Assignment 2: Linear Function Approximation

Bernie Cassidy

February 28, 2026

Section 2: GridWorld with Terminal States

Question 2.1

The values for states 2 and 3 do not appear because they are impossible to reach using the optimal policy. If we start at state 0, and our policy tells us to move left from state 1 and up from state 6, our probability of reaching state 2 or 3 is zero so they will not appear in our simulation.

Question 2.2

In the previous assignment, the reward was associated with transitioning into a state. In the current assignment, the reward is associated with the transition from a reward state to an explicit terminal state (index 12).

Previous Expression: The value was defined as the expected reward based on transition probabilities T:

$$V(s) = \sum_{s' \in S} T(s, a, s') R(s, a, s')$$

For states 7 and 11, since they could only transition into themselves:

$$V(7) = 1.0 \times (-1) = -1, \quad V(11) = 1.0 \times (1) = 1$$

Current Expression: The value is now defined by the transition to the terminal state s_{12}:

$$V(s) = R(s, a, s_{12}) + \gamma V(s_{12})$$

Since the terminal state has an intrinsic value of zero ($V(s_{12}) = 0$), the expression simplifies to:

$$V(7) = -1 + \gamma(0) = -1, \quad V(11) = 1 + \gamma(0) = 1$$

Why they remain the same: The values remain nearly identical because the expected reward of entering a rewarding state (previous MDP) is mathematically equivalent to the deterministic reward of exiting that same state into a zero-value terminal state (current MDP). The introduction of state 12 acts as a formal "sink" but does not change the total discounted returns for the preceding states.

—

Section 3: Monte Carlo and TD Learning

Question 3.1: Derivation of Online Monte Carlo Update Rule

We aim to minimize the Mean Squared Value Error (VE) objective:

$$\overline{VE}(\mathbf{w}) = \mathbb{E}_\pi \left[(v_\pi(s) - \hat{v}(s, \mathbf{w}))^2 \right]$$

We use the return G_t as an unbiased estimate of $v_\pi(s_t)$. For linear function approximation, our estimate is $\hat{v}(s_t, \mathbf{w}) = \mathbf{w}^\top \phi(s_t)$. The objective for a single sample is:

$$J(\mathbf{w}) = \frac{1}{2}(G_t - \mathbf{w}^\top \phi(s_t))^2$$

We take the gradient with respect to the weight vector $\mathbf{w}$:

$$\nabla_{\mathbf{w}} J(\mathbf{w}) = \nabla_{\mathbf{w}} \left[\frac{1}{2}(G_t - \mathbf{w}^\top \phi(s_t))^2 \right]$$

Applying the chain rule:

$$\nabla_{\mathbf{w}} J(\mathbf{w}) = (G_t - \mathbf{w}^\top \phi(s_t)) \cdot \nabla_{\mathbf{w}} (G_t - \mathbf{w}^\top \phi(s_t))$$

We identify dependencies:

- G_t is the observed return from the environment; it does not depend on $\mathbf{w}$.
- $\phi(s_t)$ is the feature vector for state s_t; it does not depend on $\mathbf{w}$.
- The gradient $\nabla_{\mathbf{w}} (\mathbf{w}^\top \phi(s_t)) = \phi(s_t)$.

Thus:

$$\nabla_{\mathbf{w}} J(\mathbf{w}) = (G_t - \hat{v}(s_t, \mathbf{w})) \cdot (-\phi(s_t)) = -(G_t - \hat{v}(s_t, \mathbf{w}))\phi(s_t)$$

Step 3: Stochastic Gradient Descent Update The update rule is $\mathbf{w}_{\tau+1} = \mathbf{w}_\tau - \alpha \nabla_{\mathbf{w}} J(\mathbf{w})$. Substituting the gradient:

$$\mathbf{w}_{\tau+1} = \mathbf{w}_\tau + \alpha(G_t - \hat{v}(s_t, \mathbf{w}_\tau))\phi(s_t)$$

Question 3.2

Linear function approximation generalizes across states. Because states share features (like the same X or Y coordinate), an update to one state affects the values of all states that share those features, effectively "filling in" the grid even with sparse samples.

Question 3.3

The value of the -1 state is smaller in magnitude than expected due to the visitation distribution $\mu_\pi(s)$. Since the policy π likely favors reaching the $+1$ state, the -1 reward state is visited infrequently. Additionally, feature overlap with neighboring "safe" states can cause interference, where positive updates from visited neighbors counteract the rare negative updates from the -1 state.

Question 3.4

If the feature vector $\phi(s)$ is a one-hot encoding of state indices, the value function $\hat{v}(s, \mathbf{w}) = \mathbf{w}^\top \phi(s)$ simplifies to a direct indexing operation, $\hat{v}(s, \mathbf{w}) = w_s$. In this case, the gradient $\nabla_{\mathbf{w}} \hat{v}(s, \mathbf{w})$ is non-zero only for the element corresponding to the current state s.

Consequently, the online MC update rule:

$$\mathbf{w}_{\tau+1} = \mathbf{w}_\tau + \alpha(G_t - \hat{v}(s_t, \mathbf{w}_\tau))\phi(s_t)$$

collapses into the state-specific scalar update:

$$w_{s_t} \leftarrow w_{s_t} + \alpha(G_t - w_{s_t})$$

Using one-hot features in this way removes the benefit of generalization. Because features do not overlap across states, the agent cannot learn about unvisited states from visited ones.

—

Section 6: CartPole

Question 6.1

The optimal value function for CartPole is non-linear (often parabolic) with respect to angle and velocity. A linear model can only represent a flat hyperplane, which is insufficient to capture the sharp change in value that occurs when the pole exceeds a critical angle.

Question 6.2

I used a combination of:

- A bias term (1.0).
- Normalized raw features (s_pos, s_vel, s_angle, s_angle_vel).
- Polynomial features (s_pos^2, s_angle^2).
- Interaction terms (s_angle * s_angle_vel, s_pos * s_angle).

Question 6.3

The marginal plots showed curved (non-linear) relationships between states and Q-values. The curves in the Angle vs. Q-value plot suggested that squared terms (like θ^2) were necessary to model the decrease in value as the pole moves away from 0.