

- You have approximately 80 minutes.
- Mark your answers ON THE EXERCISE ITSELF. If you are not sure of your answer you may wish to provide a *brief* explanation. All short answer sections can be successfully answered in a few sentences AT MOST.
- For True/False questions, please *circle* your answer.

First name	
Last name	
WUSTL ID	

For staff use only:

Q1.	True/False Questions	/14
Q2.	Bayesian Networks + HMMs	/22
Q3.	Markov Decision Processes	/22
Q4.	Reinforcement Learning	/22
Total		/80

THIS PAGE IS INTENTIONALLY LEFT BLANK

Q1. [14 pts] True/False Questions

Each question is worth 1 point. Leaving a question blank is worth 0 points.

(a) Bayesian Network

- (i) [1 pt] [*true* or *false*] A fair coin is flipped three times. Assume that it must land on either “heads” or “tails”. The probability of seeing three “tails” is less than 0.2.
- (ii) [1 pt] [*true* or *false*] If X , Y , and Z are binary random variables (i.e., their possible values are *true* and *false* only), then the conditional probability table for $\Pr(X \mid Y, Z)$ can be fully specified by $3 = 2^2 - 1$ values.
- (iii) [1 pt] [*true* or *false*] A joint probability distribution can always be factored into a product of conditional distributions.
- (iv) [1 pt] [*true* or *false*] An edge in a Bayesian network indicates a causal relationship between two random variables.
- (v) [1 pt] [*true* or *false*] A Bayesian network is allowed to have a directed cycle.

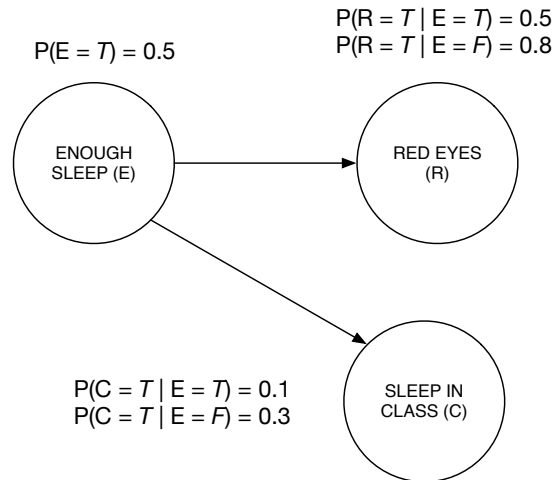
(b) Markov Models

- (i) [1 pt] [*true* or *false*] For an MDP (S, A, T, γ, R) , if we set the discount factor $\gamma = 1$, Value Iteration on this MDP is guaranteed to converge after the first iteration.
- (ii) [1 pt] [*true* or *false*] For an MDP (S, A, T, γ, R) , if we set the discount factor $\gamma = 0$, Value Iteration on this MDP is guaranteed to converge after the first iteration.
- (iii) [1 pt] [*true* or *false*] It is easier to extract optimal policies from optimal Q-values $Q^*(s, a)$ than from optimal state values $V^*(s)$.
- (iv) [1 pt] [*true* or *false*] It is possible to extract an optimal policy from V-values computed via Value Iteration before it has converged.
- (v) [1 pt] [*true* or *false*] For any MDP $(S, A, T, \gamma, R, s_0)$, if we change the start state s_0 , then the optimal policy is guaranteed to change as well.
- (vi) [1 pt] [*true* or *false*] For any MDP $(S, A, T, \gamma, R, s_0)$, if we change the start state s_0 , then the optimal policy is guaranteed to not change.

(c) Reinforcement Learning

- (i) [1 pt] [*true* or *false*] It is possible to extract an optimal policy from Q-values learned via Q-learning before it has converged.
- (ii) [1 pt] [*true* or *false*] One disadvantage of Q-learning is that it can be used only when one does not have prior knowledge of how actions affect the environment of the agent.
- (iii) [1 pt] [*true* or *false*] Q-learning can learn the optimal Q-function Q^* without ever executing the optimal policy.

Q2. [22 pts] Bayesian Networks + HMMs



Consider the Bayesian network above, where E , R , and C are random variables indicating if a student has enough sleep, has red eyes, and sleeps in class, respectively.

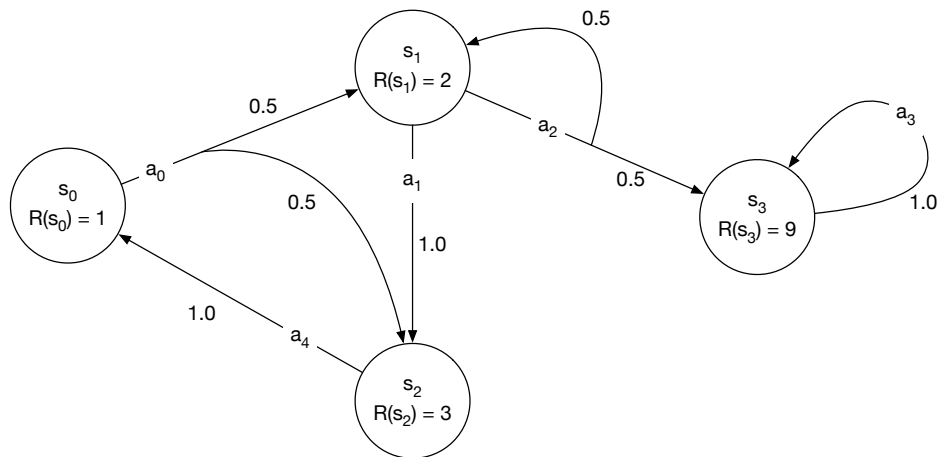
- (a) [4 pts] Calculate the probability that a student has enough sleep, has red eyes, and sleeps in class. Write down the equations before replacing them with numbers.
- (b) [5 pts] Calculate the probability that a student has red eyes and does not sleep in class. Write down the equations before replacing them with numbers.
- (c) [5 pts] Calculate the probability that a student has enough sleep given that he sleeps in class and does not have red eyes.

Imagine that 95% of the time Red Eye Disease (RED) causes red eyes in those with the disease. At any point in time 5% of the population have red eyes and at any point in time 1% of the population has RED. You have red eyes.

(d) [4 pts] Represent all the facts above using the probability notations.

(e) [4 pts] What is the probability that you have RED? Justify your answer mathematically. Write down the equations before replacing them with numbers.

Q3. [22 pts] Markov Decision Processes



Consider the MDP above with four states s_0 , s_1 , s_2 , and s_3 and their corresponding rewards denoted in the nodes. The arrows represent the transition function with the probabilities denoted with each arrow. For example, $T(s_0, a_0, s_1) = 0.5$ and $T(s_0, a_0, s_2) = 0.5$.

- (a) [8 pts] Compute the numerical value of $V_1(s_1)$ and $V_2(s_1)$, i.e., the value of state s_1 after the first two iterations of Value Iteration. Assume that the discount factor $\gamma = 0.9$ and the initial values of all states in the zero-th iteration are all zero, i.e., $V_0(s_0) = V_0(s_1) = V_0(s_2) = V_0(s_3) = 0$.

Recall that the update equation of Value Iteration is:

$$V_{k+1}(s) = \max_a \sum_{s'} T(s, a, s') [R(s) + \gamma V_k(s')].$$

- (b) [6 pts] Value Iteration has converged if the values of all states remain unchanged in two subsequent iterations. What is the value of state s_3 (i.e., $V^*(s_3)$) upon convergence? Describe how you get this value. You are to not use a computer program to compute this value.

Consider an MDP (S, A, T, γ, R) . A *nondeterministic* policy $\pi(s, a)$, rather than mapping states to actions, maps states to a probability distribution over actions. For a state–action pair (s, a) , $\pi(s, a)$ gives the probability of choosing action a in state s .

For example, for a state s , we might have

$$\pi(s, N) = 0.7 \quad \pi(s, E) = 0.1 \quad \pi(s, S) = 0.1 \quad \pi(s, W) = 0.1,$$

which means, “in state s , choose action N 70% of the time; otherwise choose one of E, S, or W with probability 10% each.”

An optimal *deterministic* policy $\pi^*(s)$ is a policy that satisfies the following equation:

$$V^*(s) = \sum_{s'} T(s, \pi^*(s), s') [R(s, \pi^*(s), s') + \gamma V^*(s')].$$

- (c) [3 pts] Extend the equation above to implicitly define an optimal nondeterministic policy. In other words, write the equation that must be satisfied by a nondeterministic policy $\pi^*(s, a)$.

- (d) [5 pts] Is it possible for an optimal nondeterministic policy to be better (i.e., have a larger expected value) than an optimal deterministic policy? Explain.

Q4. [22 pts] Reinforcement Learning

Step number	Current state	Reward received	Action taken	Successor state
1	s_1	-10	a_1	s_1
2	s_1	-10	a_2	s_2
3	s_2	+20	a_1	s_1
4	s_1	-10	a_2	s_2

Consider a system with two states s_1 and s_2 and two actions a_1 and a_2 . You performed the actions listed in the table above and observed the corresponding rewards and transitions. Each step lists the current state, the reward received, the action taken, and the resulting successor state you transitioned to. For example, in Step 1, you start at state s_1 , took action a_1 , transitioned to state s_1 and received reward -10.

- (a) [16 pts] Perform Q-learning using a learning rate $\alpha = 0.5$ and a discount factor $\gamma = 0.5$ for each step. Specifically, compute the following Q-values. You may find the Q-value update equation below helpful:

$$Q(s, a) = Q(s, a) + \alpha \left(r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right)$$

where s is the current state, a is the action taken, s' is the successor state, and r is the reward received. Assume that all Q-values are initialized to zero.

- Compute $Q(s_1, a_1)$ after Step 1.

- Compute $Q(s_1, a_2)$ after Step 2.

- Compute $Q(s_2, a_1)$ after Step 3.

- Compute $Q(s_1, a_2)$ after Step 4.

- (b) [6 pts] What is the optimal policy π^* after these four steps? More specifically, what is the policy for each of the states below?

- $\pi^*(s_1) =$
- $\pi^*(s_2) =$