REINFORCEMENT LEARNING

Homework 2

Author
Paolo Barba, 1885324
Sapienza, University of Rome

Student discussion: For this homework I discuss with Maria Vittoria Vestini.

Exercise 1

Given the following Q-Table:

$$Q(s,a) = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} Q(1,1) & Q(1,2) \\ Q(2,1) & Q(2,2) \end{pmatrix}$$

Given $\alpha = 0,1 \gamma = 0,5$ and the experience $(s,a,r,s') = (1,2,3,2)$.Consider $a' = \pi_\epsilon(s') = 2$ Compute the update for *Q-learning* and *SARSA*.

SARSA In the *SARSA* algorithm we have to compute the updating the $Q - value$ of a state and action $Q(s,a)$ as the following:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)]$$

given our data as: $s = 1, a = 2, r = 3, s' = 2$ by plug them into the formula we obtain:

$$Q(1,2) = 2 + 0.1[3 + 0.5 \cdot 4 - 2] = 2.3$$

Q-Learning In the *Q-Learning* algorithm we have to compute the updating the $Q - value$ of a state and action $Q(s,a)$ as the following:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma max_a Q(s',a) - Q(s,a)]$$

.

given our data as: $s = 1, a = 2, r = 3, s' = 2$ by plug them into the formula we obtain: Since $max_a Q(s',a) = max\{Q(2,1); Q(2,2)\} = max\{3; 4\} = 4$

$$Q(1,2) = 2 + 0.1[3 + 0.5 \cdot 4 - 2] = 2.3$$

Exercise 2

Prove that the n-step error can also be written as a sum of TD errors if the value estimates don't change from step to step.

What we want to prove is the following:

$$G_{t:t+n} - V_{t+n-1}(S_t) = \sum_{k=t}^{t+n-1} \gamma^{k-t}\delta_k$$

where :

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$

and

$$\delta_k = r_{k+1} + \gamma \cdot V(s_{k+1}) - V(s_k)$$

where in the left side of the equation we are representing the the n-step error and in the right side we are representing the sum of the TD-errors.

Assume the value estimates don't change from step to step means that:

$$V(S_t) = V(S_{t+n}) = V \ \forall \ n \in \mathbf{N}$$

We can develop the summation using the assumptions we have seen above.

$$\begin{aligned}
&\sum_{k=t}^{t+n-1} \gamma^{k-t}(r_{k+1} + \gamma \cdot V - V) \\
&= r_{t+1} + \gamma V - V + \gamma(r_{t+2} + \gamma V - V) + \gamma^2((r_{t+3} + \gamma V - V)) + \cdots \\
&\quad + \gamma^{n-2}(r_{t+n-1} + \gamma V - V) + \gamma^{n-1}(r_{t+n} + \gamma V - V) \\
&= r_{t+1} + \cancel{\gamma V} - V + \gamma r_{t+2} + \gamma^2 V - \cancel{\gamma V} + \cdots + \\
&\quad \gamma^{n-2} r_{t+n-1} + \cancel{\gamma^{n-1} V} - \gamma^{n-1} V + \gamma^{n-1} r_{t+n} + \gamma^n V - \cancel{\gamma^{n-1} V}
\end{aligned}$$

is trivial that at every iteration of the sum the general $\gamma^i V$ factor simplify $i \in \{1 \ldots n-1\}$. We can write the expression as:

$$R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R^{t+n} + \gamma^n V - V$$

that is actually what is in the left size of the equation by applying the assumptions.

Report Code Exercises

Sarsa lambda

For the Sarsa Lambda exercise, we are asked to compute the epsilon greedy action and to update the Q and the eligibility tables. I developed the epsilon greedy action as the f

I developed the epsilon greedy action as the following:

$$P(a) = \begin{cases} (1-\epsilon) & \text{if } a = \text{argmax}_{a'} Q(a') \\ \epsilon & \text{for all other actions} \end{cases}$$

For the updating of the Q and elegibility tables, first I computed the TD ERROR as:

$$\delta = \alpha \cdot (r + \gamma \cdot Q[\text{next\_state}, \text{next\_action}] - Q[\text{state}, \text{action}])$$

Then we update Q as:

$$Q(s,a) \leftarrow Q(s,a) + \alpha * \delta * e(s,a)$$

and the elegibility as :

$$e(s,a)_t = \begin{cases} \gamma\lambda e_{t-1}(s,a) + 1 & \text{if } s = s_t \text{ and } a = a_t \\ \gamma\lambda e_{t-1}(s,a) & \text{otherwise} \end{cases}$$

Note that we can do the update without actually doing the for loop since every update is done point-wisely.

rbf

For the RBF exercise, we are asked to implement the RBF encoder. I computed on my own. First, create a grid of the environment space, then encode the state values using this formula we have seen in class

$$x(s) = \exp\left(-\frac{\|\text{s} - \text{centers}\|^2}{2 \cdot \sigma^2}\right)$$

where the centers are computed by griding the space environment into a 10 X 10 grid and the σ^2 is fixed to 0.1.

For the updating transition function I computed the TD error, update the elegibility trace and weights as:

$$\delta = r + (1 - \text{done}) \cdot [\gamma \cdot max(Q_{\text{s\_prime\_feats}}) - Q_{\text{s\_feasts}}[action]]$$

$$\begin{aligned} &e \leftarrow e\gamma\lambda \\ &e[\text{action}] \leftarrow e[\text{action}] + \text{s\_feats} \\ &w \leftarrow w + (\alpha\ \delta\ e[\text{action}]) \end{aligned}$$