CMPUT 365: Introduction to Reinforcement Learning, Winter 2023
Worksheet #11: Control with Approximation

Manuscript version: *#912caf-dirty* - 2023-04-23 01:25:08-06:00

Question 1 (*Exercise 10.2 S&B*)**.** Give pseudocode for semi-gradient one-step Expected Sarsa for control. You can build on the semi-gradient Sarsa code for this question.

Episodic Semi-gradient Sarsa for Estimating $\hat{q} \approx q_*$

Input: a differentiable action-value function parameterization $\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^d \to \mathbb{R}$
Algorithm parameters: step size $\alpha > 0$, small $\varepsilon > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)

Loop for each episode:
  $S, A \leftarrow$ initial state and action of episode (e.g., ε-greedy)
  Loop for each step of episode:
    Take action A, observe R, S'
    If S' is terminal:
      $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R - \hat{q}(S, A, \mathbf{w})\big]\nabla\hat{q}(S, A, \mathbf{w})$
      Go to next episode
    Choose A' as a function of $\hat{q}(S', \cdot, \mathbf{w})$ (e.g., ε-greedy)
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R + \gamma\hat{q}(S', A', \mathbf{w}) - \hat{q}(S, A, \mathbf{w})\big]\nabla\hat{q}(S, A, \mathbf{w})$
    $S \leftarrow S'$
    $A \leftarrow A'$

Question 2. (*Exercise 10.1 S&B*) We have not explicitly considered or given pseudocode for any Monte Carlo methods in this chapter. What would they be like? Why is it reasonable not to give pseudocode for them? How would they perform on the Mountain Car task?

Question 3. (*Exercise 10.3 S&B*) Why do the results shown in Figure 10.4 have higher standard errors at large n than at small n?

Question 4. How would you use optimistic value initialization, for Sarsa with a tile coding + linear function approximator? Assume you have a two dimensional input, and you use m tilings each with $n \times n$ tiles (i.e. m grids of size $n \times n$) resulting in mn^2 features. What size is your weight vector? And how do you initialize your weights to ensure that you have optimistic initial values? Assume the maximum reward is $R_{\max}$ and we use a $\gamma < 1$.
