CS 234 Midterm – Spring 2016-17

(Do not turn this page until you are instructed to do so!)

Instructions: Please answer the following questions to the best of your ability. You have **80 minutes** to complete the exam. The exam is closed-book, closed-note, closed-internet, etc. However, you may use one letter-sized sheet (front and back) of notes as reference.

All of the intended answers can be written well within the space provided. Good luck!

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: _____

SUNetID: _____

Problem	#1	#2	#3	#4	#5	#6	Total
Maximum	10	12	6	24	16	12	80

1 Markov Decision Process (10 Points)

You are in a Las Vegas casino! You have \$20 for this casino venture and will play until you lose it all or as soon as you double your money (i.e., increase your holding to at least \$40). You can choose to play two slot machines: 1) slot machine A costs \$10 to play and will return \$20 with probability 0.05 and \$0 otherwise; and 2) slot machine B costs \$20 to play and will return \$30 with probability 0.01 and \$0 otherwise. Until you are done, you will choose to play machine A or machine B in each turn. In the space below, provide an MDP that captures the above description. Describe the state space, action space, rewards and transition probabilities. Assume the discount factor $\gamma = 1$.

You can express your solution using equations instead of a table, or a diagram.

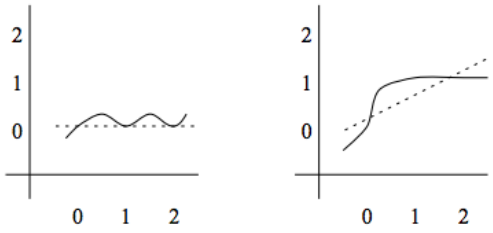
2 Bellman Operator with Function Approximation (12 Points)

Consider an MDP $M = (S, A, R, P, \gamma)$ with finite discrete state space S and action space A. Assume M has dynamics model $P(s'|s, a)$ for all $s, s' \in S$ and $a \in A$ and reward model $R(s, a)$ for all $s \in S$ and $a \in A$.

Recall that the Bellman operator B applied to a function $V : S \to \mathbb{R}$ is defined as

$$B(V)(s) = \max_a (R(s, a) + \gamma \sum_{s'} P(s'|s, a)V(s'))$$

(a) (4 Points) Now, consider a new operator which first applies a Bellman backup and then applies a function approximation, to map the value function back to a space representable by the function approximation. We will consider a linear value function approximator over a continuous state space. Consider the following graphs:



(a) (b)

The graphs show linear regression on the sample $X_0 = \{0, 1, 2\}$ for hypothetical underlying functions. On the left, a target function f (solid line), that evaluates to $f(0) = f(1) = f(2) = 0$, and its corresponding fitted function $\hat{f}(x) = 0$. On the right, another target function g (solid line), that evaluates to $g(0) = 0$ and $g(1) = g(2) = 1$, and its fitted function $\hat{g}(x) = \frac{7}{12}x$.

What happens to the distance between points $\{f(0), f(1), f(2)\}$ and $\{g(0), g(1), g(2)\}$ after we do the linear approximation? In other words, compare $\max_{x \in X_0} |f(x) - g(x)|$ and $\max_{x \in X_0} |\hat{f}(x) - \hat{g}(x)|$.

(b) (4 Points) Is the linear function approximator here a contraction operator? Explain your answer.

(c) (4 Points) Will the new operator be guaranteed to converge to single value function? If yes, will this be the optimal value function for the problem? Justify your answers.

3 Probably Approximately Correct (6 Points)

Let $A(\alpha, \beta)$ be a hypothetical reinforcement learning algorithm, parametrized in terms of α and β such that for any $\alpha > \beta > 1$, it selects action a for state s satisfying $|Q(s,a) - V^*(s)| \leq \frac{\beta}{\alpha}$ in all but $N = \frac{|S||A|\alpha\beta}{1-\gamma}$ steps with probability at least $1 - \frac{1}{\beta^2}$.

Per the definition of *Probably Approximately Correct Reinforcement Learning*, express N as a function of $|S|$, $|A|$, δ, ϵ and γ. What is the resulting N? Is algorithm A probably approximately correct (PAC)? Briefly justify.

4 Model Approximation/Modification (24 Points)

Let $M = (S, A, R, P, \gamma)$ be an MDP with $|S| < \infty$, $|A| < \infty$ and $\gamma \in [0, 1)$. Let $\hat{M}$ be a modification of M to be specified below. We compare the optimal value functions and policies in M and $\hat{M}$.

(a) (12 Points) Let $\hat{M} = (S, A, \hat{R}, P, \gamma)$ where $|\hat{R}(s, a) - R(s, a)| \leq \epsilon$ for all $s \in S$ and $a \in A$. Besides the rewards, all other components of $\hat{M}$ stay the same as in M. Prove that $V^* - \hat{V}^* \leq \frac{\epsilon}{1-\gamma}$. Will M and $\hat{M}$ have the same optimal policy? Briefly explain. Note V^* and $\hat{V}^*$ are optimal value functions in M and $\hat{M}$, respectively.

(b) (12 Points) Now, let $\hat{M} = (S, A, \hat{R}, P, \gamma)$ where $\hat{R}(s,a) - R(s,a) = \epsilon$ for all $s \in S$ and $a \in A$ for some constant ϵ. Equivalently, the same constant ϵ is added to each reward $R(s,a)$. How are V^* and $\hat{V}^*$ related? Express $\hat{V}^*$ in terms of V^*. Will M and $\hat{M}$ has the same optimal policy? Briefly explain.

5 Q-Learning (16 Points)

An agent is exploring an MDP $M = (S, A, R, P, \gamma)$ where $S = \{s_1, s_2, s_3\}$ and $A = \{a_1, a_2, a_3\}$ and $\gamma = 0.5$ and $P(s, a_i, s_i) = 1$ for any s for all i. The rewards for transitioning into a state are defined as $R(s_i) = i$. The maximum reward is 3.

(a) (8 Points) The agent follows the trajectory

$$(s1, a1, 1, s1, a2, 2, s2)$$

Consider a Q-learning agent using ϵ-greedy. A random action never happens to pick the greedy action. Ties are broken by choosing the a_i with the smallest i. The learning rate α is set to 0.5. Initialize Q to zeros.

Could such an agent generate this trajectory for $\epsilon \neq 0$? If yes, label the actions that are greedy and which are random, and justify your answers.

(b) (8 Points) Could the `Rmax` algorithm with $m = 1$ have generated this trajectory? Are any of the (s, a) pairs *known* at the end of the trajectory, and if so, which?

6 Neural Networks (12 Points)

Imagine there is a world where its states are determined by N political parties and represented by a bit string of length N. Each state corresponds to a possible composition of the government with some parties holding seats in the government and others without seats. More specifically, the i-th party holds a seat in a given state if and only if the i-th bit of the corresponding bit string is 1. We say the value of a state is 0 if there are an even number of political parties with a seat in office (because each party is argumentative and can vote for the opposite of each other), or 1 if there are an odd number of political parties with a seat (because ties in voting are not possible). We want to represent the value function by linear function approximation where the input is the binary feature vector s of a state with $s_i = 1$ if i-th party holds a seat, and $s_i = 0$ otherwise.

(a) (3 Points) If $N = 1$, can we represent the value function perfectly using linear value function approximation? Briefly explain.

(b) (3 Points) If $N = 2$, can we represent the value function perfectly using linear value function approximation? Briefly explain.

(c) (3 Points) Consider using a 2 layer (one hidden layer and 1 output layer) neural network. The hidden neurons are defined as $y = f(w^\intercal x + b)$ for an input x with weight w and bias b and activation function $f(x) = 1$ if $x > 0$ and 0 if $x \leq 0$. Can we represent the value function perfectly for $N = 2$? For any N? Briefly explain your answers.

(d) (3 Points) For $N = 2$, is there a reason to use a deeper neural network? How about for general N? Briefly justify your answers.