CS 747 (Autumn 2025) Week 9 Test (Batch 1)

5.35 p.m. – 6.00 p.m., October 14, 2025, LA 001

Name: _____ Roll number: _____

Note. There is one question in this test. You can use the space on both pages for your answer. Draw a line (either vertical or horizontal) and do all your rough work on one side of it.

Question 1. Consider an agent interacting with MDP (S, A, T, R, γ), with notation as usual. The task is continuing, and $\gamma \in (0, 1)$.

The agent executes a deterministic policy π. It aims to estimate the value function of π using linear function approximation. The agent makes updates according to the "Linear TD(0)" update rule.

Write down the formula for the "mean-squared value error", and provide the steps to demonstrate that Linear TD(0) "is in the spirit of" performing stochastic gradient descent to minimise this error. Introduce and define the relevant quantities, and explicitly specify any assumptions made. Also explain why Linear TD(0) does not exactly correspond to stochastic gradient descent. [3 marks]

Answer 1. Let us suppose that the agent uses a d-dimensional feature vector $\phi(s)$ for state $s \in S$, and that it approximates the value of s by $w \cdot \phi(s)$, where $w \in \mathbb{R}^d$ is the weight vector. The mean-squared value error of this approximation, MSVE(w), is given by

$$\text{MSVE}(w) = \frac{1}{2} \sum_{s \in S} \mu^\pi(s) \left(V^\pi(s) - w \cdot \phi(s)\right)^2,$$

where $\mu^\pi(s)$ is the stationary distribution of π (over S). The factor of $\frac{1}{2}$ is a convention; it is not necessary for the definition. The gradient of MSVE for $w \in \mathbb{R}^d$ is

$$\nabla_w \text{MSVE}(w) = -\sum_{s \in S} \mu^\pi(s) \left(V^\pi(s) - w \cdot \phi(s)\right) \phi(s).$$

If an agent goes through a state-reward trajectory $s^0, r^0, s^1, r^1, \ldots, s^t, r^t, \ldots$, while taking actions according to π, then for large t, $s^t \sim \mu^\pi$. Hence, we have

$$\mathbb{E}[-\left(V^\pi(s^t) - w \cdot \phi(s^t)\right) \phi(s^t)] = \nabla_w \text{MSVE}(w).$$

In theory, stochastic gradient descent (in the space of the weights w) can be performed by taking steps against the direction $-\left(V^\pi(s^t) - w \cdot \phi(s^t)\right)$. However, since $V^\pi(s^t)$ is not known, the agent cannot perform gradient descent using this quantity. On the other hand, an empirical return could be used in place of $V^\pi(s^t)$. If the Monte Carlo return $r^t + \gamma r^{t+1} + \gamma^2 r^{t+2} + \ldots$ is used, then the update would indeed correspond to stochastic gradient descent. In Linear TD(0), the agent instead uses the *bootstrapping* return $r^t + \gamma w \cdot \phi(s^{t+1})$ as a proxy for $V^\pi(s^t)$. The update is

$$w_{\text{new}} \leftarrow w_{\text{old}} + \alpha \left(r^t + \gamma w_{\text{old}} \cdot \phi(s^{t+1}) - w_{\text{old}} \cdot \phi(s^t)\right) \phi(s^t),$$

where α is the learning rate. Since $\mathbb{E}[r^t + \gamma w \cdot \phi(s^{t+1})]$ is in general not equal to $V^\pi(s^t)$, the Linear TD(0) update does *not* exactly correspond to stochastic gradient descent to minimise MSVE(w).

CS 747 (Autumn 2025) Week 9 Test (Batch 2)

6.15 p.m. – 6.40 p.m., October 14, 2025, LA 001

Name: _____ Roll number: _____

Note. There is one question in this test. You can use the space on both pages for your answer. Draw a line (either vertical or horizontal) and do all your rough work on one side of it.

Question 1. Consider an agent interacting with MDP (S, A, T, R, γ), with notation as usual. The set of states is $S = \{s_1, s_2, s_\top\}$, where $s_\top$ is the only terminal state. The task is *episodic*, and no discounting is applied (that is, $\gamma = 1$).

The agent executes a deterministic policy π. It uses the TD(n) update rule (that is, based on n-step returns) for estimating the value function of π, with $n = 2$. Initially, the value estimate for every state is set to 0. Updates are made to this estimate based on the data gathered, using a constant learning rate $\alpha = 0.25$ The following is the "state-reward" trajectory obtained in the first two episodes.

Episode 1: $s_1, 1, s_2, 0, s_1, -1, s_\top$.

Episode 2: $s_2, -1, s_\top$.

Write down the value function estimate after each successive learning update is made, up until the end of the second episode. [3 marks]

Answer 1. Corresponding to every time step in which an action is taken, an update is performed after 2 steps elapse, or the episode ends, whichever is earlier. For $t \geq 0$, let V^t denote the value estimate after t updates have been made. Note that there is no update made "across episodes", in the sense of a starting state in one episode being updated based on data gathered in the next episode.

Below we list V^t for $t = 0, 1, 2, 3, 4$; the estimate is V^3 after the first episode terminates, and V^4 after the second episode terminates. Since the terminal state has 0 value, we define $V(s_\top) \stackrel{\text{def}}{=} 0$ to be a constant (not updated while learning).

$$V^0(s_1) = 0;$$
$$V^0(s_2) = 0.$$

$$V^1(s_1) = (1-\alpha)V^0(s_1) + \alpha(1 + 0 + V^0(s_1)) = \frac{1}{4};$$
$$V^1(s_2) = V^0(s_2) = 0.$$

$$V^2(s_1) = V^1(s_1) = \frac{1}{4};$$
$$V^2(s_2) = (1-\alpha)V^1(s_2) + \alpha(0 + (-1) + V(s_\top)) = -\frac{1}{4}.$$

$$V^3(s_1) = (1-\alpha)V^2(s_1) + \alpha(-1 + V(s_\top)) = \frac{3}{16} - \frac{1}{4}. = -\frac{1}{16};$$
$$V^3(s_2) = V^2(s_2) = -\frac{1}{4}.$$

$$V^4(s_1) = V^3(s_1) = -\frac{1}{16};$$

$$V^4(s_2) = (1-\alpha)V^3(s_2) + \alpha(-1 + V(s_\top)) = -\frac{3}{16} - \frac{1}{4}. = -\frac{7}{16}.$$