Reinforcement Learning
Exercise 7 - Solution

Jonathan Schnitzler - st166934
Eric Choquet - st160996

June 22, 2024

1 Linear function approximation

a) Tabular linear function approximation The method of just writing down all values for each state $V(s)$ can be achieved via the identity as a feature vector, also known as one-hot coding, e.g.

$$x(s_i) = \begin{pmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{pmatrix}^T . \tag{1}$$

This results in the same number as features as states in the state space. The tabular value is now just the weight

$$V(s) = w_i \tag{2}$$

and the possible non-linear function f is the identity. The state-action value $Q(s, a)$ could be achieved in a similar way, for $x(s, a) = e_j$ and an according enumeration. The linear function approximation

$$\hat{v}(s, w) = \sum_{i=1}^{d} w_i f(x_i)$$

is a generalization of the tabular methods.

b) Update rule for Sarsa

1. Tabular case

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$

which is equivalent, shown in task a) to

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha[R_{t+1} + \gamma \hat{q}(S_{t+1}, A_{t+1}, \mathbf{w}) - \hat{q}(S_t, A_t, \mathbf{w}_t)]$$

since there $w_{t,i} = Q(s, a)$

2. Function approximation

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha[R_{t+1} + \gamma \hat{q}(S_{t+1}, A_{t+1}, \mathbf{w}) - \hat{q}(S_t, A_t, \mathbf{w}_t)]\nabla \hat{q}(S_t, A_t, \mathbf{w})$$

3. Linear function approximation

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha[R_{t+1} + \gamma \mathbf{w}^T \mathbf{x}(S_{t+1}, A_{t+1}) - \mathbf{w}^T \mathbf{x}(S_t, A_t)]\mathbf{x}(S_t, A_t)$$

2 Mountain Car

TODO: