CS 188 Fall 2018 — Introduction to Artificial Intelligence — Practice Final

- You have approximately 2 hours 50 minutes.
- The exam is closed book, closed calculator, and closed notes except your one-page crib sheet.
- Mark your answers ON THE EXAM 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.

First name	
Last name	
SID	
edX username	
Name of person on your left	
Name of person on your right	

For staff use only:

Q1.	Search and Probability	/11
Q2.	Games	/8
Q3.	Value of Gambling and Bribery	/12
Q4.	Encrypted Knowledge Base	/6
Q5.	The Nature of Discounting	/10
Q6.	Sampling	/12
Q7.	Chameleon	/10
Q8.	Perceptron	/10
	Total	/79

To earn the extra credit, one of the following has to hold true. Please circle and sign.

A I spent 2 hours and 50 minutes or more on the practice exam.

B I spent fewer than 2 hours and 50 minutes on the practice exam, but I believe I have solved all the questions.

Signature: ____

Follow the directions on the website to submit the practice exam and receive the extra credit.

Q1. [11 pts] Search and Probability

(a) Consider a graph search problem where for every action, the cost is at least ϵ, with $\epsilon > 0$. Assume the heuristic is admissible.

(i) [1 pt] [true or *false*] Uniform-cost graph search is guaranteed to return an optimal solution.
True. UCS expands paths in order of least total cost so that the optimal solution is found.

(ii) [1 pt] [true or *false*] The path returned by uniform-cost graph search may change if we add a positive constant C to every step cost. True. Consider that there are two paths from the start state (S) to the goal (G), $S \rightarrow A \rightarrow G$ and $S \rightarrow G$. $cost(S, A) = 1$, $cost(A, G) = 1$, and $cost(S, G) = 3$. So the optimal path is through A. Now, if we add 2 to each of the costs, the optimal path is directly from S to G. Since uniform cost search finds the optimal path, its path will change.

(iii) [1 pt] [*true* or false] A* graph search is guaranteed to return an optimal solution.
False, the heuristic is admissible, but is not guaranteed to be consistent, which is required for optimal graph search.

(iv) [1 pt] [*true* or false] A* graph search is guaranteed to expand no more nodes than depth-first graph search.
False. Depth-first graph search could, for example, go directly to a sub-optimal solution.

(v) [1 pt] [true or *false*] If $h_1(s)$ and $h_2(s)$ are two admissible A* heuristics, then their average $f(s) = \frac{1}{2}h_1(s) + \frac{1}{2}h_2(s)$ must also be admissible. True. Let $h^*(s)$ be the true distance from s. We know that $h_1(s) \leq h^*(s)$ and $h_2(s) \leq h^*(s)$, thus $h_{avg}(s) = \frac{1}{2}h_1(s) + \frac{1}{2}h_2(s) \leq \frac{1}{2}h^*(s) + \frac{1}{2}h^*(s) = h^*(s)$

(vi) [1 pt] [true or *false*] AND/OR search either returns "failure" or a list of actions from start to goal
False. AND/OR search returns a contingency plan, a tree that branches whenever the environment introduces uncertainty, and provides a path of action in all cases.

(b) [3 pts] A, B, C, and D are random variables with binary domains. How many entries are in the following probability tables and what is the sum of the values in each table? Write a "?" in the box if there is not enough information given.

Table	Size	Sum
$P(A\|C)$	4	2
$P(A, D\| +b, +c)$	4	1
$P(B\| +a, C, D)$	8	4

(c) [2 pts] Write all the possible chain rule expansions of the joint probability $P(a, b, c)$. No conditional independence assumptions are made.

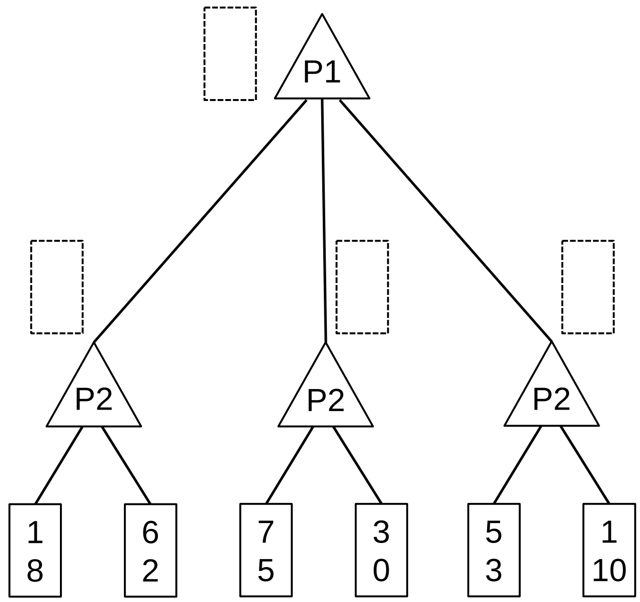
$P(a)P(b|a)P(c|a, b)$, $P(a)P(c|a)P(b|a, c)$,

$P(b)P(a|b)P(c|a, b)$, $P(b)P(c|b)P(a|b, c)$,

$P(c)P(b|c)P(a|b, c)$, $P(c)P(a|c)P(b|a, c)$

Q2. [8 pts] Games

For the following game tree, each player maximizes their respective utility. Let x, y respectively denote the top and bottom values in a node. Player 1 uses the utility function $U_1(x, y) = x$.

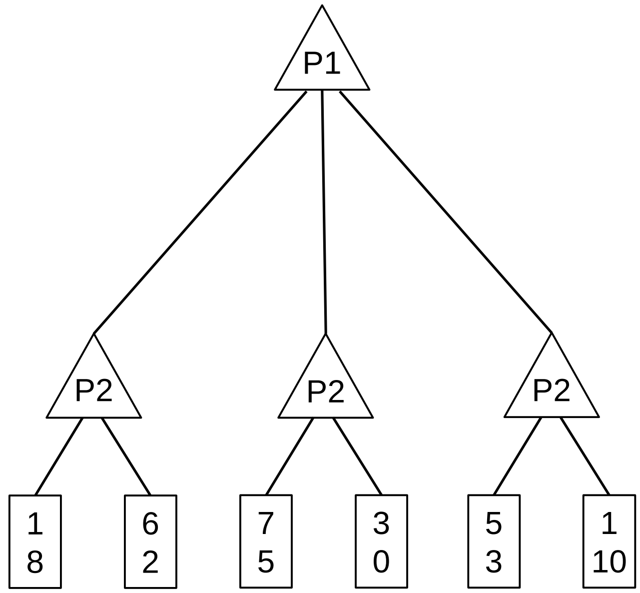


(a) Both players know that Player 2 uses the utility function $U_2(x, y) = x - y$.

(i) [2 pts] Fill in the rectangles in the figure above with pair of values returned by each max node. From top-down, left-right: $(6, 2), (6, 2), (3, 0), (5, 3)$

(ii) [2 pts] You want to save computation time by using pruning in your game tree search. On the game tree above, put an 'X' on branches that do not need to be explored or simply write 'None'. Assume that branches are explored from left to right. None.

Figure repeated for convenience



(b) Now assume Player 2 changes their utility function based on their mood. The probabilities of Player 2's utilities and mood are described in the following table. Let M, U respectively denote the mood and utility function of Player 2.

$P(M = happy)$	$P(M = mad)$
a	b

	$M = happy$	$M = mad$
$P(U_2(x, y) = -x \mid M)$	c	f
$P(U_2(x, y) = x - y \mid M)$	d	g
$P(U_2(x, y) = x^2 + y^2 \mid M)$	e	h

(i) [4 pts] Calculate the maximum expected utility of the game for Player 1 in terms of the values in the game tree and the tables. It may be useful to record and label your intermediate calculations. You may write your answer in terms of a max function.

We first calculate the new probabilities of each utility function as follows.

$P(U_2(x, y) = -x)$	$P(U_2(x, y) = x - y)$	$P(U_2(x, y) = x^2 + y^2)$
$ac + bf$	$ad + bg$	$ae + bh$

$EU(\text{Left Branch}) = (ac + bf)(1) + (ad + bg)(6) + (ae + bh)(6)$
$EU(\text{Middle Branch}) = (ac + bf)(3) + (ad + bg)(3) + (ae + bh)(7)$
$EU(\text{Right Branch}) = (ac + bf)(1) + (ad + bg)(5) + (ae + bh)(1)$

$MEU(\phi) = max((ac+bf)(1)+(ad+bg)(6)+(ae+bh)(6), (ac+bf)(3)+(ad+bg)(3)+(ae+bh)(7), (ac+bf)(1) + (ad + bg)(5) + (ae + bh)(1))$

Q3. [12 pts] Value of Gambling and Bribery

The local casino is offering a new game. There are two biased coins that are indistinguishable in appearance. There is a *head-biased* coin, which yields head with probability 0.8 (and tails with probability 0.2). There is a *tail-biased* coin, which yields tail with probability 0.8 (and head with probability 0.2).

At the start of the game, the dealer gives you one of the two coins at random, with equal probability. You get to flip that coin once. Then you decide if you want to stop or continue. If you choose to continue, you flip it 10 more times. In those 10 flips, each time it yields head, you get $1, and each time it yields tail, you lose $1.

(a) [1 pt] What is the expected value of your earnings if continuing to play with a head-biased coin?

$10 \cdot (0.8 \cdot 1 + 0.2 \cdot -1) = 6$

(b) [1 pt] What is the expected value of your earnings if continuing to play with a tail-biased coin?

$10 \cdot (0.8 \cdot -1 + 0.2 \cdot 1) = -6$

(c) [3 pts] Suppose the first flip comes out head.

(i) [1 pt] What is the posterior probability that the coin is *head-biased*? 4/5

$$\frac{P(hb|heads)}{P(tb|heads)} = \frac{P(heads|hb)P(hb)/P(heads)}{P(heads|tb)P(tb)} = \frac{P(heads|hb)P(hb)}{P(heads|tb)P(tb)} = \frac{(.8)\cdot(1/2)}{(.2)\cdot(1/2)} = 4 \qquad (1)$$

Thus $\frac{P(hb|heads)}{P(hb|heads)+P(hb|tails)} = \frac{4}{1+4} = \frac{4}{5}$.

(ii) [1 pt] What is the expected value of your earnings for continuing to play? $\frac{4}{5} \cdot 6 + \frac{1}{5} \cdot -6 = 3.6$

(iii) [1 pt] Which is the action that maximizes the expected value of your earnings? ● Continue ○ Stop

(d) Suppose the first flip comes out tail.

(i) [1 pt] What is the posterior probability that the coin is *tail-biased*? 4/5

(ii) [1 pt] What is the expected value of your earnings for continuing to play? -3.6

(iii) [1 pt] Which is the action that maximizes the expected value of your earnings? ○ Continue ● Stop

(e) [1 pt] What is the expected value of your earnings after playing the game optimally one time?

$0.5 \cdot 3.6 + .5 \cdot 0 = 1.8$

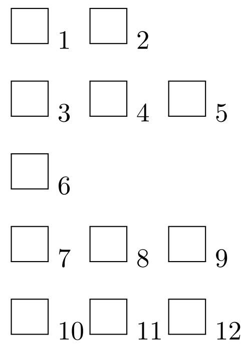
(f) [3 pts] Suppose again that the first flip yields head. The dealer knows which coin you picked. How much are you willing to pay the dealer to find out the type of the coin? Assume that your utility function is the amount of money you make.

6/5 You only change your action (from continue to stop) if the dealer tells you the coin is tails biased. The probability that it's a tails-biased coin is 1/5. The expected returns from a tails-biased coin is -6, so your payout improves by 6 by switching from continue to stop. So the answer is $\frac{1}{5} \cdot 6$.

Q4. [6 pts] Encrypted Knowledge Base

We have a propositional logic knowledge base, but unfortunately, it is encrypted. The only information we have is that:

- Each of the following 12 boxes contains a propositional logic symbol (A, B, C, D, or E) or a propositional logic operator and
- Each line is a valid propositional logic sentence.



(a) [3 pts] We are going to implement a constraint satisfaction problem solver to find a valid assignment to each box from the domain $\{A, B, C, D, E, \wedge, \vee, \neg, \Rightarrow, \Leftrightarrow\}$.

Propositional logic syntax imposes constraints on what can go in each box. What values are in the domain of boxes 1-6 after enforcing the unary syntax constraints?

Box	Remaining Values
1	$\neg$
2	A B C D E
3	A B C D E $\neg$
4	$\wedge$ $\vee$ $\neg$ $\Rightarrow$ $\Leftrightarrow$
5	A B C D E
6	A B C D E

(b) [2 pts] You are given the following assignment as a solution to the knowledge base CSP on the previous page:

$$\neg A$$
$$B \Rightarrow A$$
$$D$$
$$C \vee B$$
$$D \vee E$$

Now that the encryption CSP is solved, we have an entirely new CSP to work on: finding a model. In this new CSP the variables are the symbols $\{A, B, C, D, E\}$ and each variable could be assigned to *true* or *false*.

We are going to run CSP backtracking search with forward checking to find a propositional logic model M that makes all of the sentences in this knowledge base true.

After choosing to assign C to false, what values are removed by running forward checking? On the table of remaining values below, cross off the values that were removed.

Symbol	Remaining Values
A	F
B	T ~~F~~
C	F
D	T
E	T F

Forward checking removes the value false from the domain of B. Forward checking does not continue on to make any other arcs consistent.

(c) [2 pts] We eventually arrive at the model $M = \{A = False, B = False, C = True, D = True, E = True\}$ that causes all of the knowledge base sentences to be true. We have a query sentence α specific as $(A \vee C) \Rightarrow E$. Our model M also causes α to be true. Can we say that the knowledge base entails α? Explain briefly (in one sentence) why or why not.

No, the knowledge base does not entail α. There are other models for which the knowledge base could be true and the query be false. Specifically $\{A = False, B = False, C = True, D = True, E = False\}$ satisfies the knowledge base but causes the query α to be false.

Q5. [10 pts] The Nature of Discounting

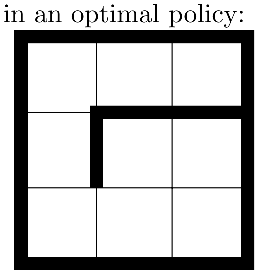
Pacman in stuck in a friendlier maze where he gets a reward every time he visits state (0,0). This setup is a bit different from the one you've seen before: Pacman can get the reward multiple times; these rewards do not get "used up" like food pellets and there are no "living rewards". As usual, Pacman can not move through walls and may take any of the following actions: go North (↑), South (↓), East (→), West (←), or stay in place (∘). State (0,0) gives a total reward of 1 every time Pacman takes an action in that state regardless of the outcome, and all other states give no reward.

The first sentence in the paragraph above was confusing at exam time. The precise reward function is: $R_{(0,0),a} = 1$ for any action a and $R_{s',a} = 0$ for all $s' \neq (0,0)$

You should not need to use any other complicated algorithm/calculations to answer the questions below. We remind you that geometric series converge as follows: $1 + \gamma + \gamma^2 + \cdots = 1/(1-\gamma)$.

(a) [2 pts] Assume finite horizon of $h = 10$ (so Pacman takes exactly 10 steps) and no discounting ($\gamma = 1$).

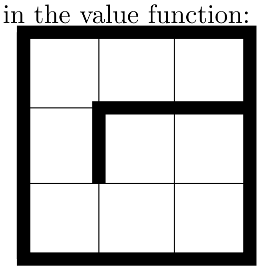
Fill in an optimal policy:



← ← ↑ ↓ ↓ ↑ ← ← ∘

(available actions: ↑, ↓, →, ←, ∘)

Fill in the value function:



(b) The following Q-values correspond to the value function you specified above.

$Q_{s,a}^{10 \text{ steps to go}} = R_s + V_{s'}^{9 \text{ steps to go}}$ where s' is the successor of state s after taking actions a

(i) [1 pt] The Q value of state-action $(0,0)$, (East) is: 9

(ii) [1 pt] The Q value of state-action $(1,1)$, (East) is: 4

(c) Assume finite horizon of $h = 10$, no discounting, but the action to stay in place is temporarily (for this sub-point only) unavailable. Actions that would make Pacman hit a wall are not available. Specifically, Pacman can not use actions North or West to remain in state $(0,0)$ once he is there.

(i) [1 pt] [*true* or *false*] There is just one optimal action at state $(0,0)$

East and South are both optimal actions

(ii) [1 pt] The value of state $(0,0)$ is: 5

Since the "stay action" is no longer available, Pacman needs to exit state $(0,0)$ at even time steps

(d) [2 pts] Assume infinite horizon, discount factor $\gamma = 0.9$.

The value of state $(0,0)$ is: $1/(1-\gamma) = 10$

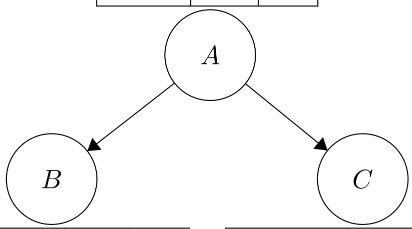
(e) [2 pts] Assume infinite horizon and no discount ($\gamma = 1$). At every time step, after Pacman takes an action and collects his reward, a power outage could suddenly end the game with probability $\alpha = 0.1$.

The value of state $(0,0)$ is: $1/\alpha = 10$

Q6. [12 pts] Sampling

Consider the following Bayes net. The joint distribution is not given, but it may be helpful to fill in the table before answering the following questions.

$P(A)$		
	$+a$	23
	$-a$	13



$P(B\|A)$			
	$+a$	$+b$	12
	$+a$	$-b$	12
	$-a$	$+b$	14
	$-a$	$-b$	34

$P(C\|A)$			
	$+a$	$+c$	12
	$+a$	$-c$	12
	$-a$	$+c$	23
	$-a$	$-c$	13

$P(A,B,C)$			
$+a$	$+b$	$+c$	16
$+a$	$+b$	$-c$	16
$+a$	$-b$	$+c$	16
$+a$	$-b$	$-c$	16
$-a$	$+b$	$+c$	118
$-a$	$+b$	$-c$	136
$-a$	$-b$	$+c$	16
$-a$	$-b$	$-c$	112

We are going to use sampling to approximate the query $P(C|+b)$. Consider the following samples:

Sample 1	Sample 2	Sample 3
$(+a,+b,+c)$	$(+a,-b,-c)$	$(-a,+b,+c)$

(a) [6 pts] Fill in the following table with the probabilities of drawing each respective sample given that we are using each of the following sampling techniques.

$P(+b) = \frac{2}{6} + \frac{1}{12} = \frac{5}{12}$

P(sample | method)	Sample 1	Sample 2
Prior Sampling	16	16
Rejection Sampling	$\frac{16}{512} = 25$	0
Likelihood Weighting	$23 \cdot 12 = 13$	0

Lastly, we want to figure out the probability of getting Sample 3 by Gibbs sampling. We'll initialize the sample to $(+a,+b,+c)$, and resample A then C.

(b) [1 pt] What is the probability the sample equals $(-a,+b,+c)$ after resampling A?

$P(-a|+b,+c) = \frac{P(-a,+b,+c)}{P(-a,+b,+c)+P(+a,+b,+c)} = \frac{118}{118+16} = \frac{118}{418} = \frac{1}{4}$

(c) [1 pt] What is the probability the sample equals $(-a,+b,+c)$ after resampling C, given that the sample equals $(-a,+b,+c)$ after resampling A?

$P(+c|-a,+b) = P(+c|-a) = \frac{2}{3}$

(d) [1 pt] What is the probability of drawing Sample 3, $(-a,+b,+c)$, using Gibbs sampling in this way?

$P(-a|+b,+c) \cdot P(+c|-a,+b) = \frac{1}{4} \cdot \frac{2}{3} = \frac{1}{6}$

(e) [2 pts] Suppose that through some sort of accident, we lost the probability tables associated with this Bayes net. We recognize that the Bayes net has the same form as a naïve Bayes problem. Given our three samples:

$$(+a, +b, +c),\ (+a, -b, -c),\ (-a, +b, +c)$$

Use naïve Bayes maximum likelihood estimation to approximate the parameters in all three probability tables.

$P(A)$ –	$+a$	23
	$-a$	13

	$+a$	$+b$	12
$P(\overline{B}\vert A)$	$+a$	$-b$	12
—	$-a$	$+b$	1
—	$-a$	$-b$	0

	$+a$	$+c$	12
$P(\overline{C}\vert A)$	$+a$	$-c$	12
—	$-a$	$+c$	1
—	$-a$	$-c$	0

(f) [1 pt] What problem would Laplace smoothing fix with the maximum likelihood estimation parameters above?

Laplace smoothing would help prevent overfitting to our very few number of samples. It would avoid the zero probabilities found in the parameters above. It would bring the estimated parameters closer to uniform, which in this case is closer to the original parameters than the maximum likelihood estimated parameters.

Q7. [10 pts] Chameleon

A team of scientists from Berkeley discover a rare species of chameleons. Each one can change its color to be blue or gold, once a day. The probability of colors on a certain day are determined solely by its color on the previous day.

The team spends 5 days observing 10 chameleons changing color from day to day. The recorded counts for the chameleons' color transitions are below.

# of $C_{t+1}\|C_t$	$t=0$	$t=1$	$t=2$	$t=3$
# of $C_{t+1} = gold\|C_t = gold$	0	0	8	2
# of $C_{t+1} = blue\|C_t = gold$	7	0	0	8
# of $C_{t+1} = gold\|C_t = blue$	0	8	2	0
# of $C_{t+1} = blue\|C_t = blue$	3	2	0	0

(a) [3 pts] They suspect that this phenomenon obeys the stationarity assumption – that is, the transition probabilites are actually the same between all the days. Estimate the transition probabilites $P(C_{t+1}|C_t)$ from the above simulation.

	$P(C_{t+1}\|C_t)$
$P(C_{t+1} = gold\|C_t = gold)$	$10/25 = 2/5$
$P(C_{t+1} = blue\|C_t = gold)$	$15/25 = 3/5$
$P(C_{t+1} = gold\|C_t = blue)$	$10/15 = 2/3$
$P(C_{t+1} = blue\|C_t = blue)$	$5/25 = 1/3$

To solve this problem, find the total number of chameleons that were gold ($8 + 2 + 7 + 8 = 25$) and then split it into those that turned gold ($8 + 2 = 10$) and those that turned blue ($7 + 8 = 15$). Normalizing yields $10/25$ and $15/25$ for the first two probabilies. Repeat for the chameleons that were blue.

One common mistake was incorrectly normalizing of the probability table (e.g. dividing by 40 instead of 25). Another was to use only the transitions on t=1 and t=3 to get 0.2, 0.8, 0.8, 0.2, which fails to account for the other observed transitions on t=0 and t=2.

(b) [2 pts] Further scientific tests determine that these chameleons are, in fact, immortal. As a result, they want to determine the distribution of a chameleon's colors over an infinite amount of time.

Given the estimated transition probabilities, what is the steady state distribution for $P(C_\infty)$?

	$P(C_\infty)$
$P(C_\infty = gold)$	$10/19$
$P(C_\infty = blue)$	$9/19$

Let $g = P(C_\infty = gold)$ and $b = P(C_\infty = blue)$.

$$g = \frac{2}{5} + \frac{2}{3}b \implies \frac{3}{5}g = \frac{2}{3}b \implies g = \frac{10}{9}b$$

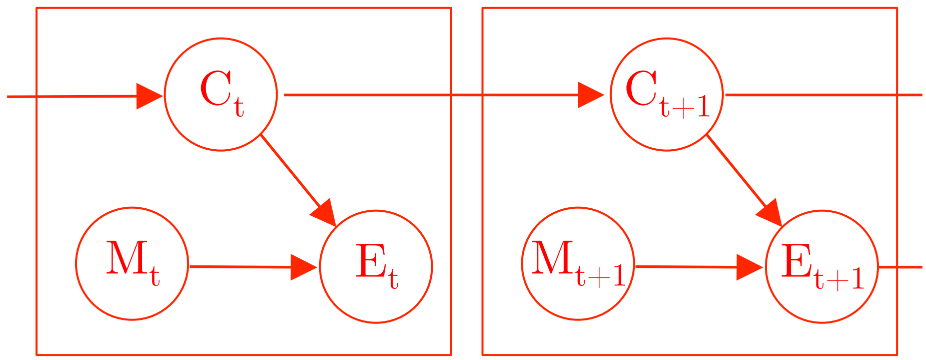
$$g + b = 1$$

Combining the two:

$$\frac{2}{5}b + b = \frac{19}{9}b = 1 \implies b = \frac{9}{19} \implies g = \frac{10}{19}$$

The chameleons, realizing that these tests are being performed, decide to hide. The scientists can no longer observe them directly, but they can observe the bugs that one particular chameleon likes to eat. They know that the chameleon's color influences the probability that it will eat some fraction of a nest. The scientists will observe the size of the nests twice per day: once in the morning, before the chameleon eats, and once in the evening, after the chameleon eats. Every day, the chameleon moves on to a new nest.

(c) [1 pt] Draw a DBN using the variables C_t, C_{t+1}, M_t, M_{t+1}, E_t, and E_{t+1}. C refers to the color of the chameleon, M is the size of a nest in the morning, and E is the size of that nest in the evening.



When the chameleon is blue, it eats half of the bugs in the chosen nest with probability 1/2, one-third of the bugs with probability 1/4, and two-thirds of the bugs with probability 1/4.

When the chameleon is gold, it eats one-third, half, or two-thirds of the bugs, each with probability 1/3.

(d) [4 pts] You would like to use particle filtering to guess the chameleon's color based on the observations of M and E. You observe the following population sizes: $M_1 = 24$, $E_1 = 12$, $M_2 = 36$, and $E_2 = 24$. Fill in the following tables with the weights you would assign to particles in each state at each time step.

State at $t = 1$	Weight
Blue	$\frac{1}{2}$
Gold	$\frac{1}{3}$

State at $t = 2$	Weight
Blue	$\frac{1}{4}$
Gold	$\frac{1}{3}$

The weights in HMM particle filtering are exactly equal to P(emission | parents of emission). In this problem, the change is that the emission is dependent on an addiitonal parameter, the number of morning bugs. For the blue state at t=1, the weight is equal to $P(E_1 = 12|M_1 = 24, C_1 = Blue) = 1/2$.

One extra step that was commonly made was to normalize the weights afterwards to 1 or some other number; this is extraneous as the resample step of particle filtering only depends on the relative (not absolute) weights of the particles.

Q8. [10 pts] Perceptron

We would like to use a perceptron to train a classifier for datasets with 2 features per point and labels 1 or 0.

Let's use a learning rate of $\alpha = .25$. Consider the following labeled training data:

Features (x_1, x_2)	Label y^*
(-1,2)	1
(3,-1)	0
(1,2)	0
(3,1)	1

(a) [2 pts] Our two perceptron weights have been initialized to $w_1 = 2$ and $w_2 = -2$. After processing the first point with the perceptron algorithm, what will be the updated values for these weights?

For the first point, $y = g(w_1x_1 + w_2x_2) = g(2 \cdot -1 + -2 \cdot 2) = g(-5) = 0$, which is incorrectly classified. To update the weights, we add the $\alpha(y^* - h(x))x$: $w_1 = 2 + .25(1-0)(-1) = 1.75$ and $w_2 = -2 + .25(1-0)(2) = -1.5$.

(b) [2 pts] After how many steps will the perceptron algorithm converge? Write "never" if it will never converge.

Note: one steps means processing one point. Points are processed in order and then repeated, until convergence.

The data is not seperable, so it will never converge.

(c) Instead of the standard perceptron algorithm, we decide to treat the perceptron as a single node neural network and update the weights using gradient descent on the loss function.

The loss function for one data point is $Loss(y, y^*) = (y - y^*)^2$, where y^* is the training label for a given point and y is the output of our single node network for that point.

(i) [3 pts] Given a general activation function $g(z)$ and its derivative $g'(z)$, what is the derivative of the loss function with respect to w_1 in terms of g, g', y^*, x_1, x_2, w_1, and w_2?

$\frac{\partial Loss}{\partial w_1} = 2(g(w_1x_1 + w_2x_2) - y^*)g'(w_1x_1 + w_2x_2)x_1$

(ii) [2 pts] For this question, the specific activation function that we will use is:

$$g(z) = 1 \text{ if } z \geq 0 \text{ and } = 0 \text{ if } z < 0$$

Given the following gradient descent equation to update the weights given a single data point. With initial weights of $w_1 = 2$ and $w_2 = -2$, what are the updated weights after processing the first point?

Gradient descent update equation: $w_i = w_i - \alpha \frac{\partial Loss}{\partial w_1}$

Because the gradient of g is zero, the weights will stay $w_1 = 2$ and $w_2 = -2$.

(iii) [1 pt] What is the most critical problem with this gradient descent training process with that activation function?

The gradient of that activation function is zero, so the weights will not update.

THIS PAGE IS INTENTIONALLY LEFT BLANK