AI 533 **Intelligent Agents and Decision Making** Winter 2025

OREGON STATE UNIVERSITY

School of Electrical Engineering and Computer Science

Instructor: Sandhya Saisubramanian

Total points: 100 Assignment 2: Monte Carlo, TD methods, and Actor-Critic Due date: Feb 24, 2025

Instructions: Collaboration is not allowed on any part of this assignment. It is acceptable to discuss concepts or clarify questions but you must document who you worked with for this assignment. Copying answers or reusing solutions from individuals/the internet is unacceptable and will be dealt with strictly. Solutions must be typed (hand written and scanned submissions will not be accepted) and saved as a .pdf file.

1. **(10 points)** After applying SARSA, can we estimate V^π upon its termination, based on the information available to the agent? If yes, write the equation to estimate V^π . If no, explain why.

2. **(10 points)** Does SARSA update equation change when we modify the reward notations ($R(s), R(s,a)$, or $R(s,a,s')$). Why or why not?

3. **(10 points)** Policy iteration algorithm consists of a policy evaluation phase and a policy improvement phase. If we replace dynamic programming policy evaluation with First Visit Monte Carlo prediction for policy evaluation, will policy iteration converge to an optimal policy in a finite number of iterations? Why or why not? Assume the agent has access to T and R.

4. **(20 points)** Consider actor-critic methods in tabular format. (i) We will replace the TD-critic module with that of an Oracle critic: the critic has an MDP model (unknown to the actor) and can calculate $V(s)$ accurately using Bellman equation. How does this affect the actor updates and convergence of the algorithm? (ii) We will now replace the TD-critic module with that of a lazy critic: it updates the values $V(s)$ in every other iteration (instead of every iteration). How does this affect the actor updates and convergence of the algorithm?

5. **(25 points)** Consider the following episodes of an MDP with five states and five actions. Each entry in the episode includes the state id, action id, and the corresponding reward observed. Discount factor $\gamma = 0.9$. (i) For each episode, calculate the returns of each state using *first-visit Monte Carlo* algorithm and calculate the value of each state $V(s)$ using both the episodes. (ii) For each episode, calculate the returns of each state using e*very-visit Monte Carlo* algorithm and calculate the value of each state $V(s)$ using both the episodes. Report your values as a table to facilitate faster grading.

   - Episode 1: $\{(s_1, a_1, 1), (s_1, a_1, 5), (s_2, a_4, 8), (s_3, a_5, 5), (s_4, a_1, 10)\}$
   - Episode 2: $\{(s_1, a_1, 5), (s_2, a_4, -1), (s_3, a_5, -2), (s_5, a_2, 1), (s_4, a_1, 10)\}$

6. **(25 points)** Consider two MDPs $\mathcal{M} = \langle S, A, T, R\rangle$ and $\mathcal{M}' = \langle S, A, T, R'\rangle$. The MDPs have the same state space, action space, transition functions, and discuount factor γ but differ in their reward functions such that $R'(s,a,s') = R(s,a,s') + F(s,a,s')$ where $F(s,a,s')$ is a bonus reward that could help speed up the learning process (also referred to as reward shaping). In our case, $F(s,a,s') = \gamma\phi(s') - \phi(s)$ for some arbitrary function $\phi : S \to \mathbb{R}$.

Consider running tabular Q-learning in each MDP $\mathcal{M}$ and $\mathcal{M}'$, with initial values $Q^0_{\mathcal{M}}(s,a) = q_{\text{init}} + \phi(s)$ and $Q^0_{\mathcal{M}'}(s,a) = q_{\text{init}}$, $\forall(s,a) \in S \times A$ and $q_{\text{init}} \in \mathbb{R}$. At any moment in time, the current Q-value of any state-action pair is always equal to its initial value plus some Δ value denoting the total change in the Q-value across all updates:

$$Q_{\mathcal{M}}(s,a) = Q^0_{\mathcal{M}}(s,a) + \Delta Q_{\mathcal{M}}(s,a) \tag{1}$$
$$Q_{\mathcal{M}'}(s,a) = Q^0_{\mathcal{M}'}(s,a) + \Delta Q_{\mathcal{M}'}(s,a) \tag{2}$$

Show that if $\boldsymbol{\Delta Q_{\mathcal{M}}(s,a) = \Delta Q_{\mathcal{M}'}(s,a)}$ for all $\boldsymbol{(s,a) \in S \times A}$, then these two Q-learning agents yield identical updates for any state-action pair. That is, both agents will converge to policies that behave identically, and the offset $\phi(s)$ will not affect action selection.

Hints:

- Equations (1) and (2) indicate that $\Delta Q_{\mathcal{M}}(s,a) = \alpha_M \delta_M$ and $\Delta Q_{\mathcal{M}'}(s,a) = \alpha_{M'} \delta_{M'}$ where α_M and $\alpha_{M'}$ denote the learning rates or step sizes and δ_M and $\delta_{M'}$ denotes the corresponding TD-errors.
- For ease, you can assume that $\alpha_M = \alpha_{M'}$.
- What you then need to show is $\delta_M = \delta_{M'}$. To do so, use Equations (1) and (2) to rewrite $Q_{\mathcal{M}}(s,a)$ and r_M in terms of $Q_{\mathcal{M}'}(s,a)$ and $r_{M'}$ or vice versa.