Reinforcement Learning: Tutorial 9

Off-policy RL with approximation

Week 5
University of Amsterdam

Milena Kapralova
September 2024

Check-in

- How is it going?
- How is HW4?
- If you have any feedback so far, please mail me at *m.kapralova@uva.nl*

Outline

1. Admin
2. Off-policy RL with approximation exercises
3. Ask anything about HW4

Admin

- Please direct any questions about grading to Pieter Pierrot
- Any questions?

Tutorial 9 Overview

1. Off-policy RL with approximation exercises
2. Ask anything about HW4

Tutorial 7 Overview

1. Off-policy RL with approximation exercises
   - Questions 8.1-8.2
2. Ask anything about HW4

Theory Intermezzo: Everything is called Bellman

1. Bellman equation for the value function v_π
$v_\pi(s) = \Sigma_a \pi(a|s) \Sigma_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')]$

2. Bellman operator: plug in $v_\mathbf{w}$ instead of v_π
$B_\pi v_\mathbf{w}(s) \doteq \Sigma_a \pi(a|s) \Sigma_{s',r} p(s', r|s, a)[r + \gamma v_\mathbf{w}(s')]$

3. Bellman error at state s (expectation of the TD error)
$\overline{\delta}_\mathbf{w}(s) \doteq B_\pi v_\mathbf{w}(s) - v_\mathbf{w}(s)$
$\rightarrow \quad = \mathbb{E}_\pi[R_{t+1} + \gamma v_\mathbf{w}(S_{t+1}) - v_\mathbf{w}(S_t)|S_t = s, A_t \sim \pi]$

4. Projected Bellman error at state s: $PBE_\mathbf{w}(s) = \Pi \overline{\delta}_\mathbf{w}(s)$

5. Bellman error vector: Bellman errors for all states in a vector: $\overline{\delta}_\mathbf{w}$

6. Mean squared Bellman error: weigh the norm of the vector by μ
$\overline{BE}(\mathbf{w}) = ||\overline{\delta}_\mathbf{w}||^2_\mu$

Theory Intermezzo: All types of error

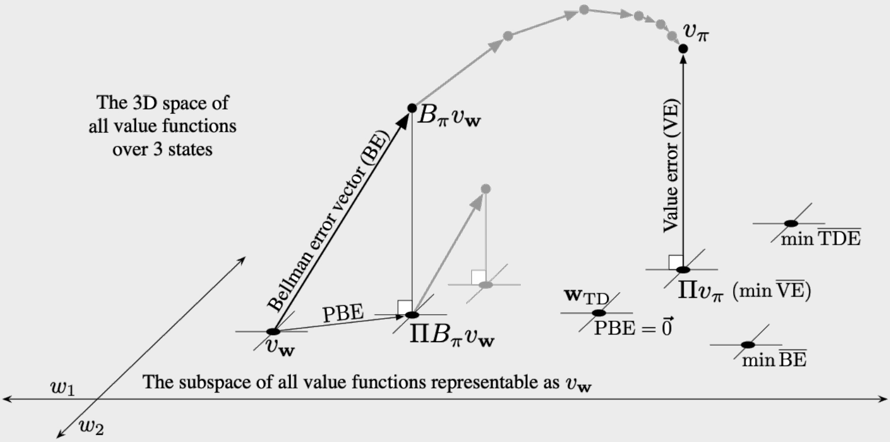
a $\overline{VE}(\mathbf{w}) \doteq ||VE||^2_\mu$; $VE = v_\pi(s) - \hat{v}(s, \mathbf{w})$

b $\overline{TDE}(\mathbf{w}) \doteq \mathbb{E}_b[\rho_t \delta_t^2]$

c $\overline{BE}(\mathbf{w}) \doteq ||\bar{\delta}_\mathbf{w}||^2_\mu$

d $\overline{PBE}(\mathbf{w}) \doteq ||\Pi\bar{\delta}_\mathbf{w}||^2_\mu$

Theory Intermezzo: Geometry of value functions



Q 8.1 Geometry of linear value-function approximation

1. Which error function is minimized by gradient Monte Carlo?

Q 8.1 Geometry of linear value-function approximation

1. Which error function is minimized by gradient Monte Carlo?

   Value error.

Q 8.1 Geometry of linear value-function approximation

2. The Bellman error is zero only when the value error is zero (recall the Bellman equations). Why then does minimizing a TD objective (such as the mean squared (projected) Bellman error) not in general result in minimal mean squared value error ($\overline{VE}$) in the function approximation setting?

Q 8.1 Geometry of linear value-function approximation

2. The Bellman error is zero only when the value error is zero (recall the Bellman equations). Why then does minimizing a TD objective (such as the mean squared (projected) Bellman error) not in general result in minimal mean squared value error ($\overline{VE}$) in the function approximation setting?

If we could update the value function according to the Bellman error vector, we would indeed find the point where BE=VE=0. However, the Bellman error vector takes us out of the representable subspace (see fig 11.3 in RL:AI), so there is no way to update the weights accordingly. Instead, we can try to take gradients of the mean squared temporal difference error or the mean squared (projected) Bellman error, but these objectives have different solutions in general.

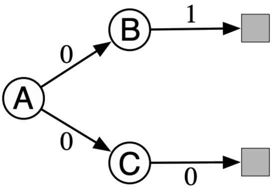
Q 8.1 Geometry of linear value-function approximation

3. Is applying (full) gradient descent on the TD error a good approach to approximate the value function? Motivate your answer.

Q 8.1 Geometry of linear value-function approximation

3. Is applying (full) gradient descent on the TD error a good approach to approximate the value function? Motivate your answer.

If we do full gradient descent on the TD error, we update the weights considering the target value function. This means we also backpropagate the error of the target value to the weights. This can lead to strange situations, where the estimated value of a state depends on how you got there, rather than possible future trajectories from that state. (see example 11.2, p.271 RL:AI).



Q 8.2 *Exam Question: Function approximation

1. Consider two types of function approximation for scalar s:
   a) Using "Gaussian" radial basis features $\left(\phi_j(s) = \exp(-\frac{(s-\mu_j)^2}{2\lambda^2})\right)$ with λ the width of the kernel.
   b) Using polynomial features $\left(\phi_j(s) = s^j\right)$.

   Name one advantage of a) compared to b), and one advantage of b) compared to a). Assume the same number of features is used in both cases.

Q 8.2 *Exam Question: Function approximation

1. Consider two types of function approximation for scalar s:
   a) Using "Gaussian" radial basis features $\left(\phi_j(s) = \exp(-\frac{(s-\mu_j)^2}{2\lambda^2})\right)$ with λ the width of the kernel.
   b) Using polynomial features $\left(\phi_j(s) = s^j\right)$.

   Name one advantage of a) compared to b), and one advantage of b) compared to a). Assume the same number of features is used in both cases.

   Valid advantage RBF: it is local, no extrapolation.
   Valid advantage polynomial: it is global, so extrapolates more, does not require specification of RBF means or knowing the range of inputs.

Q 8.2 *Exam Question: Function approximation

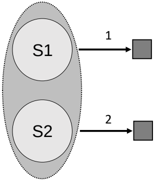
2. With linear function approximation, does gradient Monte Carlo (gradient MC) **always, sometimes, or never** converge to the same solution as semi-gradient TD(0)? Explain your answer.

Q 8.2 *Exam Question: Function approximation

2. With linear function approximation, does gradient Monte Carlo (gradient MC) **always, sometimes, or never** converge to the same solution as semi-gradient TD(0)? Explain your answer.

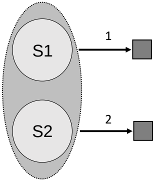
   Sometimes it converges to the same solution as semi-gradient TD(0). In tabular setting, they will find the same solution. But with function approximation, a different objective is optimized that generally has a different optimum.

Q 8.2 *Exam Question: Function approximation



3. Consider the simple MDP shown in Figure above. States S1 and S2 are indistinguishable (have the same features). Only a single action can be applied, that always ends the episode. The reward obtained is 1 or 2, respectively. Episodes start in S1 or S2 with equal probability.
   a) For the shown MDP, what is the minimal mean squared Bellman error? Why?

Q 8.2 *Exam Question: Function approximation



Both states need to be assigned the same value. The best compromise is average: V(S1) = V(S2) = 1.5.Using linear value function approximation with a constant feature c for both states:

$$v_w = \mathbf{w}\ c$$

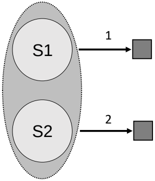
OLS solution $\hat{\mathbf{w}}$:

$$\hat{\mathbf{w}} = \frac{1}{c}\begin{bmatrix}1.5\\1.5\end{bmatrix} \Rightarrow v_{\hat{\mathbf{w}}} = \frac{1}{c}c\begin{bmatrix}1.5\\1.5\end{bmatrix} = \begin{bmatrix}1.5\\1.5\end{bmatrix}$$

s.t. minimal MSBE:

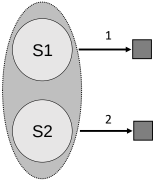
$$\overline{BE}(\hat{\mathbf{w}}) = ||\bar{\delta}_{\hat{\mathbf{w}}}||^2_\mu = (1 + \gamma 0 - 1.5)^2 \cdot 0.5 + (2 + \gamma 0 - 1.5)^2 \cdot 0.5 = 0.25$$

Q 8.2 *Exam Question: Function approximation



b) For the shown MDP, what is the minimal mean squared projected Bellman error? Why?

Q 8.2 *Exam Question: Function approximation



We are in the case of linear function approximation. Therefore we know that MSPBE = 0 at the TD fixed point, so the answer must be 0. Alternative: If we project the bellman errors [-0.5, 0.5] on a basis consisting of just the constant feature, we get projected errors of [0,0].


WELL DONE!

Tutorial 9 Overview

1. Off-policy RL with approximation exercises
2. Ask anything about HW4
   - Questions 7.4, 8.3-8.4

Ask anything about HW4

- 7.4: Theory
- 8.3: Theory
- 8.4: Coding (+ Little bit of theory)

That's it!

See you tomorrow