Homework 2

ELEN E6885: Introduction to Reinforcement Learning

Due: October 15, 2019

Problem 1 (True or False, 10 Points)

Please indicate *True* or *False* for the following statements. No explanation is needed.

1. [2 pts] In non-stationary reinforcement learning problems, we usually use a constant step-size in the incremental update of a reinforcement learning algorithm, e.g., $Q_{k+1} = Q_k + \alpha[r_{k+1} - Q_k]$. Choose constant step size for nonstationary problem

2. [2 pts] To find the optimal policy for a given MDP, one can choose to compute and store state-value function $v(s)$ or action-value function $q(s, a)$. One argument in favor of $q(s, a)$ is that it needs to store fewer values.

3. [2 pts] In the Monte Carlo estimation of action values, if a deterministic policy π is used, in most cases one agent (with the same starting state) could not learn the values of all actions.

4. [2 pts] For a stationary problem, Sarsa converges with probability 1 to an optimal policy and action-value function as long as all state-action pairs are visited an infinite number of times and the policy converges in the limit to the greedy policy.

5. [2 pts] Both SARSA and Q-learning are on-policy learning algorithms. Q-Learning: off-policy algorithm
SARSA: on-policy algorithm

Problem 2 (Policy Iteration vs. Value Iteration, 10 Points)

Regarding the policy iteration and value iteration algorithms, answer the following questions.

1. [5 pts] Summarize the differences between policy iteration and value iteration algorithms.

2. [5 pts] Prove that given the same starting value function, one iteration of the policy iteration algorithm, including greedy policy improvement followed by **one** step of policy evaluation, will generate the same value function as one iteration of the value iteration algorithm.

Problem 3 (Model-Based MDP, 25 Points)

Consider the model-based MDP problem as shown in Fig. 1. The number above the line refers to the probability of taking that action and r is the reward for the corresponding action.

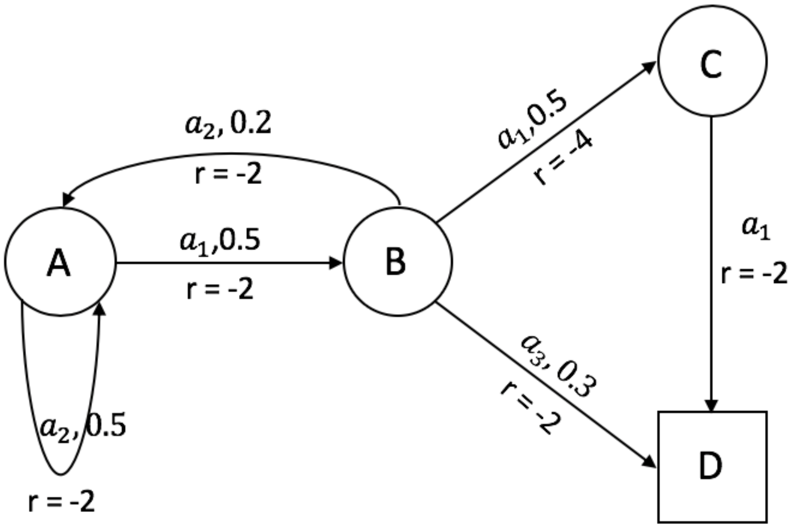

Figure 1: Model-based MDP.

1. [10 pts] Use matrix-form Bellman equation to find the values of all states ($\gamma = 0.5$).

2. [10 pts] Suppose that the initial values of all states are 0. The agent follows the policy given in the graph and assumes the discount factor $\gamma = 1$. Find the values of all states in the first two steps $k = 1$ and $k = 2$ in the algorithm of iterative policy evaluation.

3. [5 pts] Following the results of the previous question, update the values of all states for $k = 3$ using the value iteration method.

Problem 4 (**Convergence of Value Iteration, 15 Points**)

For the value iteration algorithm, the best policy at step k can be extracted as a result of the "max" operation (i.e., greedy action selection) based on the value function at step k.

1. [5 pts] Prove that if the value function at step k is the same as that of step $k + 1$, the best policy will never change with further iterations of the value function.

2. [10 pts] Provide an example to show that if the best policy at step k is the same as that of step $k + 1$, the best policy can still change with further iterations of the value function.

Problem 5 (**Last-Visit Monte Carlo, 15 Points**)

Similarly as in first-visit and every-visit Monte Carlo methods, we can estimate $v_\pi(s)$ by averaging over the discounted return starting from the *last* occurrence of s in each episode following policy π. We call it the *last-visit* Monte Carlo method. Prove by providing an example that the last-visit Monte Carlo method for estimating v_π is not guaranteed to converge to v_π for a finite MDP with bounded rewards and $\gamma \in [0, 1)$.

Problem 6 (**Small Grid World, 25 Points**)

Consider the small grid world MDP as shown in Fig. 2. The states are grid squares, identified by their row and column numbers (row first). The agent always starts in state $(1, 1)$. There are two terminal states, $(2, 3)$ with reward $+5$ and $(1, 3)$ with reward -5. Rewards are 0 in non-terminal states. (The reward for a state is received as the agent moves into the state.) The agent can take one action each time from up, down, left and right. However, the intended movement only happens with probability 0.8. And with probability 0.1 each, the agent ends up in one of the states perpendicular to the intended direction. For example, if the agent at $(1, 2)$ chooses to move up to $(2, 2)$, it can reach $(2, 2)$ only with probability 0.8. The agent may hit $(1, 1)$ or the terminal state (with reward -5) with equal probability 0.1. If a collision with a wall happens, the agent stays in the same state.

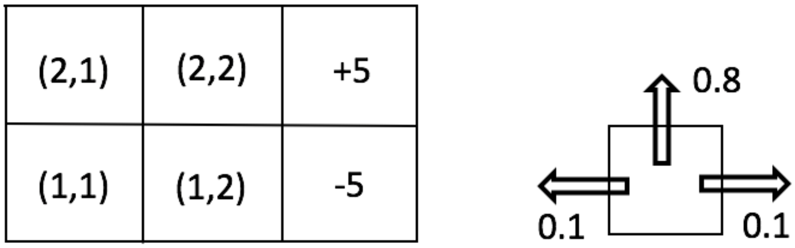

Figure 2: Grid world MDP and one transition example.

1. [5 pts] What is the optimal policy for this small grid world problem? Assume the discount factor $\gamma = 1$.

2. [8 pts] Suppose the agent knows the transition probabilities. Write down the first two rounds of (synchronous) *value iteration* updates for state $(1, 2)$ and $(2, 1)$, with a discount factor of 0.9. (Assume V_0 is 0 everywhere. Also, we assume the values of terminal states $V(1, 3) = V(2, 3) = 0$ for all iterations).

3. [4 pts] The agent starts with the policy that always chooses to go right, and executes the following three trials: 1) $(1, 1) - (1, 2) - (1, 3)$, 2) $(1, 1) - (1, 2) - (2, 2) - (2, 3)$, and 3) $(1, 1) - (2, 1) - (2, 2) - (2, 3)$. What are the Monte Carlo estimates for states $(1, 1)$ and $(2, 2)$, given these traces? Assume the discount factor $\gamma = 1$.

4. [8 pts] Using a learning rate of $\alpha = 0.1$, a discount factor of $\gamma = 0.9$, and assuming initial V values of 0, what updates does the TD(0)-learning agent make after trials 1) and 2) above?