\pi(s) - \mathbf{w}^\top\mathbf{x}(s)\Big]^2 \\
&= \sum_{s\in\mathcal{S}} \mu(s)\nabla\Big[v_\pi(s) - \mathbf{w}^\top\mathbf{x}(s)\Big]^2 \\
&= -\sum_{s\in\mathcal{S}} \mu(s)2\Big[v_\pi(s) - \mathbf{w}^\top\mathbf{x}(s)\Big]\nabla\mathbf{w}^\top\mathbf{x}(s)
\end{aligned}
$$

4. Simplify the general gradient expression

$$\begin{aligned}\nabla \overline{\mathrm{VE}}(\mathbf{w}) &= -\sum_{s\in\mathcal{S}} \mu(s) 2\left[v_\pi(s) - \mathbf{w}^\top \mathbf{x}(s)\right] \boxed{\nabla \mathbf{w}^\top \mathbf{x}(s)} \\ &= -\sum_{s\in\mathcal{S}} \mu(s) 2\left[v_\pi(s) - \mathbf{w}^\top \mathbf{x}(s)\right] \mathbf{x}(s)\end{aligned}$$

$$\nabla \mathbf{w}^\top \mathbf{x}(s) = \mathbf{x}(s)$$

5. Make a weight update rule

$$\nabla\overline{\mathrm{VE}}(\mathbf{w}) = -\sum_{s\in\mathcal{S}}\mu(s)2\Big[v_\pi(s) - \mathbf{w}^\top\mathbf{x}(s)\Big]\mathbf{x}(s)$$

$$\begin{aligned}\mathbf{w}_{t+1} &= \mathbf{w}_t + \alpha 2\Big[v_\pi(s) - \mathbf{w}^\top\mathbf{x}(s)\Big]\mathbf{x}(s)\\ &= \mathbf{w}_t + \alpha\Big[\boxed{v_\pi(s)} - \mathbf{w}^\top\mathbf{x}(s)\Big]\mathbf{x}(s)\end{aligned}$$

Stochastic-gradient and Semi-gradient Methods

- The approximate value function, $\hat{v}(s,\mathbf{w})$, needs to be a differentiable function of **w** for all states.
- For this class, consider that, on each step, we observe a new example $S_t \mapsto v_\pi(S_t)$. Even with the exact target, we need to properly allocate resources.
- *Stochastic gradient-descent (SGD)* is a great strategy:

$$\begin{aligned}\mathbf{w}_{t+1} &\doteq \mathbf{w}_t - \frac{1}{2}\alpha\nabla\Big[v_\pi(S_t) - \hat{v}(S_t,\mathbf{w}_t)\Big]^2 \\ &= \mathbf{w}_t + \alpha\Big[v_\pi(S_t) - \hat{v}(S_t,\mathbf{w}_t)\Big]\nabla\hat{v}(S_t,\mathbf{w}_t)\end{aligned}$$

$$\nabla f(\mathbf{w}) \doteq \left(\frac{\partial f(\mathbf{w})}{\partial w_1}, \frac{\partial f(\mathbf{w})}{\partial w_2}, \ldots, \frac{\partial f(\mathbf{w})}{\partial w_d}\right)^\top$$

Few (one) state at a time

We need to consider the impact of our update. Thus, small updates are often preferred.

A More Realistic Update

- Let U_t denote the *t*-th training example, $S_t \mapsto v_\pi(S_t)$, of some (possibly random), approximation to the true value.

$$\mathbf{w}_{t+1} \doteq \mathbf{w}_t + \alpha\Big[U_t - \hat{v}(S_t,\mathbf{w}_t)\Big]\nabla\hat{v}(S_t,\mathbf{w}_t)$$

Gradient Monte Carlo Algorithm for Estimating $\hat{v} \approx v_\pi$

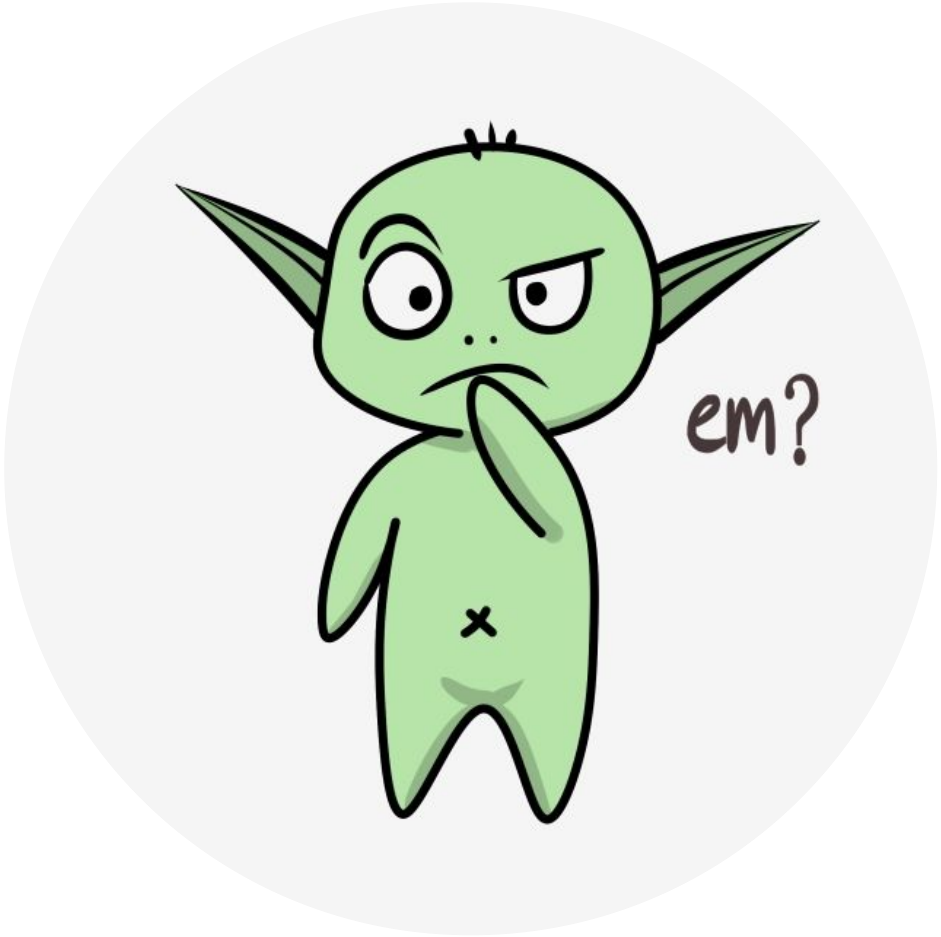
```
Input: the policy $\pi$ to be evaluated
Input: a differentiable function $\hat{v} : \mathcal{S} \times \mathbb{R}^d \to \mathbb{R}$
Algorithm parameter: step size $\alpha > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)

Loop forever (for each episode):
    Generate an episode $S_0, A_0, R_1, S_1, A_1, \ldots, R_T, S_T$ using $\pi$
    Loop for each step of episode, $t = 0, 1, \ldots, T-1$:
        $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[G_t - \hat{v}(S_t,\mathbf{w})\big]\nabla\hat{v}(S_t,\mathbf{w})$
```

A Clearer Instantiation — Linear Function Approximation

- Let $\hat{v}(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\top \mathbf{x}$. We have $\nabla_{\mathbf{w}} \hat{v}(\mathbf{x}, \mathbf{w}) = \mathbf{x}(s)$.
- Thus, $\mathbf{w}_{t+1} \doteq \mathbf{w}_t + \alpha \left[U_t - \hat{v}(\mathbf{x}, \mathbf{w})\right] \nabla_{\mathbf{w}} \hat{v}(\mathbf{x}, \mathbf{w})$ becomes:

  $\mathbf{w}_{t+1} \doteq \mathbf{w}_t + \alpha \left[U_t - \hat{v}(\mathbf{x}, \mathbf{w})\right]\mathbf{x}$.

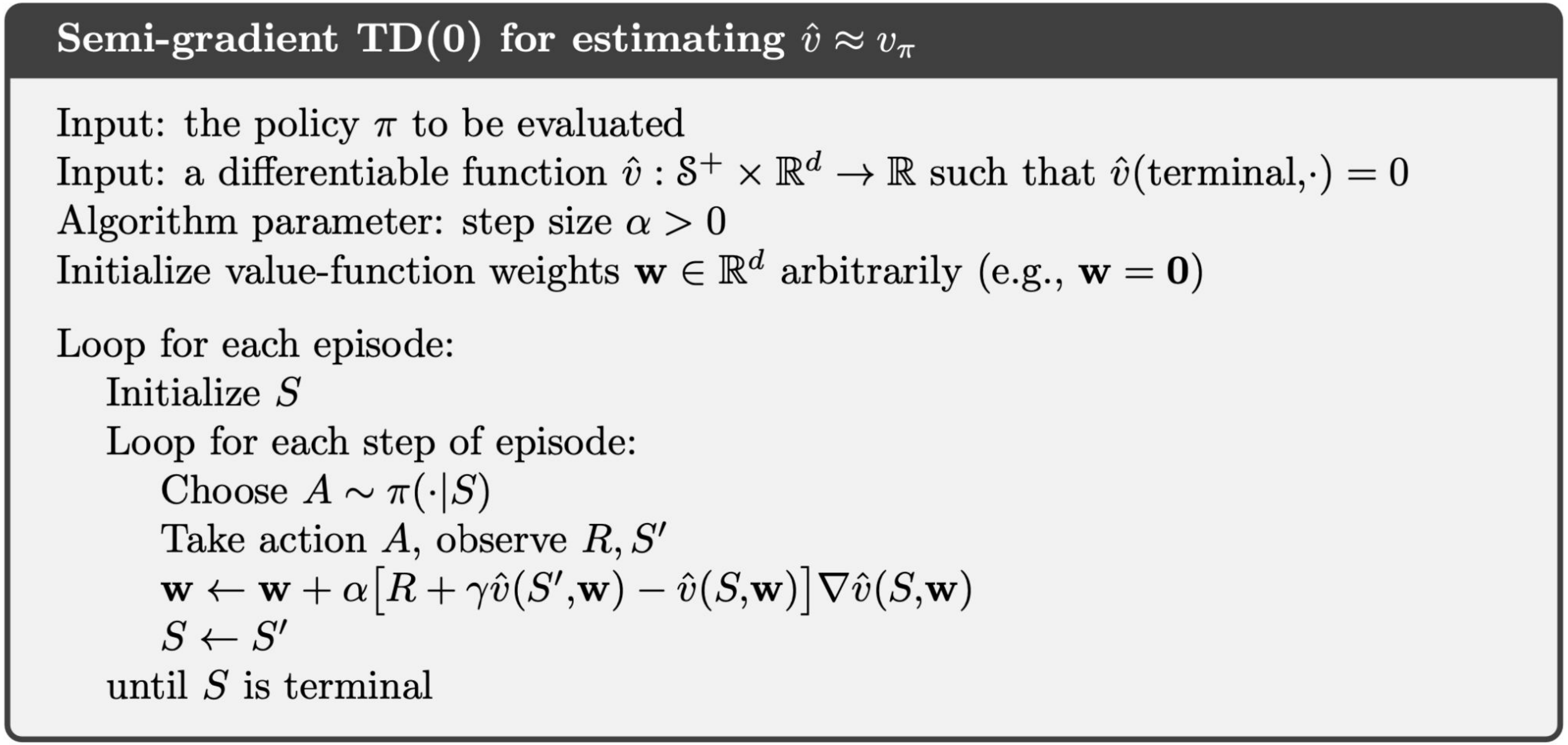


https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Semi-gradient TD

- What if $U_t \doteq R_{t+1} + \gamma\, \hat{v}(S_{t+1},\mathbf{w}_t)$?
- We lose several guarantees when we use a bootstrapping estimate as target.
  - The target now also depends on the value of $\mathbf{w}_t$, so the target is not independent of $\mathbf{w}_t$.
- Bootstrapping are not instances of true gradient descent. They take into account the effect of changing the weight vector $\mathbf{w}_t$ on the estimate, but ignore its effect on the target. Thus, they are a *semi-gradient method*.
- Regardless of the theoretical guarantees, we use them all the time ¯_(ツ)_/¯

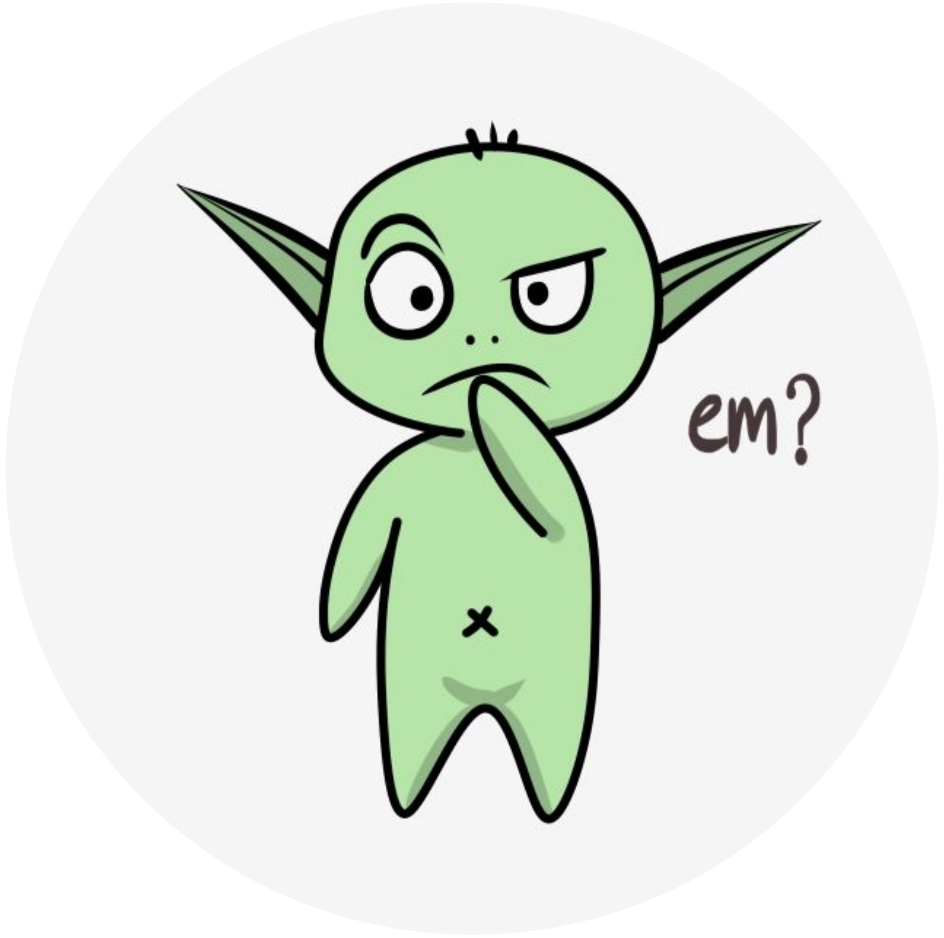
Semi-gradient TD(0)



Semi-gradient TD(0) for estimating $\hat{v} \approx v_\pi$

```
Input: the policy $\pi$ to be evaluated
Input: a differentiable function $\hat{v} : \mathcal{S}^+ \times \mathbb{R}^d \to \mathbb{R}$ such that $\hat{v}(\text{terminal},\cdot) = 0$
Algorithm parameter: step size $\alpha > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)

Loop for each episode:
    Initialize $S$
    Loop for each step of episode:
        Choose $A \sim \pi(\cdot|S)$
        Take action $A$, observe $R, S'$
        $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R + \gamma\hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})\big]\nabla\hat{v}(S,\mathbf{w})$
        $S \leftarrow S'$
    until $S$ is terminal
```



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

TD Fixed Point with Linear Function Approximation

- We do have convergence results for linear function approximation.

$$\begin{aligned}\mathbf{w}_{t+1} &\doteq \mathbf{w}_t + \alpha\Big(R_{t+1} + \gamma \mathbf{w}_t^\top \mathbf{x}_{t+1} - \mathbf{w}_t^\top \mathbf{x}_t\Big)\mathbf{x}_t \\ &= \mathbf{w}_t + \alpha\Big(R_{t+1}\mathbf{x}_t - \mathbf{x}_t\big(\mathbf{x}_t - \gamma \mathbf{x}_{t+1}\big)^\top \mathbf{w}_t\Big)\end{aligned}$$

TD Fixed Point with Linear Function Approximation

- We do have convergence results for linear function approximation.

$$\begin{aligned}\mathbf{w}_{t+1} &\doteq \mathbf{w}_t + \alpha\Big(R_{t+1} + \gamma \mathbf{w}_t^\top \mathbf{x}_{t+1} - \mathbf{w}_t^\top \mathbf{x}_t\Big)\mathbf{x}_t \\ &= \mathbf{w}_t + \alpha\Big(R_{t+1}\mathbf{x}_t - \mathbf{x}_t\big(\mathbf{x}_t - \gamma \mathbf{x}_{t+1}\big)^\top \mathbf{w}_t\Big)\end{aligned}$$

In a steady state, for any given $\mathbf{w}_t$, the expected next weight vector can be written

$$\mathbb{E}[\mathbf{w}_{t+1}|\mathbf{w}_t] = \mathbf{w}_t + \alpha(\mathbf{b} - \mathbf{A}\mathbf{w}_t)$$

where $\mathbf{b} \doteq \mathbb{E}[R_{t+1}\mathbf{x}_t] \in \mathbb{R}^d$ and $\mathbf{A} \doteq \mathbb{E}\Big[\mathbf{x}_t\big(\mathbf{x}_t - \gamma\mathbf{x}_{t+1}\big)^\top\Big] \in \mathbb{R}^{d\times d}$

TD Fixed Point with Linear Function Approximation

- We do have convergence results for linear function approximation.

$$\begin{aligned}\mathbf{w}_{t+1} &\doteq \mathbf{w}_t + \alpha\Big(R_{t+1} + \gamma \mathbf{w}_t^\top \mathbf{x}_{t+1} - \mathbf{w}_t^\top \mathbf{x}_t\Big)\mathbf{x}_t \\ &= \mathbf{w}_t + \alpha\Big(R_{t+1}\mathbf{x}_t - \mathbf{x}_t\big(\mathbf{x}_t - \gamma\mathbf{x}_{t+1}\big)^\top \mathbf{w}_t\Big)\end{aligned}$$

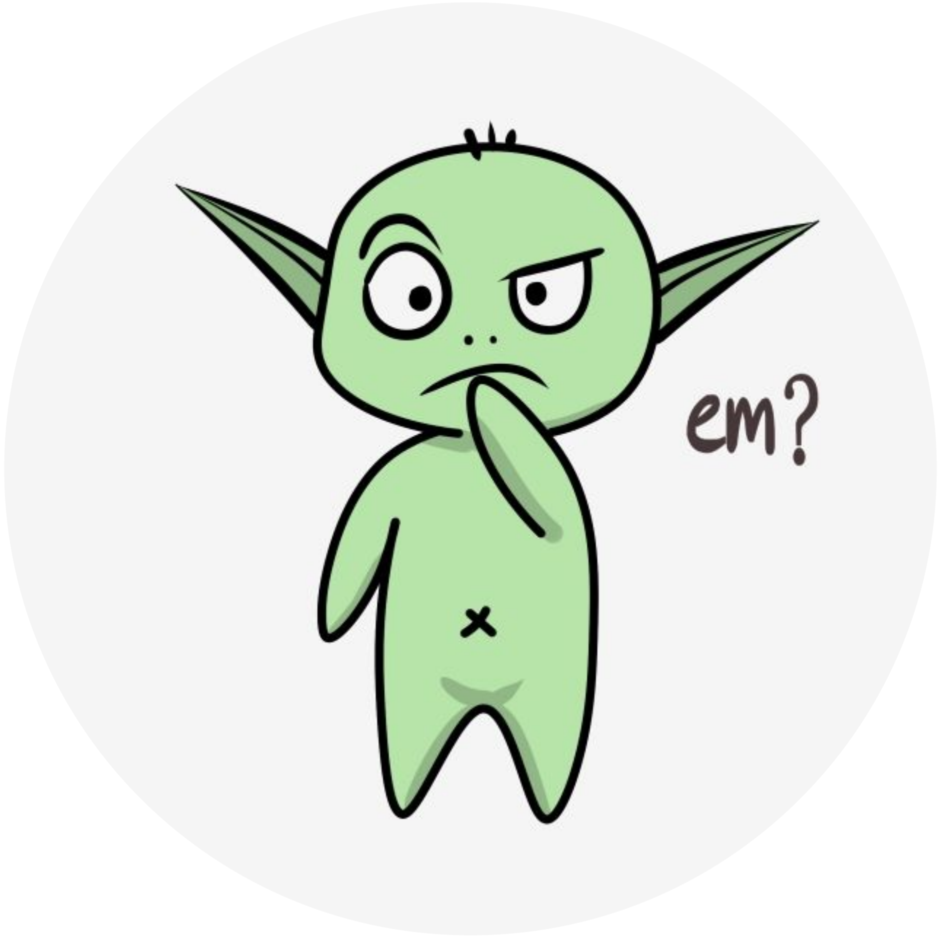
In a steady state, for any given **w**$_t$, the expected next weight vector can be written

$$\mathbb{E}[\mathbf{w}_{t+1}|\mathbf{w}_t] = \mathbf{w}_t + \alpha(\mathbf{b} - \mathbf{A}\mathbf{w}_t)$$

where $\mathbf{b} \doteq \mathbb{E}[R_{t+1}\mathbf{x}_t] \in \mathbb{R}^d \quad \text{and} \quad \mathbf{A} \doteq \mathbb{E}\Big[\mathbf{x}_t\big(\mathbf{x}_t - \gamma\mathbf{x}_{t+1}\big)^\top\Big] \in \mathbb{R}^{d\times d}$

It converges to:

$$\begin{aligned} & \mathbf{b} - \mathbf{A}\mathbf{w}_{\text{TD}} = \mathbf{0} \\ \Rightarrow \quad & \mathbf{b} = \mathbf{A}\mathbf{w}_{\text{TD}} \\ \Rightarrow \quad & \mathbf{w}_{\text{TD}} \doteq \mathbf{A}^{-1}\mathbf{b}.\end{aligned}$$



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

Question

How are tabular methods related to linear function approximation?