COMPSCI 687 Homework 6
(100 Points)

Instructions: Submit your answers as a single typed PDF file in Gradescope. You may use any software to create the PDF, and any reasonable formatting of your answers will suffice. You may collaborate with other students and may use generative AI to complete this assignment without reporting the collaboration or generative AI usage, but you are responsible for the accuracy of your answers.

Part One: Written (60 Points)

Assume finite MDPs, bounded rewards, and $\gamma < 1$. Keep $v(s_\infty) = 0$.

1. (30 Points) Fix a state $s \in \mathcal{S}$ and a policy π, and define

$$f_s(v) = \frac{1}{2}\mathbf{E}[\delta_t^2 \mid S_t = s; \pi]. \tag{1}$$

   Derive one stochastic gradient descent update for f_s using a transition S_t, A_t, R_t, S_{t+1} sampled from s under π. Give the update for the case $S_{t+1} \neq s$. Show the derivatives and write your answer as $\delta_t \leftarrow \cdots$, $v(S_t) \leftarrow \cdots$, and $v(S_{t+1}) \leftarrow \cdots$. How does this differ from TD?

2. (30 Points)

   (a) (10 Points) Prove that, when $v = v^\pi$, $\mathbf{E}[\delta_t \mid S_t = s; \pi] = 0$.

   (b) (20 Points) A researcher claims to have developed a new policy-evaluation algorithm that **(1)** converges to v^π for every MDP and policy and that **(2)** makes $\mathbf{E}[\delta_t^2 \mid S_t = s; \pi]$ converge to zero for every state s. Explain why these claims cannot both be true. Give an MDP and policy that demonstrate the problem, and compute the expected squared TD error at $v = v^\pi$.

Part Two: Programming (40 Points)

3. (40 Points) Implement *Temporal Difference (TD) Learning* from the course notes. Adapt your Homework 2 environment to read MDPs in the Homework 4 format and provide the following methods.

   - `newEpisode()`: Sample the initial state from d_0 and store it.
   - `getState()`: Return the current state.
   - `inTerminalState()`: Return whether the current state is s_∞. For this grid, it returns false at square 24 and true at state 25.
   - `step(a)`: Sample and store the next state, and return $R(s, a)$, where s is the state before the transition.

   Note that we will only consider MDPs where the reward R_t is deterministic given S_t and A_t, so you can use $R_t = R(S_t, A_t)$. We will also only consider MDPs that include s_∞, and where s_∞ is the last state. Use `grid5_MDP.txt` and `grid5_policy.txt` from `HW6_files.zip` (Homework 4 formats). They describe the following 5×5 gridworld and its uniform random policy:

   - Number the squares $0, \ldots, 24$ across rows, starting at the top left. Every episode starts at square 0; $\gamma = 0.9$.
   - Actions 0, 1, 2, 3 move up, down, left, right deterministically. A move off the grid stays in the same square.
   - Every action from square 24 (bottom right) enters $s_\infty = 25$ with reward 0. Every action from any other square gives reward -1.

   Run TD for **100 episodes in each of 100 independent trials (runs)**, using step size $\alpha = 0.01$. Ensure that each trial uses a different RNG seed. Initialize all state value estimates to zero at the start of each trial.

   In your PDF, report a 5×5 table of the final TD value estimates, averaged across the 100 runs. Arrange entries like the grid and round to two decimal places. Average each state's final estimate across runs, not across episodes.