



DEPARTMENT OF COMPUTER SCIENCE ST. FRANCIS XAVIER UNIVERSITY

St. Francis Xavier University
Department of Computer Science

CSCI-531 - Reinforcement Learning Practice Exercises: On-Policy Prediction and Function Approximation

Part I: Function Approximation Fundamentals

1. Tabular vs Function Approximation

Consider a robot navigation task in a continuous 2D environment where the robot's state is represented by its position (x, y) with $x, y \in [0, 100]$.

- Explain why tabular methods are impractical for this problem. Calculate the number of states if we discretize the space into 1-unit grid cells.
- Design a simple linear function approximation for this robot's value function. Define the feature vector $\mathbf{x}(s)$ and explain your choice.
- If the robot's goal is at position $(100, 100)$ and it receives a reward of -1 per step plus $+100$ for reaching the goal, what would be reasonable initial values for $\mathbf{w} = [w_1, w_2, w_3]^T$?

2. State Distribution and MSVE

Consider a simple 3-state MDP with states $S = \{s_1, s_2, s_3\}$ and the following information:

- True values: $v_\pi(s_1) = 10, v_\pi(s_2) = 5, v_\pi(s_3) = 15$
 - Approximate values: $\hat{v}(s_1, \mathbf{w}) = 8, \hat{v}(s_2, \mathbf{w}) = 7, \hat{v}(s_3, \mathbf{w}) = 12$
 - State distribution: $\mu(s_1) = 0.5, \mu(s_2) = 0.3, \mu(s_3) = 0.2$
- Calculate the Mean Squared Value Error (MSVE) for this approximation.
 - How would the MSVE change if we used a uniform distribution $\mu(s_1) = \mu(s_2) = \mu(s_3) = 1/3$ instead?
 - Explain why the choice of state distribution $\mu(s)$ matters for learning.

Part II: Gradient Methods

3. Gradient Descent Fundamentals

- (a) For the function $f(\mathbf{w}) = w_1^2 + 2w_2^2 - 4w_1 + 6w_2 + 10$, compute the gradient $\nabla f(\mathbf{w})$.
- (b) Starting from $\mathbf{w}_0 = [3, -1]^T$ with learning rate $\alpha = 0.1$, perform three steps of gradient descent.

4. Stochastic Gradient Descent in RL

Consider a simple 2-state MDP where we want to approximate $v_\pi(s)$ using linear function approximation with features:

- $\mathbf{x}(s_1) = [1, 0]^T$, $\mathbf{x}(s_2) = [0, 1]^T$
 - True values: $v_\pi(s_1) = 3$, $v_\pi(s_2) = 7$
- (a) Write the linear approximation formula and explain what the weight vector represents in this case.
 - (b) Starting with $\mathbf{w}_0 = [0, 0]^T$ and $\alpha = 0.2$, perform two SGD updates if we observe samples: $(s_1, v_\pi(s_1))$ then $(s_2, v_\pi(s_2))$.
 - (c) What would happen if we continued this process indefinitely with the same sampling pattern?

Part III: Prediction Algorithms

5. Gradient Monte Carlo

Consider a simple episodic task where an agent can be in states $\{s_1, s_2, s_3\}$ and always follows the episode: $s_1 \rightarrow s_2 \rightarrow s_3$ (terminal) with rewards $r_1 = 1, r_2 = 2, r_3 = 5$. Use $\gamma = 0.9$.

- (a) Calculate the true returns G_0, G_1, G_2 for this episode.
- (b) Using linear function approximation with features $\mathbf{x}(s_1) = [1, 0]^T$, $\mathbf{x}(s_2) = [1, 1]^T$, $\mathbf{x}(s_3) = [0, 1]^T$, write the Gradient Monte Carlo updates for this episode with $\alpha = 0.1$ and initial weights $\mathbf{w}_0 = [0, 0]^T$.
- (c) What are the final approximate values $\hat{v}(s_1), \hat{v}(s_2), \hat{v}(s_3)$ after this episode?

6. Semi-Gradient TD(0)

Using the same MDP setup as the previous question, but now we'll use Semi-Gradient TD(0) for online learning.

- (a) Write the Semi-Gradient TD(0) update equation for linear function approximation.
- (b) Starting with $\mathbf{w}_0 = [0, 0]^T$ and $\alpha = 0.1$, perform the TD(0) updates for the transitions $(s_1, 1, s_2)$ and $(s_2, 2, s_3)$.
- (c) Compare the convergence properties of Gradient Monte Carlo vs Semi-Gradient TD(0).