CS 747 (Spring 2025) Week 10 Test (Batch 1)

5.35 p.m. – 6.00 p.m., March 27, 2025, LA 001

Name: _____ Roll number: _____

Note. There are two questions, one on each page. Provide your answer to each question in the space given below it. Draw a line (either vertical or horizontal) and do all your rough work on one side of it.

Question 1. Suppose the "Linear TD(1)" algorithm (that is, TD(1) with linear function approximation) is executed to evaluate a policy π on an MDP $M = (S, A, T, R, \gamma)$, with notation as usual. Assume that the MDP is ergodic and that the learning rate is handled appropriately. What is the relevant theoretical guarantee that applies? Feel free to define quantities based on M. [1 mark]

Answer 1. Let $\mu^\pi : S \to [0, 1]$ be the stationary distribution of π. Suppose $V^\pi(s)$ is being approximated as a linear function of a d-dimensional feature vector $\phi : S \to \mathbb{R}^d$. Then the coefficient vector $\mathbf{w} \in \mathbb{R}^d$ is assured (under suitable conditions) to converge to

$$\mathbf{w}^\star = \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^d} \sum_{s \in S} \mu^\pi(s) \{V^\pi(s) - \mathbf{w} \cdot \phi(s)\}^2.$$

In other words, at convergence the weight vector minimises the mean squared value error.

Question 2. Is tile coding a form of linear function approximation or of non-linear function approximation? Would it enjoy the theoretical guarantee mentioned in Question 1? Justify your claim of "linear" or "non-linear" by describing a typical application—such as the soccer task considered in class, or any control task presented in the textbook by Sutton and Barto. [2 marks]

Answer 2.

Tile coding is a form of linear function approximation, which indeed would enjoy the convergence guarantee described in Answer 1. It is "linear" in the sense that the function it represents is parameterised by a weight vector $\mathbf{w}$, whose dot product with a feature vector $\phi(s)$ for each state s (or each state-action pair in the case of control problems) approximates the target value (or action value) function. However, typically ϕ is *not* the raw input features. For example, in the soccer task discussed in class, the raw features are distances and angles. On the other hand, the features in ϕ are *tiles*, which are boolean features derived from the raw features. Hence, even though the approximated function is linear in ϕ, it is in general non-linear in the raw input features.

CS 747 (Spring 2025) Week 10 Test (Batch 2)

6.15 p.m. – 6.40 p.m., March 27, 2025, LA 001

Name: _____ Roll number: _____

Note. There are two questions, one on each page. Provide your answer to each question in the space given below it. Draw a line (either vertical or horizontal) and do all your rough work on one side of it.

Question 1. A designer who is interested in optimising a practically-relevant decision-making task abstracts it as an MDP $M = (S, A, T, R, \gamma)$, with notation as usual. Since S is very large (assume $|S| = 10^{100}$), it is impractical to apply an algorithm such as Q-learning using the usual "tabular" approach, since that would involve enumerating S. Assume A is finite and small—say single digit.

Suppose the developer decides to use Q-learning on M with *linear function approximation*. Describe in abstract terms the elements of the solution they must devise, specifying what has to be designed, and what has to be learned. How do these elements affect the quality of the learned solution? Feel free to define relevant quantities based on M. [2 marks]

Answer 1. The main element to design is a feature vector $\phi : S \times A \to \mathbb{R}^d$, where d is the dimension or the number of features. Usually d is much smaller than the number of states—say a few tens or thousands or millions. Each feature is associated with a learned weight, which must be stored in memory and updated using Q-learning. For $s \in S, a \in A$, $Q^\star(s, a)$ is approximated by $\mathbf{w} \cdot \phi(s, a)$, where $\mathbf{w} \in \mathbb{R}^d$ is the weight vector. Since features force different states to share values, they play a role in the sample efficiency as well as the eventual performance of learned behaviour.

Question 2. Suppose a suitable linear function approximation scheme is devised in Question 1. Is Q-learning guaranteed to converge, and if so, will it achieve *optimal* behaviour for M? Assume that learning and exploration rates are handled appropriately. You can also make reasonable assumptions about M, but state your assumptions clearly. [1 mark]

Answer 2. The combination of control, generalisation, and off-policy updating make Q-learning with function approximation difficult to analyse—there are no guarantees of convergence for the general case. In general one cannot even expect that the linear function approximation scheme can represent an action value function that would indice an optimal policy. In practice, success on control tasks is almost always an empirical matter—not the outcome of a theoretical guarantee.