

Reinforcement Learning: Tutorial 7

On-policy TD learning with approximation

Week 4
University of Amsterdam

Alejandro Garcia
February 2024

Outline

- 1 On-policy TD learning with approximation exercises
- 2 Ask anything about HW3



Tutorial 8 Overview

- 1 On-policy TD learning with approximation exercises
- 2 Ask anything about HW3

Tutorial 7 Overview

- 1 On-policy TD learning with approximation exercises
 - Questions 6.1-6.4
- 2 Ask anything about HW3



Theory Intermezzo: Semi-gradient TD(0), LSTD

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

Input: the policy π to be evaluated

Input: a differentiable function $\hat{v} : \mathcal{S}^+ \times \mathbb{R}^d \rightarrow \mathbb{R}$ 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'

$\mathbf{w} \leftarrow \mathbf{w} + \alpha [R + \gamma \hat{v}(S', \mathbf{w}) - \hat{v}(S, \mathbf{w})] \nabla \hat{v}(S, \mathbf{w})$

$S \leftarrow S'$

until S is terminal

LSTD for estimating $\hat{v} = \mathbf{w}^\top \mathbf{x}(\cdot) \approx v_\pi$ ($O(d^2)$ version)

Input: feature representation $\mathbf{x} : \mathcal{S}^+ \rightarrow \mathbb{R}^d$ such that $\mathbf{x}(\text{terminal}) = \mathbf{0}$

Algorithm parameter: small $\varepsilon > 0$

$\widehat{\mathbf{A}}^{-1} \leftarrow \varepsilon^{-1} \mathbf{I}$

A $d \times d$ matrix

$\widehat{\mathbf{b}} \leftarrow \mathbf{0}$

A d -dimensional vector

Loop for each episode:

Initialize S ; $\mathbf{x} \leftarrow \mathbf{x}(S)$

Loop for each step of episode:

Choose and take action $A \sim \pi(\cdot|S)$, observe R, S' ; $\mathbf{x}' \leftarrow \mathbf{x}(S')$

$\mathbf{v} \leftarrow \widehat{\mathbf{A}}^{-1} (\mathbf{x} - \gamma \mathbf{x}')$

$\widehat{\mathbf{A}}^{-1} \leftarrow \widehat{\mathbf{A}}^{-1} - (\widehat{\mathbf{A}}^{-1} \mathbf{x}) \mathbf{v}^\top / (1 + \mathbf{v}^\top \mathbf{x})$

$\widehat{\mathbf{b}} \leftarrow \widehat{\mathbf{b}} + R \mathbf{x}$

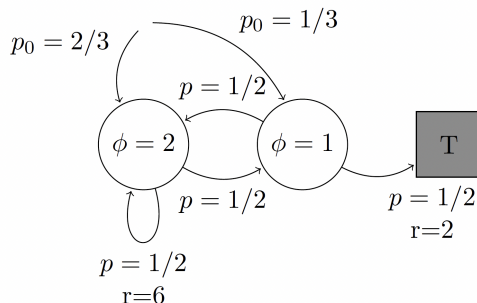
$\mathbf{w} \leftarrow \widehat{\mathbf{A}}^{-1} \widehat{\mathbf{b}}$

$S \leftarrow S'$; $\mathbf{x} \leftarrow \mathbf{x}'$

until S' is terminal

Q 6.1 On-policy distributions and LSTD

Observe the example MDP in the Figure below. This MDP has a discount factor $\gamma = 2/3$, an initial distribution p_0 as illustrated, and transition dynamics p under the current policy and rewards r as illustrated (transitions where no reward is mentioned have reward 0). Answer the following questions:



Q 6.1 On-policy distributions and LSTD

- 1 What is the on-policy distribution μ for this example? Do not forget to use the discount factor γ in your calculation!

Q 6.1 On-policy distributions and LSTD

- ① What is the on-policy distribution μ for this example? Do not forget to use the discount factor γ in your calculation!

First calculate h by solving:

$$h = p_0 + \gamma \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 0 \end{bmatrix} h \quad (1)$$

$$\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} 2/3 \\ 1/3 \end{bmatrix} + \gamma \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} 1 - \gamma/2 & -\gamma/2 \\ -\gamma/2 & 1 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} 2/3 \\ 1/3 \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} 1 - 1/3 & -1/3 \\ -1/3 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2/3 \\ 1/3 \end{bmatrix} = \begin{bmatrix} 7/5 \\ 4/5 \end{bmatrix} \quad (4)$$

To find μ we normalize the answer and find:

$$\mu = \begin{bmatrix} 7/11 \\ 4/11 \end{bmatrix}.$$

Q 6.1 On-policy distributions and LSTD

- 2 How frequently would each of the transitions occur?

Q 6.1 On-policy distributions and LSTD

- 2 How frequently would each of the transitions occur?

The transitions out of state 1 each happen with frequency $7/11 * 1/2 = 7/22$. The transitions out of state 2 each with frequency $4/11 * 1/2 = 4/22$.

Q 6.1 On-policy distributions and LSTD

- 3 Taking the previous question into account, think of a way to use the LSTD equations to find the TD fixpoint for this example.

Q 6.1 On-policy distributions and LSTD

- 3 Taking the previous question into account, think of a way to use the LSTD equations to find the TD fixpoint for this example.

LSTD is normally based on samples. To get the real minimizer based on the dynamics, we have to add each transition with the frequency that occurs (equivalently, multiply their contribution in the sum with their frequency). We should also not use regularization ($\epsilon = 0$). We thus calculate:

$$A = 7 \cdot x_1(x_1 - \gamma x_1) + 7 \cdot x_1(x_1 - \gamma x_2) + 4 \cdot x_2(x_2 - \gamma x_1) + 4 \cdot x_2(x_2 - 0) \quad (1)$$

$$= 7 \cdot 2(2 - 2/3 \cdot 2) + 7 \cdot 2(2 - 2/3) + 4 \cdot 1(1 - 2/3 \cdot 2) + 4 \cdot (1 - 0) \quad (2)$$

$$= 92/3 \quad (3)$$

Q 6.1 On-policy distributions and LSTD

- ③ Taking the previous question into account, think of a way to use the LSTD equations to find the TD fixpoint for this example.

Hence, $A = 92/3$, and

$$b = 7 \cdot x_1 R_{11} + 7 \cdot x_1 R_{12} + 4 \cdot x_2 R_{21} + 4 \cdot x_2 R_{2T} \quad (1)$$

$$= 7 \cdot 2 \cdot 6 + 0 + 0 + 4 \cdot 2 \quad (2)$$

$$= 92. \quad (3)$$

Now,

$$A^{-1}b = 3/92 * 92 = 3$$

Q 6.2 Basis functions

- 1 Tabular methods can be seen as a special case of linear function approximation. Show that this is the case and give the corresponding feature vectors.

Q 6.2 Basis functions

- 1 Tabular methods can be seen as a special case of linear function approximation. Show that this is the case and give the corresponding feature vectors.

Let s be a state index, \vec{s} its feature vector and \vec{w} a weight vector. Then for linear function approximation, $v(s; \vec{w}) = \vec{s} \cdot \vec{w}$. If we let the feature vector \vec{s} be a vector that is zero everywhere, except at the index corresponding to the state's tabular index, calling $v(s; \vec{w})$ for state i will simply return the i 'th weight, which will correspond to that state's value.

Q 6.2 Basis functions

- 2 What are the advantages of linear function approximation and what are the advantages of non-linear function approximation?

Q 6.2 Basis functions

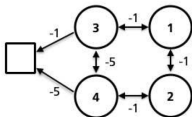
- 2 What are the advantages of linear function approximation and what are the advantages of non-linear function approximation?

There are several advantages for either that can be mentioned: linear function approximation leads to an easy form of the derivative of the Q- or V- value (easier implementation, computationally faster); allows calculating the (approximate) TD-fixpoint in closed form using LSTD, and has robust convergence guarantees to the global maximum. Non-linear function approximation are more expressive (generally better performance if enough data is available), require less hand design, offer flexibility to choose many different architectures.

Q 6.3 Semi-gradient TD and the TD fixed point

We are considering how to travel to a goal location from various locations labeled 1, 2, 3, and 4. There are different travel costs between these locations. A "map" for this problem (showing the possible actions per state) and the associated costs are summarized in the Figure below. We model the problem as an MDP (Figure below), with discount factor $\gamma = 1$. To use only 2 parameters to represent the value function, approximation can be used. We use a linear approximation $\hat{v}(s, \mathbf{w}) = \mathbf{w}^T \phi(s)$. For the four states and the terminal (goal) state, we use the following features respectively:

$$\phi(s1) = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \phi(s2) = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \phi(s3) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \phi(s4) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \phi(T) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$



Q 6.3 Semi-gradient TD and the TD fixed point

- 1 Assume the parameter for the value function approximator in the last time step is $w_t = [0.5, 0.5]^T$, the agent take a new step and receive a transition sample $(s_2, -1, s_4)$, please compute the one step updated value of parameters w_{t+1} with a learning rate α .

Q 6.3 Semi-gradient TD and the TD fixed point

- ① Assume the parameter for the value function approximator in the last time step is $w_t = [0.5, 0.5]^T$, the agent take a new step and receive a transition sample $(s_2, -1, s_4)$, please compute the one step updated value of parameters w_{t+1} with a learning rate α .

Following the semi-gradient equation, we need to evaluate

$$w_{t+1} \leftarrow w_t + \alpha[R + \gamma \hat{v}(s', w_t) - \hat{v}(s, w_t)] \nabla \hat{v}(s, w_t).$$

With a single sample $(s_2, -1, s_4)$ at hand, we can have

$$w_{t+1} = [0.5, 0.5]^T + \alpha \cdot (-1 + \gamma * (0.5) - 1.0) \cdot [0, 2]^T = [0.5, 0.5 - 3\alpha]^T.$$

Q 6.3 Semi-gradient TD and the TD fixed point

- 2 What is the relation between the solution found by LSTD and the Semi-gradient TD method?

Q 6.3 Semi-gradient TD and the TD fixed point

- ② What is the relation between the solution found by LSTD and the Semi-gradient TD method?

LSTD finds the TD fixpoint. Semi-gradient TD, if it converges, converges to this same TD fix point.

Q 6.3 Semi-gradient TD and the TD fixed point

- 3 For the MDP, you have access to the following set of trajectories (with actions not shown):

$$\{(s_1, -1, s_3, -1, T), (s_2, -1, s_4, -5, T)\}$$

where the end of an episode means you reached a terminal state. What solution do TD algorithms converge to when repeatedly trained on this dataset with the given feature function ? Hint: consider the previous sub-question.

Q 6.3 Semi-gradient TD and the TD fixed point

Use LSTD on the data. Since there is only a single solution (the 'A' matrix is well conditioned), there is no need to use ϵ . (Also, the TD algorithms wouldn't use such regularization). Compute the sample estimates by considering all steps for each of the two trajectories:

$$\begin{aligned}\hat{A}_t &= \phi(s_1^{(1)}) \left(\phi(s_1^{(1)}) - \phi(s_3^{(1)}) \right)^T + \phi(s_3^{(1)}) \left(\phi(s_3^{(1)}) - \phi(T^{(1)}) \right)^T \\ &\quad + \phi(s_2^{(2)}) \left(\phi(s_2^{(2)}) - \phi(s_4^{(2)}) \right)^T + \phi(s_4^{(2)}) \left(\phi(s_4^{(2)}) - \phi(T^{(2)}) \right)^T \\ &= \begin{bmatrix} 2 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 2 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right)^T + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right)^T \\ &\quad + \begin{bmatrix} 0 \\ 2 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 2 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)^T + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right)^T \\ &= \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}\end{aligned}$$

Q 6.3 Semi-gradient TD and the TD fixed point

$$\hat{b}_t = -\phi(s_1^{(1)}) - \phi(s_3^{(1)}) + 0 - \phi(s_2^{(2)}) - 5\phi(s_4^{(2)}) + 0 = \begin{bmatrix} -3 \\ -7 \end{bmatrix}$$

Then the solution is,

$$w_t \doteq \hat{A}_t^{-1} \hat{b}_t = \begin{bmatrix} -1 \\ -\frac{7}{3} \end{bmatrix}$$

The value functions can then be computed as:

$$\hat{v}(s_1, w_t) = \begin{bmatrix} -1, -\frac{7}{3} \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = -2$$

$$\hat{v}(s_2, w_t) = \begin{bmatrix} -1, -\frac{7}{3} \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = -\frac{14}{3} \approx -4.67$$

$$\hat{v}(s_3, w_t) = \begin{bmatrix} -1, -\frac{7}{3} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = -1$$

$$\hat{v}(s_4, w_t) = \begin{bmatrix} -1, -\frac{7}{3} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = -\frac{7}{3} \approx -2.33$$

Q 6.3 Semi-gradient TD and the TD fixed point

- 4 Comment on the solution you found under 2. Where is the solution good or bad? Where the solution seems to be bad, can you understand why that is the case?

Q 6.3 Semi-gradient TD and the TD fixed point

- 4 Comment on the solution you found under 2. Where is the solution good or bad? Where the solution seems to be bad, can you understand why that is the case?

In the 'top route' of the MDP the features are able to capture the value function perfectly. In the 'bottom route' of the MDP, the features capture the value function badly. The value at s_2 should be -3 . The only way to do that with the given features is to set w_2 to $-3/2$. However, the value at s_4 clearly needs to be -4 . The only way to do is to set w_2 to -4 . The solution given by the algorithm makes a trade-off of taking into account the on-policy distribution μ , ending up somewhere in the middle.

Q 6.3 Semi-gradient TD and the TD fixed point

- 5 The TD fixed point is independent of the learning rate and certain algorithms based on finding it are said to "never forget". Elaborate what is meant by this and provide one advantage and one disadvantage of "never forgetting".

Q 6.3 Semi-gradient TD and the TD fixed point

- 5 The TD fixed point is independent of the learning rate and certain algorithms based on finding it are said to "never forget". Elaborate what is meant by this and provide one advantage and one disadvantage of "never forgetting".

Advantage: If you never forget a datapoint your method is more sample efficient as you do not throw away any data. Hence you need less data overall.

Disadvantage: If the MDP or the policy changes, never forgetting is a disadvantage since in this case we might want to overwrite older experience with newer experience to correct for this and 'forget' old datapoints to a certain extent. This is talked about in the recording of Lecture 5.

Q 6.3 Semi-gradient TD and the TD fixed point

- 6 (Deep) neural networks are popularly used as function approximators (instead of linear function approximators). In this case $\hat{v}(s, \mathbf{w}) = \text{NN}_{\mathbf{w}}(s)$, where $\text{NN}_{\mathbf{w}}$ is a neural network with parameters (weights and biases) \mathbf{w} . Assuming you have access to a 'autograd()' function that stores $\partial \text{NN}_{\mathbf{w}}(s) / \partial \mathbf{w}$ to $\mathbf{w}.\text{grad}$, how would you implement an update of the v-function for a transition (s, a, r, s', a') ?

Q 6.3 Semi-gradient TD and the TD fixed point

- 6 (Deep) neural networks are popularly used as function approximators (instead of linear function approximators). In this case $\hat{v}(s, \mathbf{w}) = \text{NN}_{\mathbf{w}}(s)$, where $\text{NN}_{\mathbf{w}}$ is a neural network with parameters (weights and biases) \mathbf{w} . Assuming you have access to a 'autograd()' function that stores $\partial \text{NN}_{\mathbf{w}}(s)/\partial \mathbf{w}$ to $\mathbf{w}.\text{grad}$, how would you implement an update of the v-function for a transition (s, a, r, s', a') ?

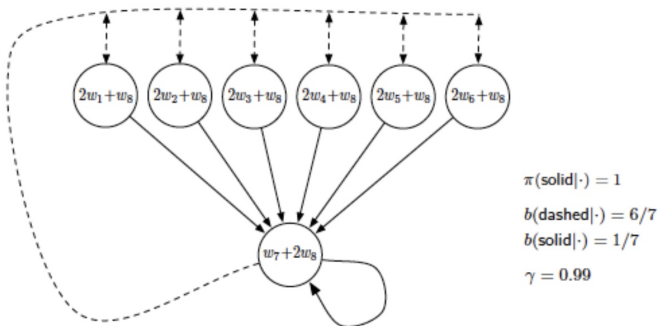
- $\text{val} \leftarrow \text{NN}_{\mathbf{w}}(s)$ (forward pass)
- $\text{valprime} \leftarrow \text{NN}_{\mathbf{w}}(s')$ (forward pass)
- $\text{val.backward}()$ (backward pass)

Then evaluate:

- $\mathbf{w} \leftarrow \mathbf{w} + \alpha[R + \gamma \text{valprime} - \text{val}]\mathbf{w}.\text{grad}.$

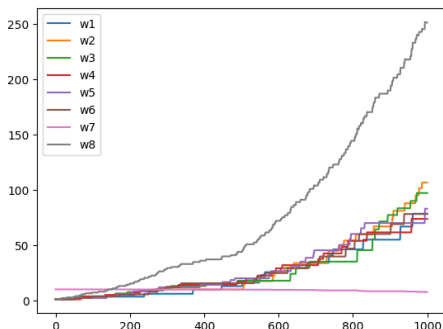
Q 6.4 Preparatory question: Off-policy approximation

Off-policy learning with approximation is a tricky topic. Before we'll dive into it in the next chapter, we'll investigate what happens if we apply the methods you know so far in this setting. On Canvas, you'll find a notebook prepared with an exercise on a problem called 'Baird's Counterexample'.



Q 6.4 Preparatory question: Off-policy approximation

Off-policy learning with approximation is a tricky topic. Before we'll dive into it in the next chapter, we'll investigate what happens if we apply the methods you know so far in this setting. On Canvas, you'll find a notebook prepared with an exercise on a problem called 'Baird's Counterexample'.



'Deadly triad':

- Function approximation
- Semi-gradient bootstrapping
- Off-policy training



Tutorial 7 Overview

- 1 On-policy TD learning with approximation exercises
- 2 Ask anything about HW3
 - Questions 5.3-5.4, 6.5



That's it!



Good luck with the HW and see you the next day!