CS 234 Midterm - Winter 2017-18

**Do not turn this page until you are instructed to do so.

Instructions

Please answer the following questions to the best of your ability. Read all the questions first before answering. You have **80** minutes to complete the exam. The exam is closed-book and closed-internet. Additionally use of all electronic items during the exam is prohibited. However, you may use an one sided one letter-sized page of notes as reference. All of the intended answers can be written well within the space provided. Good luck!

Stanford University Honor Code

The following is a statement of the Stanford University Honor Code:

1. The Honor Code is an undertaking of the students, individually and collectively: (1) that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading; (2) that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.
2. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.
3. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

By signing your name below, you acknowledge that you have abided by the Stanford Honor Code while taking this exam.

Signature:

Name:

SUNet ID:

Grading (For Midterm Graders Only)

Question #	1	2	3	4	5	6
Maximum Points	15	14	10	10	8	15
Student Grade						

Question 1 – True/False and Short answer [15pts]

In the question below $V^*(s)$ is the optimal value function in state s, π is a policy and V^π is its value function. We use the subscript t to indicate their dependence on the timestep for finite horizon MDPs. We also indicate with H the horizon of the MDP.

A) Circle True or False. Read each statement completely before answering. [1 pt each]

1. True False In a *discounted infinite* horizon MDP $V^*(s) \geq V^\pi(s)$ for all states s and policies π.
2. True False In an *undiscounted finite* horizon MDP $V_t^*(s) \geq V_t^\pi(s)$ for all states s, policies π and timesteps $1 \leq t \leq H$.
3. True False Suppose a MDP $M = (S, A, P, R, \gamma)$ with finite state space, finite action space, finite rewards and $\gamma = 1$, contains a terminal state S_{end} that ends the episode, such that for every state s and action a there is a positive (non-zero) probability to reach S_{end}. Then the value function V^π for every policy π must be finite for all states s.
4. True False For a finite horizon MDP, we require at most $H+1$ iterations of value iteration to compute the optimal policy
5. True False For a given MDP, the optimal policy π^* is always unique.
6. True False Due to the maximization bias, Q-learning may or may not converge to the optimal actions in the limit of infinite episodes, but Double-Q learning will always converge.
7. True False The Universal Approximation Theorem (Hornik, 1991), guarantees that for a hidden layer neural network with a linear output unit, any RL algorithm can converge to the optimal parameter values given enough hidden units.

B) For the next few questions answer in 2 - 3 sentences. [2 pts each]

1. In an infinite horizon MDP with S states and A actions, how many *deterministic* policies are there ? How many *stochastic* policies are there ?

2. Is the REINFORCE algorithm guaranteed to converge to the optimal policy if the policy class can represent the optimal policy ? Explain briefly.

3. Under what conditions does SARSA for finite-state and finite-action MDPs converge to the optimal action value ?

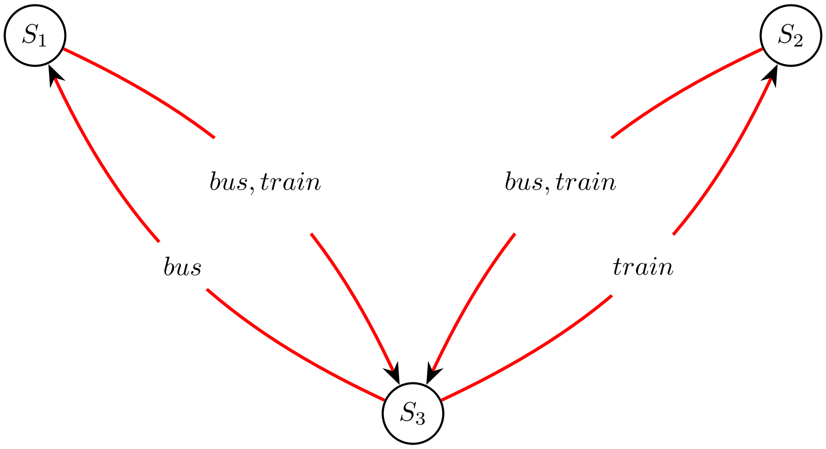
4. State the principle of maximum entropy in the context of inverse reinforcement learning.

Question 2 – Commuting from Stanford [14 pts]

*** With thanks to Andrew McCallum's PhD thesis for the underlying problem.*

Every weekend you travel from Stanford to either San Francisco or San Jose, and return at the end of the weekend back to Stanford. Since this is a repeated activity, we want to model it as an infinite horizon process. The goal is to figure out an optimal policy for how to commute from each location.

This can be represented exactly as a 3 state non-episodic (infinite horizon) Markov decision process, with states S_1, S_2 and S_3, and actions *bus* and *train* as shown in the figure below :



Taking both actions *bus* or *train* from either S_1 or S_2, takes you to state S_3. Taking the *bus* from S_3 takes you to S_1, while taking the *train* from S_3 takes you to S_2.

a) If S_1 represents **San Francisco** and S_3 represents **Stanford**, add a short description of state S_2. [2 pts]

b) All dynamics and rewards in this 3 state MDP are deterministic. The true (but unknown) rewards are specified by the following table

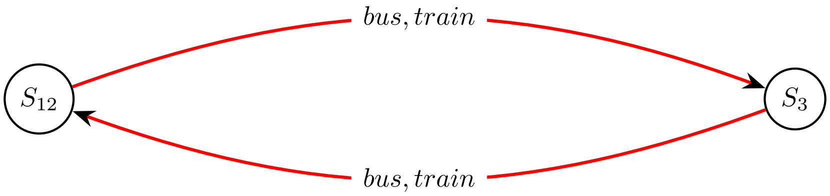
Rewards	bus	train
S_1	+0.7	-1.0
S_2	+1.0	-1.3
S_3	-0.5	-0.7

The optimal non-episodic Q-values with $\gamma = 0.9$ were computed for you and are in the table below:

Q-values	bus	train
S_1	+1.64	-0.06
S_2	+1.94	-0.36
S_3	+0.96	+1.03

What is the optimal policy ? [2 pts]

c) You now notice that a simpler state representation is sufficient to represent the optimal policy as you wrote down in part b. Namely you can simply distinguish between being at Stanford or not being at Stanford. Call state S_{12} the **Not Stanford** state. The actions are still the same, i.e. to take the *bus* or *train*, and both are available from S_{12} and S_3 in this combined representation. Note that in this new representation taking either *bus* or *train* from S_3 takes you to S_{12} and vice-versa.



Can this representation be used to express the optimal policy in part b) ? [2 pts]

d) We now explore whether we could've learned the optimal policy in this simpler state representation. You run $\epsilon-$greedy Q-learning in the true non-episodic environment (representable with 3 states) but using the 2 state representation, with $\epsilon = 0.5$, $\gamma = 0.9$ (same as in part b) for 10^9 (essentially infinite) time steps in the 2-state 2-action tabular setting, as specified in part c, and you get the following Q^* :

Q^*	bus	train
S_{12}	$+2.08$	0.08
S_3	$+1.38$	1.18

What is the optimal policy given these Q-values ? Why does the policy differ from the policy computed using a true 3 state representation ? [4 pts]

e) There are 4 possible deterministic policies for this 2-state 2-action problem. Consider running Monte Carlo policy evaluation in the true non-episodic environment (representable with 3 states) but using the 2 state representation, with $\gamma = 0.9$ (same as above) on each of the 4 policies and then selecting the one with the highest value. Would this find the same policy as c) or d) or a different one ? Why ? [4 pts]

Question 3 – Value Iteration [10 pts]

Non-Monotonic Convergence of Value Iteration

a) Consider a MDP $M = (S, A, P, R, \gamma)$ with finite state and action spaces, and $\gamma = 1$. Let $V^*(s)$ be the optimal value function in state s. As you learned in class, value iteration produces iterates $V_1(s), V_2(s), V_3(s), \ldots$ that eventually converge to $V^*(s)$. However, convergence need not be monotonic for all the states. Find an example of an MDP such that if you run value iteration then the error $|V^*(s) - V_k(s)|$ increases in at least one state s while moving from the k-th to the $(k+1)$-th iteration. Your value function must be initialized to zero and use an MDP with exactly 2 states and exactly 2 actions in each state. The rewards that you specify must be either -2, -1 or 3. Draw the MDP. [8 pts]

b) How do you reconcile the possibility that the error may incease in some state during value iteration with the fact that the Bellman operator is a contraction ? [2 pts]

Question 4 – MC Update vs TD Update [10 pts]

A) Monte Carlo update

a) Write down the formula for a MC update of a Q-value with a lookup table representation. [2 pts]

b) Identify which parts (if any) of the equation involve bootstrapping and/or sampling. [2 pt]

c) What is the computational complexity of performing MC update for an entire episode ? [2 pt]

B) Temporal Difference update

a) Write down the formula for a TD update of a Q-value with a lookup table representation. [2 pts]

b) Identify which parts (if any) of the equation involve bootstrapping and/or sampling. [2 pt]

Question 5 – Value Function Approximation [8 pts]

A) Q-learning with VFA

Consider executing Q-learning using a linear value function approximation. Computing the greedy policy with respect to the current Q does not change the policy (no states would have a different action). Does this imply that the optimal policy has been found ? Choose between

- Yes
- No
- Depends

Provide 1-3 sentences to justify your answer. [4 pts]

B) Functional capacity

You are designing a deep RL system for a new consumer modeling problem. You are choosing between 3 neural network architectures

- **S2.** A fully connected neural network with an input layer, 2 nodes in the hidden layer, and an output layer.
- **S100.** A fully connected neural network with an input layer, 100 nodes in the hidden layer, and an output layer.
- **D10.** A fully connected deep neural network with an input layer, 10 hidden layers each with 10 nodes, and an output layer.

Assume that the input layer consists of 10 nodes, with each node representing a feature for each consumer, and the output layer is a single node. Except the input layer, assume that the activation function for each node in the neural network is a sigmoid. You do not need to consider dropout or batch normalization for this problem.

Which network would you pick and why ? Provide a short explanation. How does the number of **trainable** parameters vary across the 3 networks ? You do not need to compute the exact number of parameters for each network – just do an order of magnitude comparison. [4 pts]

Question 6 – Alice in Wonderland [15 pts]

Alice is taking CS234 and has just learned about the Q-values. She is trying to explore a large **finite-horizon MDP with $\gamma = 1$**. The transitions are **deterministic** and $Q_{H+1}(s,a) = 0$ for all s, a. To help her with her MDP you tell her the optimal policy $\pi^*(s,t)$, defined in every state s and timestep t, that Alice should follow to maximize her reward. Denote with $Q_t^*(s,a)$ the Q-values of the optimal policy upon taking action a in state s at timestep t.

A) First Step Error

In the first timestep $t = 1$ Alice is in state s_1 and chooses action a, which is suboptimal. If she then follows the optimal policy from $t = 2$ until the end of the episode, what is the value of this policy compared to the optimal one? Express your result only using $Q_1^*(s_1, \cdot)$. [3 pts]

B) Relating the Value Function to the Q-Values

Unfortunately Alice does not seem to follow π^* at all. Instead, she keeps following policy π until the end of the episode. Denote with $V_1^*(s_1)$ and $V_1^\pi(s_1)$ the value function of policy π^* and π, respectively, at the start state s_1 in the initial timestep $t = 1$. Show that the loss can be computed using the expression below:

$$V_1^*(s_1) - V_1^\pi(s_1) = \sum_{t=1}^{H} (Q_t^*(s_t, \pi^*(s_t, t)) - Q_t^*(s_t, \pi(s_t, t))) \qquad (1)$$

where $s_1, s_2, s_3, \ldots$ are the deterministic states that Alice encounters upon following policy π.
Hint: Use induction. [6 pts]

C) Loss Due to Inaccurate Q-values

After getting low reward, Alice decides to improve her policy. She will try to infer the best policy using $\tilde{Q}$, which are a close approximation to the Q-values of the optimal value function Q^*. In particular, for every state s, action a and timestep t we have that $|Q_t^*(s,a) - \tilde{Q}_t(s,a)| \leq \epsilon$. Suppose that Alice chooses her policy $\tilde{\pi}(s,t) = \text{argmax}_a \tilde{Q}_t(s,a)$, show that the loss can be bounded with the expression below:

$$V_1^*(s_1) - V_1^{\tilde{\pi}}(s_1) \leq 2\epsilon H$$

Hint: Use equation (1) *from part* ***B***. [6 pts]