

# Exercise Set 4 - Reinforcement Learning

Control with approximation and policy gradients

Giulio Starace - 13010840

October 4, 2022

## Homework: Geometry of linear value-function approximation (Application)

1. To compute the Bellman error vector after initialization, we compute the Bellman error for each state. We first recall the definition of the Bellman operator  $B^\pi$ :

$$(B^\pi v)(s) \doteq \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) [r + \gamma v(s')]. \quad (1)$$

We can plug this into the definition of the Bellman error:

$$\begin{aligned} \bar{\delta}_w(s) &= B^\pi v_w - v_w \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) [r + \gamma v_w(s')] - v_w(s). \end{aligned}$$

For  $s_0$ , we have a single action available that is always taken, so our  $\sum_a \pi(a|s)$  term disappears and we are left with:

$$\bar{\delta}_w(s) = \sum_{s',r} p(s', r|s) [r + \gamma v_w(s')] - v_w(s) \quad (2)$$

Our action always leads to the same state, with the same reward (of 0), so we can simplify further and write

$$\bar{\delta}_w(s) = \gamma v_w(s') - v_w(s). \quad (3)$$

Finally, we have that  $v_{w,s} = w \cdot \phi_s$ , so that we can write

$$\bar{\delta}_w(s) = \gamma w \cdot \phi_{s'} - w \cdot \phi_s. \quad (4)$$

The same arguments can be applied to  $s_1$ , so we can write the Bellman error *vector* as

$$\text{BE}(w) = (\bar{\delta}_w(s_0), \bar{\delta}_w(s_1))^T = (\gamma w \cdot \phi_{s_1} - w \cdot \phi_{s_0}, \gamma w \cdot \phi_{s_0} - w \cdot \phi_{s_1})^T. \quad (5)$$

We can plug in our values  $w = 1$ ,  $\phi_{s_0} = 2$ ,  $\phi_{s_1} = 1$ , and  $\gamma = 1$  and obtain

$$\text{BE}(w) = (1 \cdot 1 - 1 \cdot 2, 1 \cdot 2 - 1 \cdot 1)^T = (-1, 1)^T. \quad (6)$$

2. The Mean Squared Bellman Error  $\overline{\text{BE}}(w)$  is the measure of the overall error in the value function, computed by taking the  $\mu$  weighted norm of the Bellman error vector. Here,  $\mu$  is a distribution  $\mu : \mathcal{S} \rightarrow [0, 1]$  specifying the extent to which each state is considered in the computation. Mathematically:

$$\overline{\text{BE}}(w) = \|\text{BE}(w)\|_{\mu}^2 = \sum_s \mu(s) \bar{\delta}_w(s)^2. \quad (7)$$

In our case, this would be expressed as

$$\overline{\text{BE}}(w) = \mu(s_0) \cdot (-1)^2 + \mu(s_1) \cdot 1^2 = \mu(s_0) + \mu(s_1) = \sum_s \mu(s). \quad (8)$$

3. The target values  $B^{\pi}v_w$  we found in question 1. are 1 for  $s_0$  and 2 for  $s_1$ . The  $w$  that results in the value function that is closest can be applied using a least-squares regression:

$$\begin{aligned} \beta &= \arg \min_{\beta} \|\mathbf{Y} - w\mathbf{X}\|^2 \\ w &= \arg \min_w \|(1, 2)^T - w(\phi_{s_0}, \phi_{s_1})^T\|^2 \\ &= \arg \min_w \|(1, 2)^T - w(2, 1)^T\|^2. \end{aligned} \quad (9)$$

This has a closed-form solution:

$$\beta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}, \quad (10)$$

which for our numbers gives  $w = 4/5$ .

4. The plot of  $v_w$ ,  $B^{\pi}v_w$  and  $\Pi B^{\pi}v_w$  is shown in Figure 1. TODO

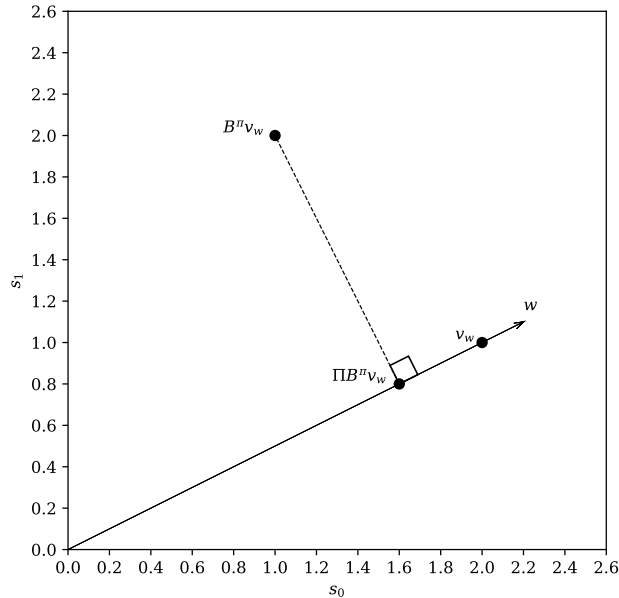


Figure 1: TODO

## Homework: Coding Assignment - Deep Q Networks

1. Coding answers have been submitted on codegra under the group “stalwart cocky sawly”.
2. hello world

## Homework: REINFORCE

1. (a) The update to the policy parameter  $\theta$  under classical REINFORCE after the  $e$ th episode is given by

$$\theta_{e+1} \leftarrow \theta_e + \alpha \widehat{\nabla_{\theta} J}, \quad (11)$$

where  $\widehat{\nabla_{\theta} J}$  is an estimate of the gradient the expectation of the return  $G(\tau_e)$  for a given trajectory  $\tau$ , and  $\alpha$  is the learning rate. This is given by

$$\widehat{\nabla_{\theta} J} = \frac{1}{N} \sum_{i=1}^N \left[ G(\tau_i) \sum_{t=0}^T \nabla_{\theta} \log \pi_{\theta}(a_t^i | s_t^i) \right], \quad (12)$$

where  $N$  is the number of trajectories observed, and  $\pi_{\theta}$  is the policy we are parametrising with  $\theta$ . Because we are interested in the update per-episode,  $N = 1$  and our equation simplifies to

$$\widehat{\nabla_{\theta} J} = G(\tau_e) \sum_{t=0}^T \nabla_{\theta} \log \pi_{\theta}(a_t^i | s_t^i) \quad (13)$$

Because in our case  $\theta$  is split into  $\theta_a$  and  $\theta_b$ , where an action in state  $A$  only depends on  $\theta_a$  and the action in state  $B$  only depends on  $\theta_b$ , we can treat the update to each parameter separately, obtaining

$$\theta_a^{e+1} \leftarrow \theta_a^e + \alpha \widehat{\nabla_{\theta_a} J} \quad (14)$$

$$\theta_b^{e+1} \leftarrow \theta_b^e + \alpha \widehat{\nabla_{\theta_b} J}. \quad (15)$$

$\widehat{\nabla_{\theta_{a/b}} J}$  takes the same form as (13), but now only depends on the actions taken in state  $A$  or  $B$  respectively. This leaves us with

$$\widehat{\nabla_{\theta_a} J} = G(\tau_e) \sum_{t:s_t=A}^T \nabla_{\theta_a} \log \pi_{\theta}(a_t^i | A, \theta_a) \quad (16)$$

$$\widehat{\nabla_{\theta_b} J} = G(\tau_e) \sum_{t:s_t=B}^T \nabla_{\theta_b} \log \pi_{\theta}(a_t^i | B, \theta_b) \quad (17)$$

We can plug this into our updates for each part of  $\theta$  and obtain

$$\theta_a^{e+1} \leftarrow \theta_a^e + \alpha G(\tau_e) \sum_{t:s_t=A}^T \nabla_{\theta_a} \log \pi_{\theta}(a_t^i | A, \theta_a) \quad (18)$$

$$\theta_b^{e+1} \leftarrow \theta_b^e + \alpha G(\tau_e) \sum_{t:s_t=B}^T \nabla_{\theta_b} \log \pi_{\theta}(a_t^i | B, \theta_b), \quad (19)$$

where the trajectory  $\tau_e$  will be different for each episode  $e$ . We can plug in the values given from our sampled episodes and obtain the following four updates:

$$\theta_a^1 \leftarrow \theta_a^0 + \alpha \cdot 175 \cdot \nabla_{\theta_a} [\log \pi_{\theta}(1|A, \theta_a) + \log \pi_{\theta}(2|A, \theta_a)] \quad (20)$$

$$\theta_a^2 \leftarrow \theta_a^1 - \alpha \cdot 60 \cdot \nabla_{\theta_a} [\log \pi_{\theta}(1|A, \theta_a) + \log \pi_{\theta}(2|A, \theta_a)] \quad (21)$$

$$\theta_b^1 \leftarrow \theta_b^0 + \alpha \cdot 175 \cdot \nabla_{\theta_b} [\log \pi_{\theta}(2|B, \theta_b) + \log \pi_{\theta}(2|B, \theta_b)] \quad (22)$$

$$\theta_b^1 \leftarrow \theta_b^0 - \alpha \cdot 60 \cdot \nabla_{\theta_b} [\log \pi_{\theta}(1|B, \theta_b) + \log \pi_{\theta}(1|B, \theta_b)] \quad (23)$$

- (b) When using REINFORCE/G(MO)MDP, we now have a different general estimate of the gradient of the expectation of the return,  $\widehat{\nabla J}$ , given by

$$\widehat{\nabla_{\theta} J} = \frac{1}{N} \sum_{i=1}^N \left( \sum_{t=1}^T r_t \sum_{t'=1}^t \nabla_{\theta} \log \pi_{\theta}(a_{t'}|s_{t'}) \right), \quad (24)$$

where  $r_t$  is the reward at time  $t$  for episode  $i$ . We can apply the same reasoning used in [1a](#) and obtain the update rule per episode for each split of  $\theta$ :

$$\theta_a^{e+1} \leftarrow \theta_a^e + \alpha \sum_{t=1}^T r_t \sum_{t' \in \{z: s_z = A \wedge z \leq t\}} \nabla_{\theta_a} \log \pi_{\theta}(a_{t'}|A, \theta_a) \quad (25)$$

$$\theta_b^{e+1} \leftarrow \theta_b^e + \alpha \sum_{t=1}^T r_t \sum_{t' \in \{z: s_z = B \wedge z \leq t\}} \nabla_{\theta_b} \log \pi_{\theta}(a_{t'}|B, \theta_b) \quad (26)$$

where the rewards  $r_t$  will be different depending on the episode. We can plug in the values given from our sampled episodes and obtain the following four updates:

$$\theta_a^1 \leftarrow \theta_a^0 + \alpha \cdot \nabla_{\theta_a} [175 \cdot \log \pi(1|A, \theta_a) - 25 \cdot \log \pi(2|A, \theta_a)] \quad (27)$$

$$\theta_a^2 \leftarrow \theta_a^1 + \alpha \cdot \nabla_{\theta_a} [-60 \cdot \log \pi(1|A, \theta_a) + 40 \cdot \log \pi(2|A, \theta_a)] \quad (28)$$

$$\theta_b^1 \leftarrow \theta_b^0 + \alpha \cdot \nabla_{\theta_b} [-8 \cdot \log \pi(2|B, \theta_B)] \quad (29)$$

$$\theta_b^2 \leftarrow \theta_b^1 + \alpha \cdot \nabla_{\theta_b} [30 \cdot \log \pi(1|B, \theta_B)] \quad (30)$$

2. hello world

3. hello world

4. hello world

5. hello world