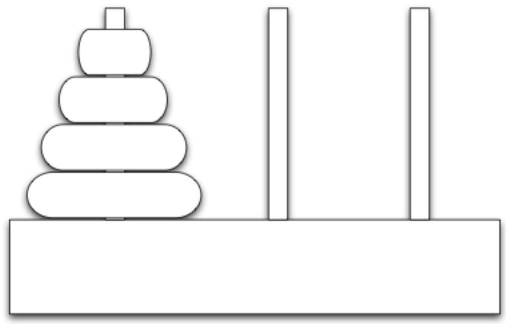
CS 188 Fall 2018 Section 1: Search

1 Towers of Hanoi



The Towers of Hanoi is a famous problem for studying recursion in computer science and recurrence equations in discrete mathematics. We start with N discs of varying sizes on a peg (stacked in order according to size), and two empty pegs. We are allowed to move a disc from one peg to another, but we are never allowed to move a larger disc on top of a smaller disc. The goal is to move all the discs to the rightmost peg (see figure).

In this problem, we will formulate the Towers of Hanoi as a search problem.

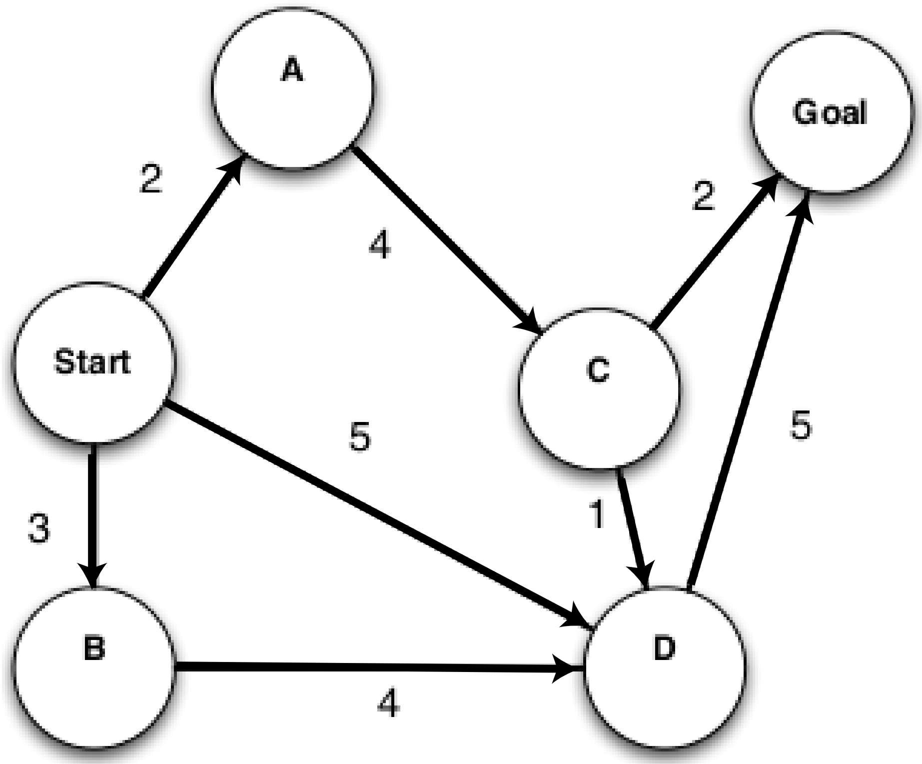
(a) Propose a state representation for the problem

(b) What is the start state?

(c) From a given state, what actions are legal?

(d) What is the goal test?

2 Search algorithms in action



For each of the following search strategies, work out the path returned by the search on the graph shown above. In all cases, assume ties resolve in such a way that states with earlier alphabetical order are expanded first. The start and goal state are S and G, respectively.

a) Depth-first search.

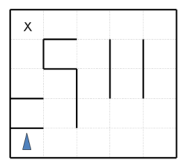
b) Breadth-first search.

c) Uniform cost search.

CS188 Fall 2018 Section 2: Graph Search + CSPs

1 Search and Heuristics

Imagine a car-like agent wishes to exit a maze like the one shown below:



The agent is directional and at all times faces some direction $d \in (N, S, E, W)$. With a single action, the agent can *either* move forward at an adjustable velocity v *or* turn. The turning actions are *left* and *right*, which change the agent's direction by 90 degrees. Turning is only permitted when the velocity is zero (and leaves it at zero). The moving actions are *fast* and *slow*. *Fast* increments the velocity by 1 and *slow* decrements the velocity by 1; in both cases the agent then moves a number of squares equal to its NEW adjusted velocity. Any action that would result in a collision with a wall crashes the agent and is illegal. Any action that would reduce v below 0 or above a maximum speed $V_{\max}$ is also illegal. The agent's goal is to find a plan which parks it (stationary) on the exit square using as few actions (time steps) as possible.

As an example: if the agent shown were initially stationary, it might first turn to the east using (*right*), then move one square east using *fast*, then two more squares east using *fast* again. The agent will of course have to *slow* to turn.

1. If the grid is M by N, what is the size of the state space? Justify your answer. You should assume that all configurations are reachable from the start state.

2. Is the Manhattan distance from the agent's location to the exit's location admissible? Why or why not?

3. State and justify a non-trivial admissible heuristic for this problem which is not the Manhattan distance to the exit.

4. If we used an inadmissible heuristic in A* graph search, would the search be complete? Would it be optimal?

5. If we used an *admissible* heuristic in A* graph search, is it guaranteed to return an optimal solution? What if the heuristic was consistent?

6. Give a general advantage that an inadmissible heuristic might have over an admissible one.

2 Course Scheduling

You are in charge of scheduling for computer science classes that meet Mondays, Wednesdays and Fridays. There are 5 classes that meet on these days and 3 professors who will be teaching these classes. You are constrained by the fact that each professor can only teach one class at a time.

The classes are:

1. Class 1 - Intro to Programming: meets from 8:00-9:00am
2. Class 2 - Intro to Artificial Intelligence: meets from 8:30-9:30am
3. Class 3 - Natural Language Processing: meets from 9:00-10:00am
4. Class 4 - Computer Vision: meets from 9:00-10:00am
5. Class 5 - Machine Learning: meets from 10:30-11:30am

The professors are:

1. Professor A, who is qualified to teach Classes 1, 2, and 5.
2. Professor B, who is qualified to teach Classes 3, 4, and 5.
3. Professor C, who is qualified to teach Classes 1, 3, and 4.

1. Formulate this problem as a CSP problem in which there is one variable per class, stating the domains (after enforcing unary constraints), and binary constraints. Constraints should be specified formally and precisely, but may be implicit rather than explicit.

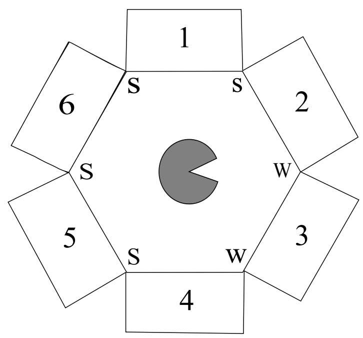
2. Draw the constraint graph associated with your CSP.

3 (Optional) CSPs: Trapped Pacman

Pacman is trapped! He is surrounded by mysterious corridors, each of which leads to either a pit (P), a ghost (G), or an exit (E). In order to escape, he needs to figure out which corridors, if any, lead to an exit and freedom, rather than the certain doom of a pit or a ghost.

The one sign of what lies behind the corridors is the wind: a pit produces a strong breeze (S) and an exit produces a weak breeze (W), while a ghost doesn't produce any breeze at all. Unfortuantely, Pacman cannot measure the the strength of the breeze at a specific corridor. Instead, he can stand *between* two adjacent corridors and feel the max of the two breezes. For example, if he stands between a pit and an exit he will sense a strong (S) breeze, while if he stands between an exit and a ghost, he will sense a weak (W) breeze. The measurements for all intersections are shown in the figure below.

Also, while the total number of exits might be zero, one, or more, Pacman knows that two neighboring squares will *not* both be exits.



Pacman models this problem using variables X_i for each corridor i and domains P, G, and E.

1. State the binary and/or unary constraints for this CSP (either implicitly or explicitly).

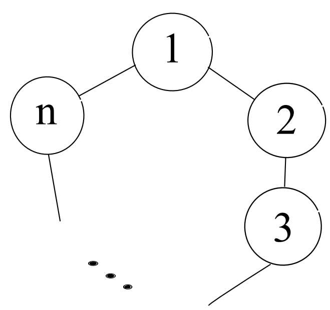
2. Cross out the values from the domains of the variables that will be deleted in enforcing arc consistency.

X_1	P	G	E
X_2	P	G	E
X_3	P	G	E
X_4	P	G	E
X_5	P	G	E
X_6	P	G	E

3. According to MRV, which variable or variables could the solver assign first?

4. Assume that Pacman knows that $X_6 = G$. List all the solutions of this CSP or write *none* if no solutions exist.

The CSP described above has a circular structure with 6 variables. Now consider a CSP forming a circular structure that has n variables ($n > 2$), as shown below. Also assume that the domain of each variable has cardinality d.



5. Explain precisely how to solve this general class of circle-structured CSPs efficiently (i.e. in time linear in the number of variables), using methods covered in class. Your answer should be at most two sentences.

6. 2 If standard backtracking search were run on a circle-structured graph, enforcing arc consistency at every step, what, if anything, can be said about the worst-case backtracking behavior (e.g. number of times the search could backtrack)?

CS188 Fall 2018 Section 3: CSPs + Games

1 CSP: Air Traffic Control

We have five planes: A, B, C, D, and E and two runways: international and domestic. We would like to schedule a time slot and runway for each aircraft to **either** land or take off. We have four time slots: $\{1, 2, 3, 4\}$ for each runway, during which we can schedule a landing or take off of a plane. We must find an assignment that meets the following constraints:

- Plane B has lost an engine and must land in time slot 1.
- Plane D can only arrive at the airport to land during or after time slot 3.
- Plane A is running low on fuel but can last until at most time slot 2.
- Plane D must land before plane C takes off, because some passengers must transfer from D to C.
- No two aircrafts can reserve the same time slot for the same runway.

a) Complete the formulation of this problem as a CSP in terms of variables, domains, and constraints (both unary and binary). Constraints should be expressed implicitly using mathematical or logical notation rather than with words.

b) For the following subparts, we add the following two constraints:

- Planes A, B, and C cater to international flights and can only use the international runway.
- Planes D and E cater to domestic flights and can only use the domestic runway.

i With the addition of the two constraints above, we completely reformulate the CSP and draw the constraint graph.

ii What are the domains of the variables after enforcing arc-consistency? Begin by enforcing unary constraints. (Cross out values that are no longer in the domain.)

A	1	2	3	4
B	1	2	3	4
C	1	2	3	4
D	1	2	3	4
E	1	2	3	4

iii Arc-consistency can be rather expensive to enforce, and we believe that we can obtain faster solutions using only **forward-checking** on our variable assignments. Using the Minimum Remaining Values heuristic, perform backtracking search on the graph, breaking ties by picking lower values and characters first. List the $(variable, assignment)$ pairs in the order they occur (including the assignments that are reverted upon reaching a dead end). Enforce unary constraints before starting the search.

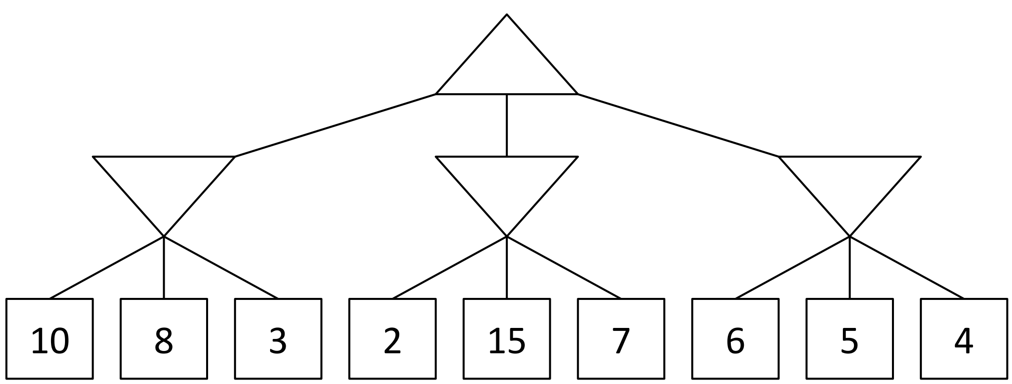
A	1	2	3	4
B	1	2	3	4
C	1	2	3	4
D	1	2	3	4
E	1	2	3	4

Answer:

c) Suppose we have just one runway and n planes, where no two planes can use the runway at once. We are assured that the constraint graph will always be tree-structured and that a solution exists. What is the runtime complexity in terms of the number of planes, n, of a CSP solver that runs arc-consistency and then assigns variables in a topological ordering?

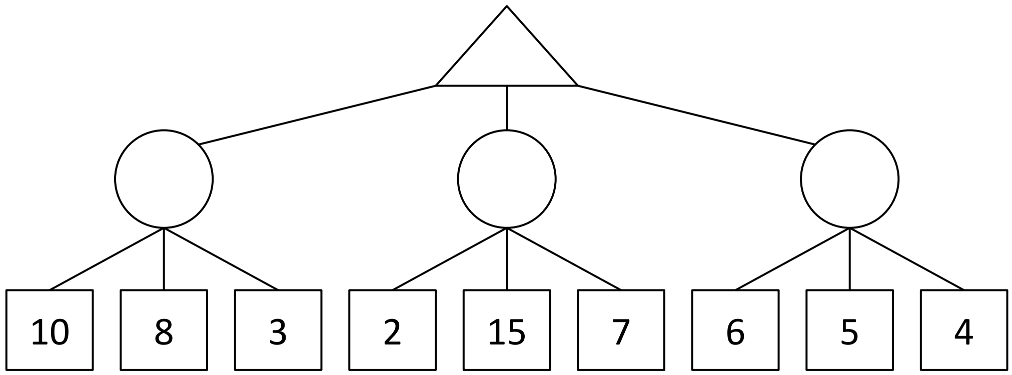
2 Games

1. Consider the zero-sum game tree shown below. Triangles that point up, such as at the top node (root), represent choices for the maximizing player; triangles that point down represent choices for the minimizing player. Assuming both players act optimally, fill in the minimax value of each node.



2. Which nodes can be pruned from the game tree above through alpha-beta pruning? If no nodes can be pruned, explain why not. Assume the search goes from left to right; when choosing which child to visit first, choose the left-most unvisited child.

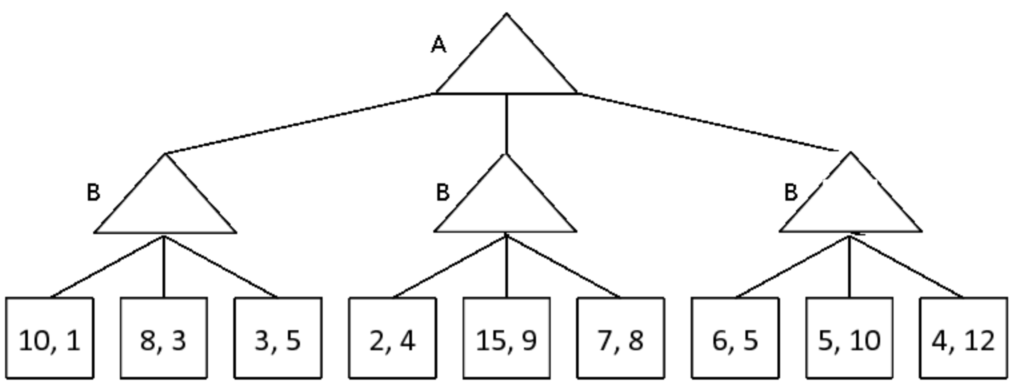
3. (optional) Again, consider the same zero-sum game tree, except that now, instead of a minimizing player, we have a chance node that will select one of the three values uniformly at random. Fill in the expectimax value of each node. The game tree is redrawn below for your convenience.



4. (optional) Which nodes can be pruned from the game tree above through alpha-beta pruning? If no nodes can be pruned, explain why not.

3 (Optional) Nonzero-sum Games

1. Let's look at a non-zero-sum version of a game. In this formulation, player A's utility will be represented as the first of the two leaf numbers, and player B's utility will be represented as the second of the two leaf numbers. Fill in this non-zero game tree assuming each player is acting optimally.



2. Which nodes can be pruned from the game tree above through alpha-beta pruning? If no nodes can be pruned, explain why not.

CS188 Fall 2018 Section 4: Games and MDPs

1 Utilities

1. Consider a utility function of $U(x) = 2x$. What is the utility for each of the following outcomes?
   (a) 3

   (b) L($\frac{2}{3}$, 3; $\frac{1}{3}$, 6)

   (c) -2

   (d) L(0.5, 2; 0.5, L(0.5, 4; 0.5, 6))

2. Consider a utility function of $U(x) = x^2$. What is the utility for each of the following outcomes?
   (a) 3

   (b) L($\frac{2}{3}$, 3; $\frac{1}{3}$, 6)

   (c) -2

   (d) L(0.5, 2; 0.5, L(0.5, 4; 0.5, 6))

3. What is the expected monetary value (EMV) of the lottery $L(\frac{2}{3}, \$3; \frac{1}{3}, \$6)$?

4. For each of the following types of utility function, state how the utility of the lottery $U(L)$ compares to the utility of the amount of money equal to the EMV of the lottery, $U(EMV(L))$. Write $<, >, =$, or ? for can't tell.
   (a) U is an arbitrary function.
   $U(L)$ ___ $U(EMV(L))$

   (b) U is monotonically increasing and its rate of increase is increasing (its second derivative is positive).
   $U(L)$ ___ $U(EMV(L))$

   (c) U is monotonically increasing and linear (its second derivative is zero).
   $U(L)$ ___ $U(EMV(L))$

   (d) U is monotonically increasing and its rate of increase is decreasing (its second derivative is negative).
   $U(L)$ ___ $U(EMV(L))$

2 MDPs: Micro-Blackjack

In micro-blackjack, you repeatedly draw a card (with replacement) that is equally likely to be a 2, 3, or 4. You can either Draw or Stop if the total score of the cards you have drawn is less than 6. If your total score is 6 or higher, the game ends, and you receive a utility of 0. When you Stop, your utility is equal to your total score (up to 5), and the game ends. When you Draw, you receive no utility. There is no discount ($\gamma = 1$). Let's formulate this problem as an MDP with the following states: $0, 2, 3, 4, 5$ and a *Done* state, for when the game ends.

1. What is the transition function and the reward function for this MDP?

2. Fill in the following table of value iteration values for the first 4 iterations.

States	0	2	3	4	5
V_0					
V_1					
V_2					
V_3					
V_4					

3. You should have noticed that value iteration converged above. What is the optimal policy for the MDP?

States	0	2	3	4	5
π^*					

3 (Optional) Minimax and Expectimax

In this problem, you will investigate the relationship between expectimax trees and minimax trees for zero-sum two player games. Imagine you have a game which alternates between player 1 (max) and player 2. The game begins in state s_0, with player 1 to move. Player 1 can either choose a move using minimax search, or expectimax search, where player 2's nodes are chance rather than min nodes.

1. Draw a (small) game tree in which the root node has a larger value if expectimax search is used than if minimax is used, or argue why it is not possible.

2. Draw a (small) game tree in which the root node has a larger value if minimax search is used than if expectimax is used, or argue why it is not possible.

3. Under what assumptions about player 2 should player 1 use minimax search rather than expectimax search to select a move?

4. Under what assumptions about player 2 should player 1 use expectimax search rather than minimax search?

5. Imagine that player 1 wishes to act optimally (rationally), and player 1 knows that player 2 also intends to act optimally. However, player 1 also knows that player 2 (mistakenly) believes that player 1 is moving uniformly at random rather than optimally. Explain how player 1 should use this knowledge to select a move. Your answer should be a precise algorithm involving a game tree search, and should include a sketch of an appropriate game tree with player 1's move at the root. Be clear what type of nodes are at each ply and whose turn each ply represents.

CS188 Fall 2018 Section 5: MDP + RL

1 MDPs: Micro-Blackjack

In micro-blackjack, you repeatedly draw a card (with replacement) that is equally likely to be a 2, 3, or 4. You can either Draw or Stop if the total score of the cards you have drawn is less than 6. If your total score is 6 or higher, the game ends, and you receive a utility of 0. When you Stop, your utility is equal to your total score (up to 5), and the game ends. When you Draw, you receive no utility. There is no discount ($\gamma = 1$). Let's formulate this problem as an MDP with the following states: $0, 2, 3, 4, 5$ and a *Done* state, for when the game ends.

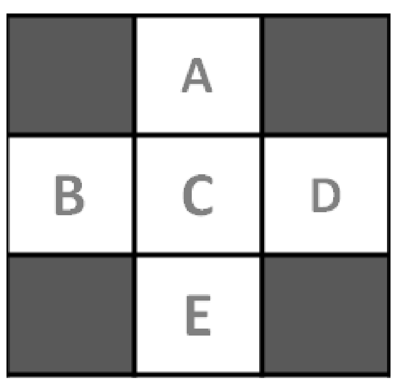
1. What is the transition function and the reward function for this MDP?

2. Perform one iteration of policy iteration for one step of this MDP, starting from the fixed policy below:

States	0	2	3	4	5
π_i	Draw	Stop	Draw	Stop	Draw
V^{π_i}					
π_{i+1}					

2 Learning in Gridworld

Consider the example gridworld that we looked at in lecture. We would like to use TD learning and q-learning to find the values of these states.



Suppose that we have the following observed transitions:
(B, East, C, 2), (C, South, E, 4), (C, East, A, 6), (B, East, C, 2)

The initial value of each state is 0. Assume that $\gamma = 1$ and $\alpha = 0.5$.

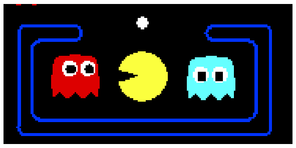
1. What are the learned values from TD learning after all four observations?

2. What are the learned Q-values from Q-learning after all four observations?

3 Pacman with Feature-Based Q-Learning

We would like to use a Q-learning agent for Pacman, but the state size for a large grid is too massive to hold in memory. To solve this, we will switch to feature-based representation of Pacman's state.

1. Say our two minimal features are the number of ghosts within 1 step of Pacman (F_g) and the number of food pellets within 1 step of Pacman (F_p). You'll notice that these features depend only on the state, not the actions you take. Keep that in mind as you answer the next couple of questions. For this pacman board:

   Extract the two features (calculate their values).

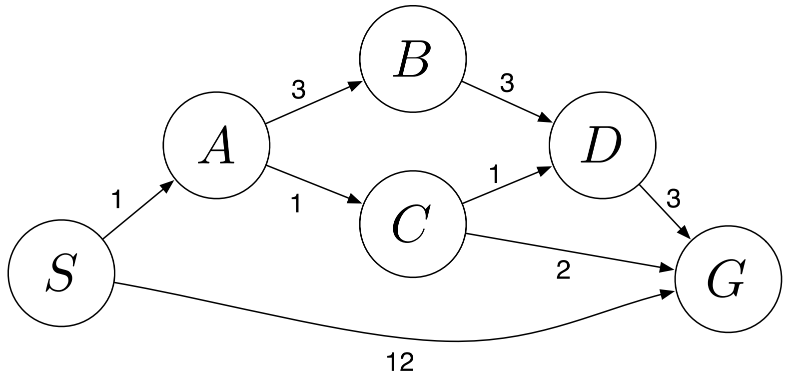
2. With Q Learning, we train off of a few episodes, so our weights begin to take on values. Right now $w_g = 100$ and $w_p = -10$. Calculate the Q value for the state above.

3. We receive an episode, so now we need to update our values. An episode consists of a start state s, an action a, an end state s', and a reward r. The start state of the episode is the state above (where you already calculated the feature values and the expected Q value). The next state has feature values $F_g = 0$ and $F_p = 2$ and the reward is 50. Assuming a discount of $\gamma = 0.5$, calculate the new estimate of the Q value for s based on this episode.

4. With this new estimate and a learning rate (α) of 0.5, update the weights for each feature.

CS188 Fall 2018 Section 6: Midterm 1 Prep

1 . Search



Answer the following questions about the search problem shown above. Assume that ties are broken alphabetically. (For example, a partial plan $S \to X \to A$ would be expanded before $S \to X \to B$; similarly, $S \to A \to Z$ would be expanded before $S \to B \to A$.) For the questions that ask for a path, please give your answers in the form '$S - A - D - G$.'

(a) What path would breadth-first graph search return for this search problem?

(b) What path would uniform cost graph search return for this search problem?

(c) What path would depth-first graph search return for this search problem?

(d) What path would A* graph search, using a consistent heuristic, return for this search problem?

(e) Consider the heuristics for this problem shown in the table below.

State	h_1	h_2
S	5	4
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

(i) Is h_1 admissible? **Yes** **No**

(ii) Is h_1 consistent? **Yes** **No**

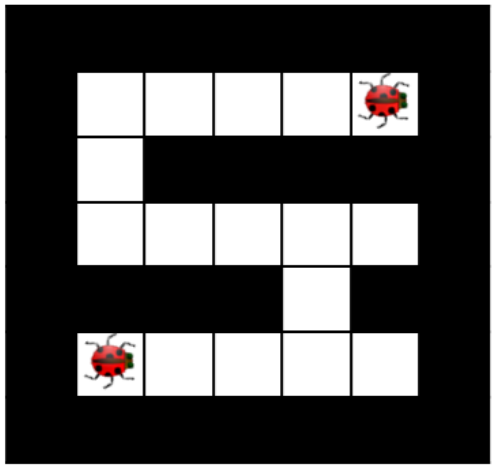
(iii) Is h_2 admissible? **Yes** **No**

(iv) Is h_2 consistent? **Yes** **No**

2 . Hive Minds: Redux

Let's revisit our bug friends. To recap, you control one or more insects in a rectangular maze-like environment with dimensions $M \times N$, as shown in the figures below. At each time step, an insect can move North, East, South, or West (but not diagonally) into an adjacent square if that square is currently free, or the insect may stay in its current location. Squares may be blocked by walls (as denoted by the black squares), but the map is known.

For the following questions, you should answer for a general instance of the problem, not simply for the example maps shown.



You now control a pair of long lost bug friends. You know the maze, but you do not have any information about which square each bug starts in. You want to help the bugs reunite. You must pose a search problem whose solution is an all-purpose sequence of actions such that, after executing those actions, both bugs will be on the same square, regardless of their initial positions. Any square will do, as the bugs have no goal in mind other than to see each other once again. Both bugs execute the actions mindlessly and do not know whether their moves succeed; if they use an action which would move them in a blocked direction, they will stay where they are. Unlike the flea in the previous question, bugs *cannot* jump onto walls. Both bugs can move in each time step. Every time step that passes has a cost of one.

(a) Give a *minimal* state representation for the above search problem.

(b) Give the size of the state space for this search problem.

(c) Give a nontrivial admissible heuristic for this search problem.

3 . CSPs: Time Management

Two of our TAs, Arjun and Dave, are making their schedules for a busy morning. There are five tasks to be carried out:

(F) Pick up food for the group's research seminar, which, sadly, takes one precious hour.
(H) Prepare homework questions, which takes 2 consecutive hours.
(P) Prepare the PR2 (robot that Pieter uses for research) for a group of preschoolers' visit, which takes one hour.
(S) Lead the research seminar, which takes one hour.
(T) Teach the preschoolers about the PR2 robot, which takes 2 consecutive hours.

The schedule consists of one-hour slots: 8am-9am, 9am-10am, 10am-11am, 11am-12pm. The requirements for the schedule are as follows:

1. In any given time slot each TA can do at most one task (F, H, P, S, T).
2. The PR2 preparation (P) should happen before teaching the preschoolers (T).
3. The food should be picked up (F) before the seminar (S).
4. The seminar (S) should be finished by 10am.
5. Arjun is going to deal with food pick up (F) since he has a car.
6. The TA not leading the seminar (S) should still attend, and hence cannot perform another task (F, T, P, H) during the seminar.
7. The seminar (S) leader does not teach the preschoolers (T).
8. The TA who teaches the preschoolers (T) must also prepare the PR2 robot (P).
9. Preparing homework questions (H) takes 2 consecutive hours, and hence should start at or before 10am.
10. Teaching the preschoolers (T) takes 2 consecutive hours, and hence should start at or before 10am.

To formalize this problem as a CSP, use the variables F, H, P, S and T. The values they take on indicate the TA responsible for it, and the starting time slot during which the task is carried out (for a task that spans 2 hours, the variable represents the starting time, but keep in mind that the TA will be occupied for the next hour also - make sure you enforce constraint (a)!). Hence there are eight possible values for each variable, which we will denote by A8, A9, A10, A11, D8, D9, D10, D11, where the letter corresponds to the TA and the number corresponds to the time slot. For example, assigning the value of A8 to a variables means that this task is carried about by Arjun from 8am to 9am.

(a) What is the size of the state space for this CSP?

(b) Which of the statements above include unary constraints?

(c) In the table below, enforce all unary constraints by crossing out values in the table on the left below. If you made a mistake, cross out the whole table and use the right one.

F	A8	A9	A10	A11	D8	D9	D10	D11
H	A8	A9	A10	A11	D8	D9	D10	D11
P	A8	A9	A10	A11	D8	D9	D10	D11
S	A8	A9	A10	A11	D8	D9	D10	D11
T	A8	A9	A10	A11	D8	D9	D10	D11

F	A8	A9	A10	A11	D8	D9	D10	D11
H	A8	A9	A10	A11	D8	D9	D10	D11
P	A8	A9	A10	A11	D8	D9	D10	D11
S	A8	A9	A10	A11	D8	D9	D10	D11
T	A8	A9	A10	A11	D8	D9	D10	D11

(d) Start from the table above, select the variable S and assign the value A9 to it. Perform forward checking by crossing out values in the table below. Again the table on the right is for you to use in case you believe you made a mistake.

F	A8	A9	A10	A11	D8	D9	D10	D11
H	A8	A9	A10	A11	D8	D9	D10	D11
P	A8	A9	A10	A11	D8	D9	D10	D11
S	A8	A9	A10	A11	D8	D9	D10	D11
T	A8	A9	A10	A11	D8	D9	D10	D11

F	A8	A9	A10	A11	D8	D9	D10	D11
H	A8	A9	A10	A11	D8	D9	D10	D11
P	A8	A9	A10	A11	D8	D9	D10	D11
S	A8	A9	A10	A11	D8	D9	D10	D11
T	A8	A9	A10	A11	D8	D9	D10	D11

(e) Based on the result of (d), what variable will we choose to assign next based on the MRV heuristic (breaking ties alphabetically)? Assign the first possible value to this variable, and perform forward checking by crossing out values in the table below. Again the table on the right is for you to use in case you believe you made a mistake.

Variable _____ is selected and gets assigned value _____.

F	A8	A9	A10	A11	D8	D9	D10	D11
H	A8	A9	A10	A11	D8	D9	D10	D11
P	A8	A9	A10	A11	D8	D9	D10	D11
S	A8	A9	A10	A11	D8	D9	D10	D11
T	A8	A9	A10	A11	D8	D9	D10	D11

F	A8	A9	A10	A11	D8	D9	D10	D11
H	A8	A9	A10	A11	D8	D9	D10	D11
P	A8	A9	A10	A11	D8	D9	D10	D11
S	A8	A9	A10	A11	D8	D9	D10	D11
T	A8	A9	A10	A11	D8	D9	D10	D11

Have we arrived at a dead end (i.e., has any of the domains become empty)?

(f) We return to the result from enforcing just the unary constraints, which we did in (c). Select the variable S and assign the value A9. Enforce arc consistency by crossing out values in the table below.

F	A8	A9	A10	A11	D8	D9	D10	D11
H	A8	A9	A10	A11	D8	D9	D10	D11
P	A8	A9	A10	A11	D8	D9	D10	D11
S	A8	A9	A10	A11	D8	D9	D10	D11
T	A8	A9	A10	A11	D8	D9	D10	D11

F	A8	A9	A10	A11	D8	D9	D10	D11
H	A8	A9	A10	A11	D8	D9	D10	D11
P	A8	A9	A10	A11	D8	D9	D10	D11
S	A8	A9	A10	A11	D8	D9	D10	D11
T	A8	A9	A10	A11	D8	D9	D10	D11

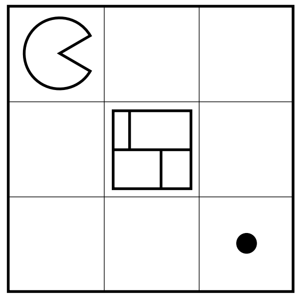
(g) Compare your answers to (d) and to (f). Does arc consistency remove more values or less values than forward checking does? Explain why.

(h) Check your answer to (f). Without backtracking, does any solution exist along this path? Provide the solution(s) or state that there is none.

4 . Surrealist Pacman

In the game of Surrealist Pacman, Pacman  plays against a moving wall . On Pacman's turn, Pacman must move in one of the four cardinal directions, and must move into an unoccupied square. On the wall's turn, the wall must move in one of the four cardinal directions, and must move into an unoccupied square. The wall cannot move into a dot-containing square. Staying still is not allowed by either player. Pacman's score is always equal to the number of dots he has eaten.

The first game begins in the configuration shown below. Pacman moves first.



(a) Draw a game tree with one move for each player. Nodes in the tree represent game states (location of all agents and walls). Edges in the tree connect successor states to their parent states. Draw only the legal moves.

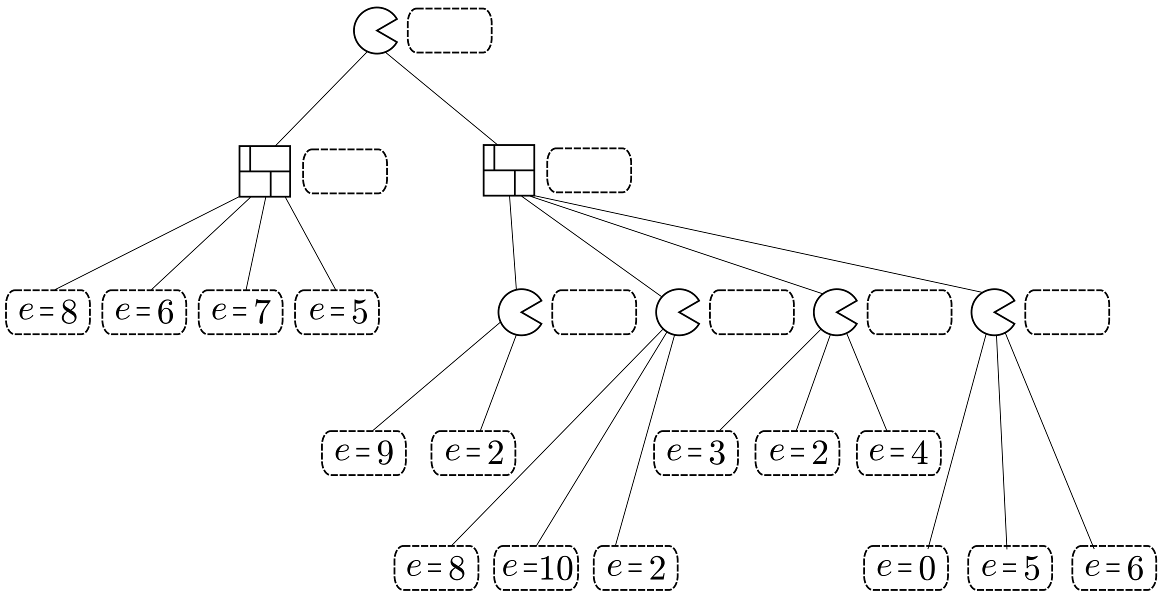
(b) According to the depth-limited game tree you drew above what is the value of the game? Use Pacman's score as your evaluation function.

(c) If we were to consider a game tree with ten moves for each player (rather than just one), what would be the value of the game as computed by minimax?

A second game is played on a more complicated board. A partial game tree is drawn, and leaf nodes have been scored using an (unknown) evaluation function e.

(d) In the dashed boxes, fill in the values of all internal nodes using the minimax algorithm.

(e) Cross off any nodes that are not evaluated when using alpha-beta pruning (assuming the standard left-to-right traversal of the tree).



Suppose that this evaluation function has a special property: it is known to give the correct minimax value of any internal node to within 2, and the correct minimax values of the leaf nodes exactly. That is, if v is the true minimax value of a particular node, and e is the value of the evaluation function applied to that node, $e - 2 \le v \le e + 2$, and $v = e$ if the node is a dashed box in the tree below.

Using this special property, you can modify the alpha-beta pruning algorithm to prune more nodes.

(f) Standard alpha-beta pseudocode is given below (only the max-value recursion). Fill in the boxes on the right to replace the corresponding boxes on the left so that the pseudocode prunes as many nodes as possible, taking account of this special property of the evaluation function.

function MAX-VALUE(*node*, α, β)
    $e \leftarrow$ EVALUATIONFUNCTION(*node*)
    if *node* is leaf **then**
        return e
    end if
    $\boxed{}$ (1)
    $v \leftarrow -\infty$
    for *child* $\leftarrow$ CHILDREN(*node*) **do**
        $\boxed{v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(child, \alpha, \beta))}$ (2)
        if $v \ge \beta$ **then**
            return v
        end if
        $\alpha \leftarrow$ MAX(α, v)
    end for
    return v
end function

Fill in these boxes:

(1)

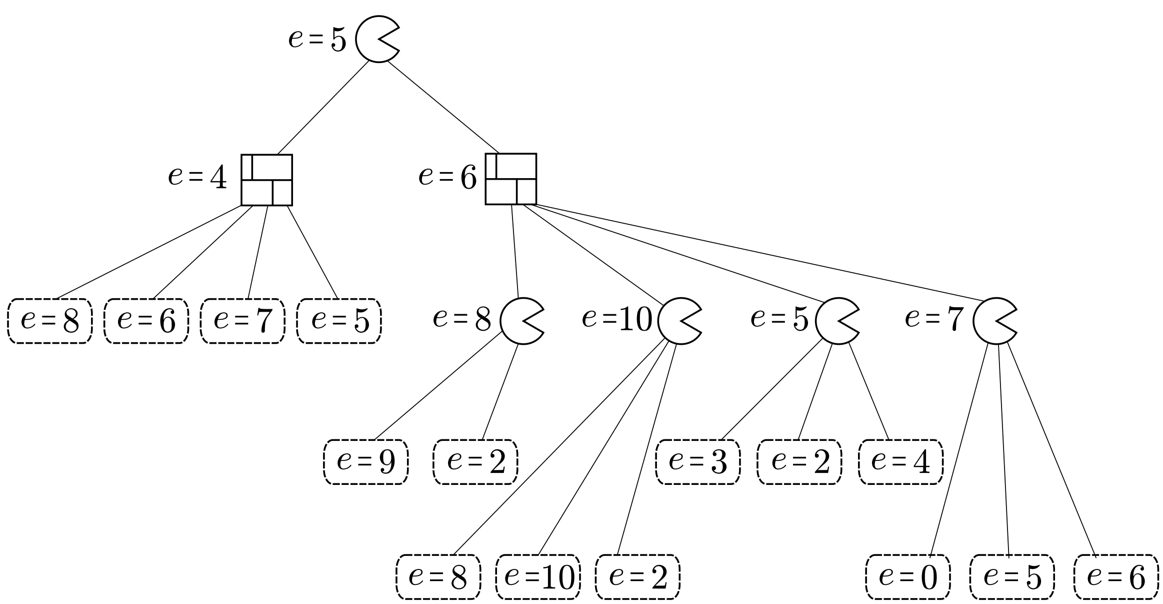
$\boxed{}$

(2)

$\boxed{}$

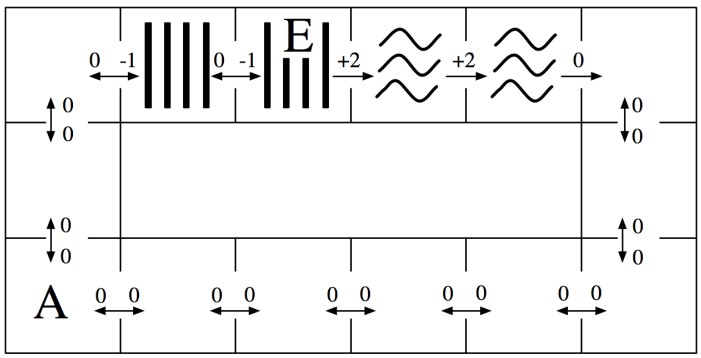
The same game tree is shown below, with the evaluation function applied to *internal* as well as leaf nodes.

(g) In the game tree below cross off any nodes that can be pruned assuming the special property holds true. If not sure you correctly formalized into pseudo-code your intuition on how to exploit the special property for improved pruning, make sure to annotate your pruned nodes with a brief explanation of why each of them was pruned.



5 . MDPs: Grid-World Water Park

Consider the MDP drawn below. The state space consists of all squares in a grid-world water park. There is a single waterslide that is composed of two ladder squares and two slide squares (marked with vertical bars and squiggly lines respectively). An agent in this water park can move from any square to any neighboring square, unless the current square is a slide in which case it must move forward one square along the slide. The actions are denoted by arrows between squares on the map and all deterministically move the agent in the given direction. The agent cannot stand still: it must move on each time step. Rewards are also shown below: the agent feels great pleasure as it slides down the water slide (+2), a certain amount of discomfort as it climbs the rungs of the ladder (-1), and receives rewards of 0 otherwise. The time horizon is infinite; this MDP goes on forever.

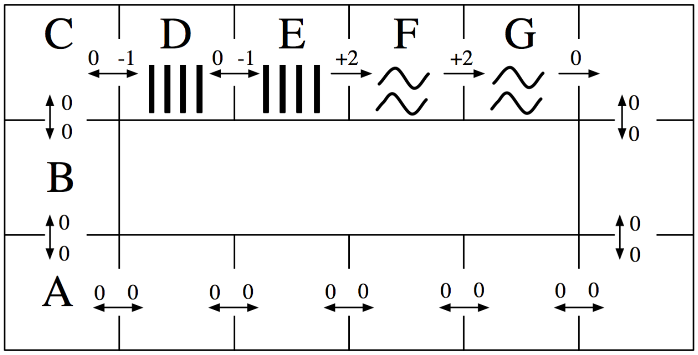


(a) How many (deterministic) policies π are possible for this MDP?

(b) Fill in the blank cells of this table with values that are correct for the corresponding function, discount, and state. *Hint: You should not need to do substantial calculation here.*

	γ	$s = A$	$s = E$
$V_3^*(s)$	1.0		
$V_{10}^*(s)$	1.0		
$V_{10}^*(s)$	0.1		
$Q_1^*(s, \text{west})$	1.0	——	
$Q_{10}^*(s, \text{west})$	1.0	——	
$V^*(s)$	1.0		
$V^*(s)$	0.1		

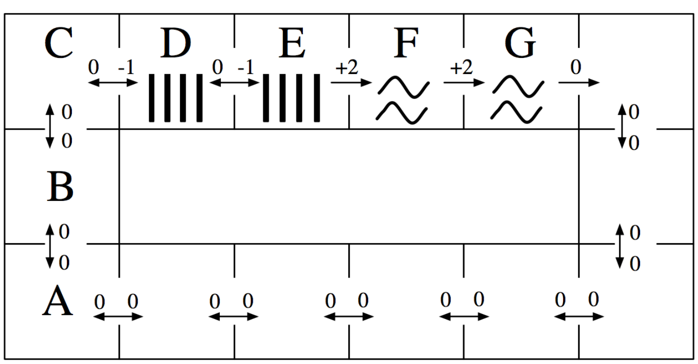
Use this labeling of the state space to complete the remaining subproblems:



(c) Fill in the blank cells of this table with the Q-values that result from applying the Q-update for the transition specified on each row. You may leave Q-values that are unaffected by the current update blank. Use discount $\gamma = 1.0$ and learning rate $\alpha = 0.5$. Assume all Q-values are initialized to 0. (Note: the specified transitions would not arise from a single episode.)

	$Q(D, \text{west})$	$Q(D, \text{east})$	$Q(E, \text{west})$	$Q(E, \text{east})$
Initial:	0	0	0	0
Transition 1: $(s = D, a = \text{east}, r = -1, s' = E)$				
Transition 2: $(s = E, a = \text{east}, r = +2, s' = F)$				
Transition 3: $(s = E, a = \text{west}, r = 0, s' = D)$				
Transition 4: $(s = D, a = \text{east}, r = -1, s' = E)$				

The agent is still at the water park MDP, but now we're going to use function approximation to represent Q-values. Recall that a policy π is *greedy* with respect to a set of Q-values as long as $\forall a, s\ Q(s, \pi(s)) \geq Q(s, a)$ (so ties may be broken in any way).



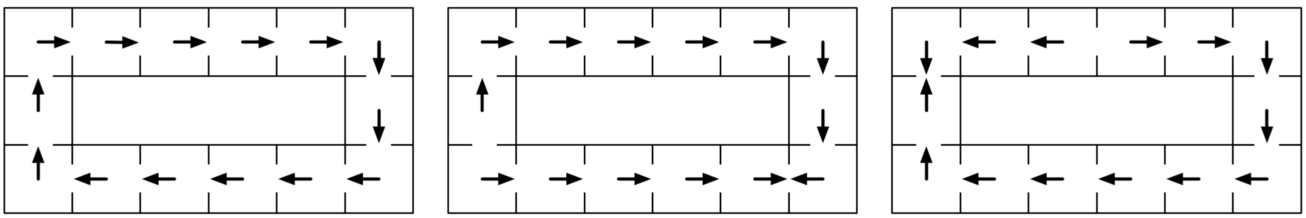
For the next subproblem, consider the following feature functions:

$$f(s,a) = \begin{cases} 1 & \text{if } a = \text{east}, \\ 0 & \text{otherwise.} \end{cases}$$

$$f'(s,a) = \begin{cases} 1 & \text{if } (a = \text{east}) \wedge \text{isSlide}(s), \\ 0 & \text{otherwise.} \end{cases}$$

(Note: isSlide(s) is true iff the state s is a slide square, i.e. either F or G.)

Also consider the following policies:



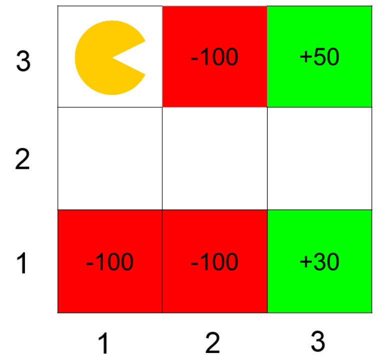
π_1 π_2 π_3

(d) Which are greedy policies with respect to the Q-value approximation function obtained by running the single Q-update for the transition ($s = F, a = \text{east}, r = +2, s' = G$) while using the specified feature function? You may assume that all feature weights are zero before the update. Use discount $\gamma = 1.0$ and learning rate $\alpha = 1.0$. Circle all that apply.

f	π_1	π_2	π_3
f'	π_1	π_2	π_3

6 . Deep inside Q-learning

Consider the grid-world given below and an agent who is trying to learn the optimal policy. Rewards are only awarded for taking the *Exit* action from one of the shaded states. Taking this action moves the agent to the Done state, and the MDP terminates. Assume $\gamma = 1$ and $\alpha = 0.5$ for all calculations. All equations need to explicitly mention γ and α if necessary.



(a) The agent starts from the top left corner and you are given the following episodes from runs of the agent through this grid-world. Each line in an Episode is a tuple containing (s, a, s', r).

Episode 1	Episode 2	Episode 3	Episode 4	Episode 5
(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0
(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0
(2,2), E, (3,2), 0	(2,2), S, (2,1), 0	(2,2), E, (3,2), 0	(2,2), E, (3,2), 0	(2,2), E, (3,2), 0
(3,2), N, (3,3), 0	(2,1), Exit, D, -100	(3,2), S, (3,1), 0	(3,2), N, (3,3), 0	(3,2), S, (3,1), 0
(3,3), Exit, D, +50		(3,1), Exit, D, +30	(3,3), Exit, D, +50	(3,1), Exit, D, +30

Fill in the following Q-values obtained from direct evaluation from the samples:

Q((3,2), N) = ________ Q((3,2), S) = ________ Q((2,2), E) = ________

(b) Q-learning is an online algorithm to learn optimal Q-values in an MDP with unknown rewards and transition function. The update equation is:

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \max_{a'} Q(s_{t+1}, a'))$$

where γ is the discount factor, α is the learning rate and the sequence of observations are $(\cdots, s_t, a_t, s_{t+1}, r_t, \cdots)$. Given the episodes in (a), fill in the time at which the following Q values first become non-zero. Your answer should be of the form (**episode#,iter#**) where **iter#** is the Q-learning update iteration in that episode. If the specified Q value never becomes non-zero, write *never*.

Q((1,2), E) = ________ Q((2,2), E) = ________ Q((3,2), S) = ________

(c) In Q-learning, we look at a window of (s_t, a_t, s_{t+1}, r_t) to update our Q-values. One can think of using an update rule that uses a larger window to update these values. Give an update rule for $Q(s_t, a_t)$ given the window $(s_t, a_t, r_t, s_{t+1}, a_{t+1}, r_{t+1}, s_{t+2})$.

$Q(s_t, a_t) =$

$Q(s_t, a_t) =$

$Q(s_t, a_t) =$

CS188 Fall 2018 Section 6: Probability + Bayes' Nets

1 Probability

Use the probability table to calculate the following values:

X_1	X_2	X_3	$P(X_1, X_2, X_3)$
0	0	0	0.05
1	0	0	0.1
0	1	0	0.4
1	1	0	0.1
0	0	1	0.1
1	0	1	0.05
0	1	1	0.2
1	1	1	0.0

1. $P(X_1 = 1, X_2 = 0)$

2. $P(X_3 = 0)$

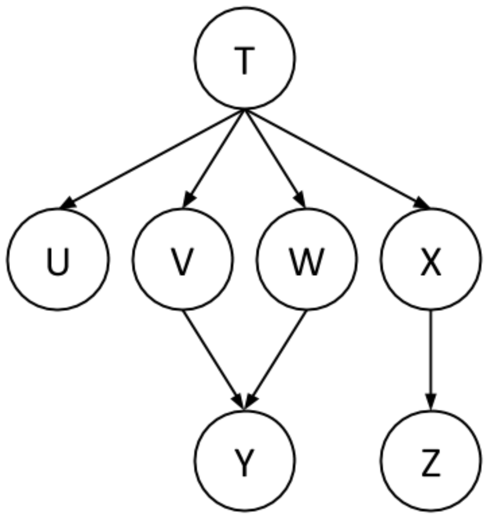
3. $P(X_2 = 1|X_3 = 1)$

4. $P(X_1 = 0|X_2 = 1, X_3 = 1)$

5. $P(X_1 = 0, X_2 = 1|X_3 = 1)$

2 D-Separation

Indicate whether each of the following conditional independence relationships is guaranteed to be true in the Bayes Net below. If the independence relationship does not hold, identify all active (d-connected) paths in the graph.



1. $U \perp\!\!\!\perp X$
2. $U \perp\!\!\!\perp X|T$
3. $V \perp\!\!\!\perp W|Y$
4. $V \perp\!\!\!\perp W|T$
5. $T \perp\!\!\!\perp Y|V$
6. $Y \perp\!\!\!\perp Z|W$
7. $Y \perp\!\!\!\perp Z|T$

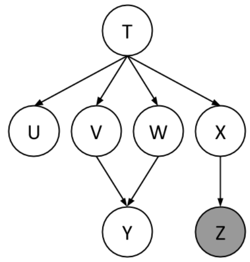
3 Variable Elimination

Using the same Bayes Net (shown below), we want to compute $P(Y \mid +z)$. All variables have binary domains. Assume we run variable elimination to compute the answer to this query, with the following variable elimination ordering: X, T, U, V, W.

Complete the following description of the factors generated in this process:

After inserting evidence, we have the following factors to start out with:

$$P(T), P(U|T), P(V|T), P(W|T), P(X|T), P(Y|V,W), P(+z|X)$$



(a) When eliminating X we generate a new factor f_1 as follows, which leaves us with the factors:

$$f_1(+z|T) = \sum_x P(x|T)P(+z|x) \qquad P(T), P(U|T), P(V|T), P(W|T), P(Y|V,W), f_1(+z|T)$$

(b) When eliminating T we generate a new factor f_2 as follows, which leaves us with the factors:

(c) When eliminating U we generate a new factor f_3 as follows, which leaves us with the factors:

(d) When eliminating V we generate a new factor f_4 as follows, which leaves us with the factors:

(e) When eliminating W we generate a new factor f_5 as follows, which leaves us with the factors:

(f) How would you obtain $P(Y \mid +z)$ from the factors left above:

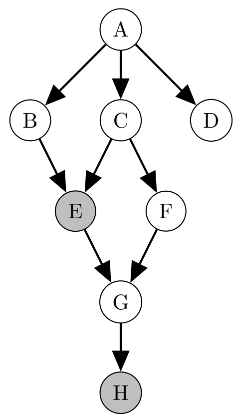
(g) What is the size of the largest factor that gets generated during the above process?

(m) Does there exist a better elimination ordering (one which generates smaller largest factors)?

CS188 Fall 2018 Section 7: Bayes Nets and Decision Nets

1 Bayes' Nets: Inference

Assume we are given the following Bayes' net, and would like to perform inference to obtain $P(B, D \mid E = e, H = h)$.



1. What is the number of rows in the largest factor generated by *inference by enumeration*, for this query $P(B, D \mid E = e, H = h)$? Assume all the variables are binary.

   ○ 2^2 ○ 2^3 ○ 2^6 ○ 2^8
   ○ None of the above.

2. Mark all of the following variable elimination orderings that are optimal for calculating the answer for the query $P(B, D \mid E = e, H = h)$. Optimality is measured by the sum of the sizes of the factors that are generated. Assume all the variables are binary.

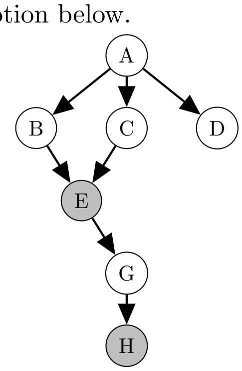
   ☐ C, A, F, G ☐ F, G, C, A ☐ A, C, F, G ☐ G, F, C, A
   ☐ None of the above.

3. Suppose we decide to perform variable elimination to calculate the query $P(B, D \mid E = e, H = h)$, and choose to eliminate F first.

   (a) When F is eliminated, what intermediate factor is generated and how is it calculated? Make sure it is clear which variable(s) come before the conditioning bar and which variable(s) come after.

   $$f_1(\_\_\_\_\_\_\_ \mid \_\_\_\_\_\_\_) = \sum_f \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$$

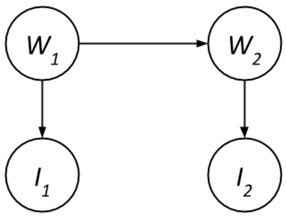
   (b) Now consider the set of distributions that can be represented by the remaining factors *after F is eliminated*. Draw the minimal number of directed edges on the following Bayes' Net structure, so that it can represent any distribution in this set. If no additional directed edges are needed, please fill in that option below.



   ○ No additional directed edges needed

2 Sampling and Dynamic Bayes Nets

We would like to analyze people's ice cream eating habits on sunny and rainy days Suppose we consider the weather, along with a person's ice-cream eating, over the span of two days. We'll have four random variables: W_1 and W_2 stand for the weather on days 1 and 2, which can either be rainy `R` or sunny `S`, and the variables I_1 and I_2 represent whether or not the person ate ice cream on days 1 and 2, and take values `T` (for truly eating ice cream) or `F`. We can model this as the following Bayes Net with these probabilities.



W_1	$P(W_1)$
S	0.6
R	0.4

W_1	W_2	$P(W_2 \mid W_1)$
S	S	0.7
S	R	0.3
R	S	0.5
R	R	0.5

W	I	$P(I \mid W)$
S	T	0.9
S	F	0.1
R	T	0.2
R	F	0.8

Suppose we produce the following samples of (W_1, I_1, W_2, I_2) from the ice-cream model:

`R, F, R, F   R, F, R, F   S, F, S, T   S, T, S, T   S, T, R, F`
`R, F, R, T   S, T, S, T   S, T, S, T   S, T, R, F   R, F, S, T`

1. What is $\widehat{P}(W_2 = \texttt{R})$, the probability that sampling assigns to the event $W_2 = \texttt{R}$?

2. Cross off samples above which are rejected by rejection sampling if we're computing $P(W_2 \mid I_1 = \texttt{T}, I_2 = \texttt{F})$.

   Rejection sampling seems to be wasting a lot of effort, so we decide to switch to likelihood weighting. Assume we generate the following six samples given the evidence $I_1 = \texttt{T}$ and $I_2 = \texttt{F}$:

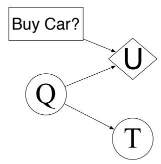
$$(W_1, I_1, W_2, I_2) = \Big\{(\texttt{S},\texttt{T},\texttt{R},\texttt{F}), (\texttt{R},\texttt{T},\texttt{R},\texttt{F}), (\texttt{S},\texttt{T},\texttt{R},\texttt{F}), (\texttt{S},\texttt{T},\texttt{S},\texttt{F}), (\texttt{S},\texttt{T},\texttt{S},\texttt{F}), (\texttt{R},\texttt{T},\texttt{S},\texttt{F})\Big\}$$

3. What is the weight of the first sample (`S`, `T`, `R`, `F`) above?

4. Use likelihood weighting to estimate $P(W_2 \mid I_1 = \texttt{T}, I_2 = \texttt{F})$.

3 Decision Networks and VPI

A used car buyer can decide to carry out various tests with various costs (e.g., kick the tires, take the car to a qualified mechanic) and then, depending on the outcome of the tests, decide which car to buy. We will assume that the buyer is deciding whether to buy car c and that there is time to carry out at most one test which costs \$50 and which can help to figure out the quality of the car. A car can be in good shape (of good quality $Q = +q$) or in bad shape (of bad quality Q=¬q), and the test might help to indicate what shape the car is in. There are only two outcomes for the test T: pass (T=pass) or fail (T=fail). Car c costs \$1,500, and its market value is \$2,000 if it is in good shape; if not, \$700 in repairs will be needed to make it in good shape. The buyers estimate is that c has 70% chance of being in good shape. The Decision Network is shown below.



1. Calculate the expected net gain from buying car c, given no test.

2. Tests can be described by the probability that the car will pass or fail the test given that the car is in good or bad shape. We have the following information:

$$P(T = \text{pass}|Q = +q) = 0.9$$
$$P(T = \text{pass}|Q = \neg q) = 0.2$$

   Calculate the probability that the car will pass (or fail) its test, and then the probability that it is in good (or bad) shape given each possible test outcome.

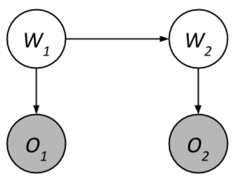
3. Calculate the optimal decisions given either a pass or a fail, and their expected utilities.

4. Calculate the value of (perfect) information of the test. Should the buyer pay for a test?

CS188 Fall 2018 Section 8: HMMs + Particle Filtering

1 HMMs

Consider the following Hidden Markov Model.



W_1	$P(W_1)$
0	0.3
1	0.7

W_t	W_{t+1}	$P(W_{t+1}\|W_t)$
0	0	0.4
0	1	0.6
1	0	0.8
1	1	0.2

W_t	O_t	$P(O_t\|W_t)$
0	A	0.9
0	B	0.1
1	A	0.5
1	B	0.5

Suppose that we observe $O_1 = A$ and $O_2 = B$.
Using the forward algorithm, compute the probability distribution $P(W_2|O_1 = A, O_2 = B)$ one step at a time.

1. Compute $P(W_1, O_1 = A)$.

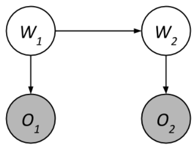
2. Using the previous calculation, compute $P(W_2, O_1 = A)$.

3. Using the previous calculation, compute $P(W_2, O_1 = A, O_2 = B)$.

4. Finally, compute $P(W_2|O_1 = A, O_2 = B)$.

2 Particle Filtering

Let's use Particle Filtering to estimate the distribution of $P(W_2|O_1 = A, O_2 = B)$. Here's the HMM again:



W_1	$P(W_1)$
0	0.3
1	0.7

W_t	W_{t+1}	$P(W_{t+1}\|W_t)$
0	0	0.4
0	1	0.6
1	0	0.8
1	1	0.2

W_t	O_t	$P(O_t\|W_t)$
0	A	0.9
0	B	0.1
1	A	0.5
1	B	0.5

We start with two particles representing our distribution for W_1.
$P_1 : W_1 = 0$
$P_2 : W_1 = 1$
Use the following random numbers to run particle filtering:

$$[0.22, 0.05, 0.33, 0.20, 0.84, 0.54, 0.79, 0.66, 0.14, 0.96]$$

1. **Observe**: Compute the weight of the two particles after evidence $O_1 = A$.

2. **Resample**: Using the random numbers, resample P_1 and P_2 based on the weights.

3. **Elapse Time**: Now let's compute the elapse time particle update. Sample P_1 and P_2 from applying the time update.

4. **Observe**: Compute the weight of the two particles after evidence $O_2 = B$.

5. **Resample**: Using the random numbers, resample P_1 and P_2 based on the weights.

6. What is our estimated distribution for $P(W_2|O_1 = A, O_2 = B)$?

3 HMMs (Optional)

Consider a process where there are transitions among a finite set of states $s_1, \cdots, s_k$ over time steps $i = 1, \cdots, N$. Let the random variables $X_1, \cdots, X_N$ represent the state of the system at each time step and be generated as follows:

- Sample the initial state s from an initial distribution $P_1(X_1)$, and set $i = 1$
- Repeat the following:
  1. Sample a duration d from a duration distribution P_D over the integers $\{1, \cdots, M\}$, where M is the maximum duration.
  2. Remain in the current state s for the next d time steps, i.e., set
  $$x_i = x_{i+1} = \cdots = x_{i+d-1} = s \tag{1}$$
  3. Sample a successor state s' from a transition distribution $P_T(X_t|X_{t-1} = s)$ over the other states $s' \neq s$ (so there are no self transitions)
  4. Assign $i = i + d$ and $s = s'$.

This process continues indefinitely, but we only observe the first N time steps.

(a) Assuming that all three states s_1, s_2, s_3 are different, what is the probability of the sample sequence $s_1, s_1, s_2, s_2, s_2, s_3, s_3$? Write an algebraic expression. Assume $M \geq 3$.

At each time step i we observe a noisy version of the state X_i that we denote Y_i and is produced via a conditional distribution $P_E(Y_i|X_i)$.

(b) Only in this subquestion assume that $N > M$. Let $X_1, \cdots, X_N$ and $Y_1, \cdots, Y_N$ random variables defined as above. What is the maximum index $i \leq N - 1$ so that $X_1 \perp\!\!\!\perp X_N | X_i, X_{i+1}, \cdots, X_{N-1}$ is guaranteed?

(c) Only in this subquestion, assume the max duration $M = 2$, and P_D uniform over $\{1, 2\}$ and each x_i is in an alphabet $\{a, b\}$. For $(X_1, X_2, X_3, X_4, X_5, Y_1, Y_2, Y_3, Y_4, Y_5)$ draw a Bayes Net over these 10 random variables with the property that removing any of the edges would yield a Bayes net inconsistent with the given distribution.

(d) In this part we will explore how to write the described process as an HMM with an extended state space. Write the states $z = (s, t)$ where s is a state of the original system and t represents the time elapsed in that state. For example, the state sequence $s_1, s_1, s_1, s_2, s_3, s_3$ would be represented as $(s_1, 1), (s_1, 2), (s_1, 3), (s_2, 1), (s_3, 1), (s_3, 2)$.

Answer all of the following in terms of the parameters $P_1(X_1), P_D(d), P_T(X_{j+1}|X_j), P_E(Y_i|X_i), k$ (total number of possible states), N and M (max duration).

- What is $P(Z_1)$?

  $P(x_1, t_1) =$

- What is $P(Z_{i+1}|Z_i)$? Hint: You will need to break this into cases where the transition function will behave differently.

  $P(X_{i+1}, t_{i+1} \mid X_i, t_i) =$

- What is $P(Y_i|Z_i)$?

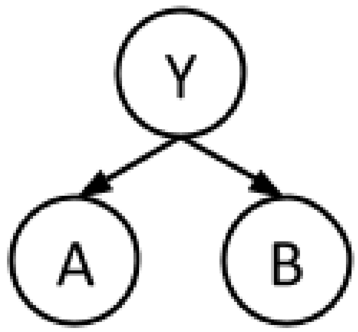
  $P(Y_i \mid X_i, t_i) =$

CS188 Fall 2018 Section 9: Machine Learning

1 Naive Bayes

In this question, we will train a Naive Bayes classifier to predict class labels Y as a function of input features A and B. Y, A, and B are all binary variables, with domains 0 and 1. We are given 10 training points from which we will estimate our distribution.

A	1	1	1	1	0	1	0	1	1	1
B	1	0	0	1	1	1	1	0	1	1
Y	1	1	0	0	0	1	1	0	0	0



1. What are the maximum likelihood estimates for the tables $P(Y)$, $P(A|Y)$, and $P(B|Y)$?

Y	$P(Y)$
0	
1	

A	Y	$P(A\|Y)$
0	0	
1	0	
0	1	
1	1	

B	Y	$P(B\|Y)$
0	0	
1	0	
0	1	
1	1	

2. Consider a new data point ($A = 1$, $B = 1$). What label would this classifier assign to this sample?

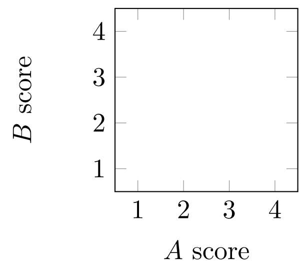
3. Let's use Laplace Smoothing to smooth out our distribution. Compute the new distribution for $P(A|Y)$ given Laplace Smoothing with $k = 2$.

A	Y	$P(A\|Y)$
0	0	
1	0	
0	1	
1	1	

2 Perceptron

You want to predict if movies will be profitable based on their screenplays. You hire two critics A and B to read a script you have and rate it on a scale of 1 to 4. The critics are not perfect; here are five data points including the critics' scores and the performance of the movie:

#	Movie Name	A	B	Profit?
1	Pellet Power	1	1	-
2	Ghosts!	3	2	+
3	Pac is Bac	2	4	+
4	Not a Pizza	3	4	+
5	Endless Maze	2	3	-



1. First, you would like to examine the linear separability of the data. Plot the data on the 2D plane above; label profitable movies with + and non-profitable movies with $-$ and determine if the data are linearly separable.

2. Now you decide to use a perceptron to classify your data. Suppose you directly use the scores given above as features, together with a bias feature. That is $f_0 = 1$, $f_1 =$ score given by A and $f_2 =$ score given by B.

   Run one pass through the data with the perceptron algorithm, filling out the table below. Go through the data points in order, e.g. using data point #1 at step 1.

step	Weights	Score	Correct?
1	[-1, 0, 0]	$-1 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 = -1$	yes
2			
3			
4			
5			

   Final weights:

3. Have weights been learned that separate the data?

4. More generally, irrespective of the training data, you want to know if your features are powerful enough to allow you to handle a range of scenarios. Circle the scenarios for which a perceptron using the features above can indeed perfectly classify movies which are profitable according to the given rules:

   (a) Your reviewers are awesome: if the total of their scores is more than 8, then the movie will definitely be profitable, and otherwise it won't be.

   (b) Your reviewers are art critics. Your movie will be profitable if and only if each reviewer gives either a score of 2 or a score of 3.

   (c) Your reviewers have weird but different tastes. Your movie will be profitable if and only if both reviewers agree.

3 Maximum Likelihood

A Geometric distribution is a probability distribution of the number X of Bernoulli trials needed to get one success. It depends on a parameter p, which is the probability of success for each individual Bernoulli trial. Think of it as the number of times you must flip a coin before flipping heads. The probability is given as follows:

$$P(X = k) = p(1 - p)^{k-1} \tag{1}$$

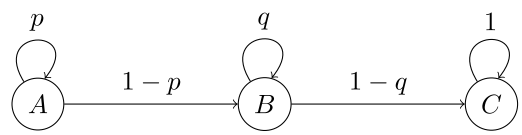
p is the parameter we wish to estimate.

We observe the following samples from a Geometric distribution: $x_1 = 5$, $x_2 = 8$, $x_3 = 3$, $x_4 = 5$, $x_5 = 7$. What is the maximum likelihood estimate for p?

CS188 Fall 2018 Section 9: Midterm 2 Prep

1 . A Not So Random Walk

Pacman is trying to predict the position of a ghost, which he knows has the following transition graph:



Here, $0 < p < 1$ and $0 < q < 1$ are arbitrary probabilities. It is known that the ghost always starts in state A. For this problem, we consider time to begin at 0. For example, at time 0, the ghost is in A with probability 1, and at time 1, the ghost is in A with probability p or in B with probability $1 - p$.

In all of the following questions, you may assume that n is large enough so that the given event occurs with non-zero probability.

(i) Suppose $p \neq q$. What is the probability that the ghost is in A at time n?

(ii) Suppose $p \neq q$. What is the probability that the ghost first reaches B at time n?

(iii) Suppose $p \neq q$. What is the probability that the ghost is in B at time n?

(iv) Suppose $p \neq q$. What is the probability that the ghost first reaches C at time n?

(v) Suppose $p \neq q$. What is the probability that the ghost is in C at time n?

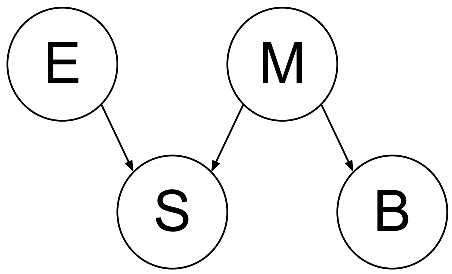
2 . December 21, 2012

A smell of sulphur (S) can be caused either by rotten eggs (E) or as a sign of the doom brought by the Mayan Apocalypse (M). The Mayan Apocalypse also causes the oceans to boil (B). The Bayesian network and corresponding conditional probability tables for this situation are shown below. For each part, you should give either a numerical answer (e.g. 0.81) or an arithmetic expression in terms of numbers from the tables below (e.g. $0.9 \cdot 0.9$).

Note: be careful of doing unnecessary computation here.

$P(E)$	
$+e$	0.4
$-e$	0.6

$P(S\|E,M)$			
$+e$	$+m$	$+s$	1.0
$+e$	$+m$	$-s$	0.0
$+e$	$-m$	$+s$	0.8
$+e$	$-m$	$-s$	0.2
$-e$	$+m$	$+s$	0.3
$-e$	$+m$	$-s$	0.7
$-e$	$-m$	$+s$	0.1
$-e$	$-m$	$-s$	0.9



$P(M)$	
$+m$	0.1
$-m$	0.9

$P(B\|M)$		
$+m$	$+b$	1.0
$+m$	$-b$	0.0
$-m$	$+b$	0.1
$-m$	$-b$	0.9

(a) Compute the following entry from the joint distribution:

$P(-e, -s, -m, -b) =$

(b) What is the probability that the oceans boil?

$P(+b) =$

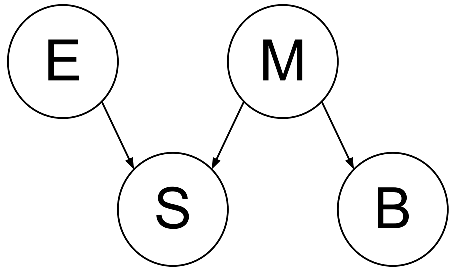
(c) What is the probability that the Mayan Apocalypse is occurring, given that the oceans are boiling?

$P(+m| + b) =$

The figures and table below are identical to the ones on the previous page and are repeated here for your convenience.

$P(E)$	
$+e$	0.4
$-e$	0.6

$P(S\|E,M)$			
$+e$	$+m$	$+s$	1.0
$+e$	$+m$	$-s$	0.0
$+e$	$-m$	$+s$	0.8
$+e$	$-m$	$-s$	0.2
$-e$	$+m$	$+s$	0.3
$-e$	$+m$	$-s$	0.7
$-e$	$-m$	$+s$	0.1
$-e$	$-m$	$-s$	0.9



$P(M)$	
$+m$	0.1
$-m$	0.9

$P(B\|M)$		
$+m$	$+b$	1.0
$+m$	$-b$	0.0
$-m$	$+b$	0.1
$-m$	$-b$	0.9

(d) What is the probability that the Mayan Apocalypse is occurring, given that there is a smell of sulphur, the oceans are boiling, and there are rotten eggs?

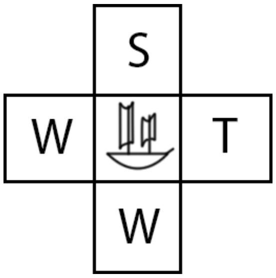
$P(+m|+s,+b,+e) =$

(e) What is the probability that rotten eggs are present, given that the Mayan Apocalypse is occurring?

$P(+e|+m) =$

3 . Argg! Sampling for the Legendary Treasure

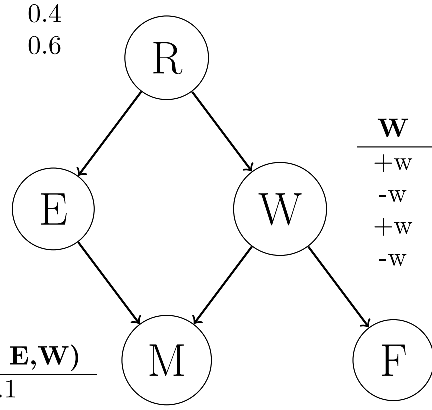
Little did you know that Jasmine and Katie are actually infamous pirates. One day, they go treasure hunting in the Ocean of Bayes, where rumor says a great treasure lies in wait for explorers who dare navigate in the rough waters. After navigating about the ocean, they are within grasp of the treasure. Their current configuration is represented by the boat in the figure below. They can only make one move, and must choose from the actions: (North, South, East, West). Stopping is not allowed. They will land in either a whirlpool (W), an island with a small treasure (S), or an island with the legendary treasure (T). The utilities of the three types of locations are shown below:



State	U(State)
T (Legendary Treasure)	100
S (Small Treasure)	25
W (Whirlpool)	-50

The success of their action depends on the random variable **Movement (M)**, which takes on one of two values: (+m, -m). The Movement random variable has many relationships with other variables: Presence of Enemy Pirates (E), Rain (R), Strong Waves (W), and Presence of Fishermen (F). The Bayes' net graph that represents these relationships is shown below:

R	P(R)
+r	0.4
-r	0.6



E	R	P(E | R)
+e	+r	0.3
-e	+r	0.7
+e	-r	0.6
-e	-r	0.4

W	R	P(W | R)
+w	+r	0.9
-w	+r	0.1
+w	-r	0.2
-w	-r	0.8

M	E	W	P(M | E,W)
+m	+e	+w	0.1
-m	+e	+w	0.9
+m	+e	-w	0.45
-m	+e	-w	0.55
+m	-e	+w	0.35
-m	-e	+w	0.65
+m	-e	-w	0.9
-m	-e	-w	0.1

F	W	P(F | W)
+f	+w	0.15
-f	+w	0.85
+f	-w	0.75
-f	-w	0.25

In the following questions we will follow a two-step process:
– (1) Jasmine and Katie observed the random variables $R = -r$ and $F = +f$. We then determine the distribution for $P(M| - r, +f)$ via sampling.

– (2) Based on the estimate for $P(M| - r, +f)$, after committing to an action, landing in the intended location of an action successfully occurs with probability $P(M = +m| - r, +f)$. The other three possible landing positions occur with probability $\frac{P(M=-m|-r,+f)}{3}$ each. Use this transition distribution to calculate the optimal action(s) to take and the expected utility of those actions.

(a) **(i)** **Rejection Sampling**: You want to estimate $P(M = +m | -r, +f)$ by rejection sampling. Below is a list of samples that were generated using prior sampling. Cross out those that would be rejected by rejection sampling.

$+r$	$+e$	$+w$	$-m$	$-f$	$-r$	$-e$	$+w$	$-m$	$+f$
$-r$	$-e$	$+w$	$-m$	$-f$	$+r$	$-e$	$+w$	$+m$	$-f$
$-r$	$+e$	$-w$	$-m$	$+f$	$-r$	$-e$	$-w$	$+m$	$+f$
$+r$	$-e$	$-w$	$+m$	$-f$	$+r$	$-e$	$-w$	$+m$	$+f$
$-r$	$-e$	$-w$	$-m$	$+f$	$-r$	$+e$	$+w$	$-m$	$+f$
$-r$	$+e$	$-w$	$-m$	$+f$	$-r$	$+e$	$-w$	$-m$	$+f$

(ii) What is the approximation for $P(M = +m | -r, +f)$ using the remaining samples?

(iii) What are the optimal action(s) for Jasmine and Katie based on this estimate of $P(M = +m | -r, +f)$?

(iv) What is the expected utility for the optimal action(s) based on this estimate of $P(M = +m | -r, +f)$?

(b) **(i)** **Likelihood Weighting:** Suppose instead that you perform likelihood weighting on the following samples to get the estimate for $P(M = +m | -r, +f)$. You receive 4 samples consistent with the evidence.

Sample	**Weight**
$-r \quad -e \quad +w \quad +m \quad +f$	
$-r \quad -e \quad -w \quad +m \quad +f$	
$-r \quad -e \quad +w \quad -m \quad +f$	
$-r \quad +e \quad -w \quad -m \quad +f$	

(ii) What is the approximation for $P(M = +m | -r, +f)$ using the samples above?

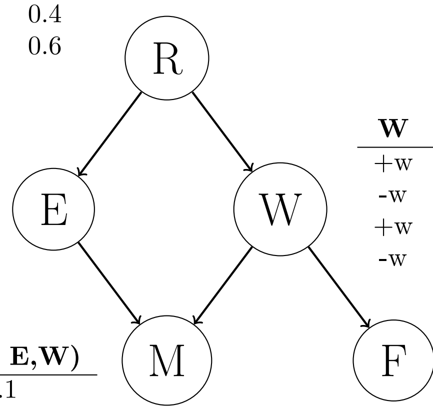
(iii) What are the optimal action(s) for Jasmine and Katie based on this estimate of $P(M = +m | -r, +f)$?

(iv) What is the expected utility for the optimal action(s) based on this estimate of $P(M = +m | -r, +f)$?

Here is a copy of the Bayes' Net, repeated for your convenience.

R	P(R)
+r	0.4
-r	0.6

E	R	P(E | R)
+e	+r	0.3
-e	+r	0.7
+e	-r	0.6
-e	-r	0.4



W	R	P(W | R)
+w	+r	0.9
-w	+r	0.1
+w	-r	0.2
-w	-r	0.8

M	E	W	P(M | E,W)
+m	+e	+w	0.1
-m	+e	+w	0.9
+m	+e	-w	0.45
-m	+e	-w	0.55
+m	-e	+w	0.35
-m	-e	+w	0.65
+m	-e	-w	0.9
-m	-e	-w	0.1

F	W	P(F | W)
+f	+w	0.15
-f	+w	0.85
+f	-w	0.75
-f	-w	0.25

(c) **(i)** **Gibbs Sampling**. Now, we tackle the same problem, this time using Gibbs sampling. We start out with initializing our evidence: $R = -r$, $F = +f$. Furthermore, we start with this random sample:

$-r \ \ +e \ \ -w \ \ +m \ \ +f.$

We select variable E to resample. Calculate the numerical value for:
$P(E = +e|R = -r, W = -w, M = +m, F = +f)$.

We resample for a long time until we end up with the sample:

$-r \ \ -e \ \ +w \ \ +m \ \ +f.$

Jasmine and Katie are happy for fixing this one sample, but they do not have enough time left to compute another sample before making a move. They will let this one sample approximate the distribution: $P(M = +m| - r, +f)$.

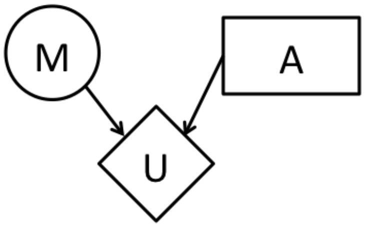
(ii) What is the approximation for $P(M = +m| - r, +f)$, using this one sample?

(iii) What are the optimal action(s) for Jasmine and Katie based on this estimate of $P(M = +m| - r, +f)$?

(iv) What is the expected utility for the optimal action(s) based on this estimate of $P(M = +m| - r, +f)$?

4 . Probability and Decision Networks

The new Josh Bond Movie (M), Skyrise, is premiering later this week. Skyrise will either be great ($+m$) or horrendous ($-m$); there are no other possible outcomes for its quality. Since you are going to watch the movie no matter what, your primary choice is between going to the theater (*theater*) or renting (*rent*) the movie later. Your utility of enjoyment is only affected by these two variables as shown below:



M	P(M)
+m	0.5
-m	0.5

M	A	U(M,A)
+m	*theater*	100
-m	*theater*	10
+m	*rent*	80
-m	*rent*	40

(a) Maximum Expected Utility

Compute the following quantities:

$EU(theater) =$

$EU(rent) =$

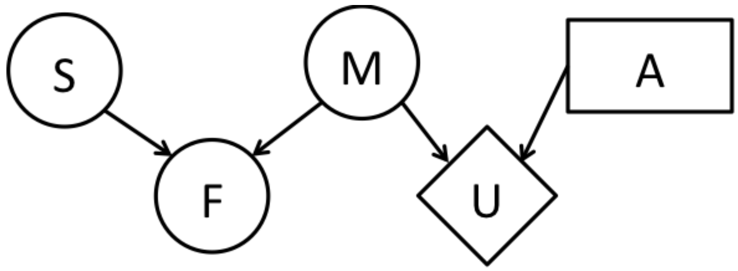
$MEU(\{\}) =$

Which action achieves $MEU(\{\}) =$

(b) Fish and Chips

Skyrise is being released two weeks earlier in the U.K. than the U.S., which gives you the perfect opportunity to predict the movie's quality. Unfortunately, you don't have access to many sources of information in the U.K., so a little creativity is in order.

You realize that a reasonable assumption to make is that if the movie (M) is great, citizens in the U.K. will celebrate by eating fish and chips (F). Unfortunately the consumption of fish and chips is also affected by a possible food shortage (S), as denoted in the below diagram.



The consumption of fish and chips (F) and the food shortage (S) are both binary variables. The relevant conditional probability tables are listed below:

S	M	F	$P(F\|S,M)$
+s	+m	+f	0.6
+s	+m	-f	0.4
+s	-m	+f	0.0
+s	-m	-f	1.0

S	M	F	$P(F\|S,M)$
-s	+m	+f	1.0
-s	+m	-f	0.0
-s	-m	+f	0.3
-s	-m	-f	0.7

S	$P(S)$
+s	0.2
-s	0.8

You are interested in the value of revealing the food shortage node (S). Answer the following queries:

$EU(theater| + s) =$

$EU(rent| + s) =$

$MEU(\{+s\}) =$

Optimal Action Under $\{+s\}$ =

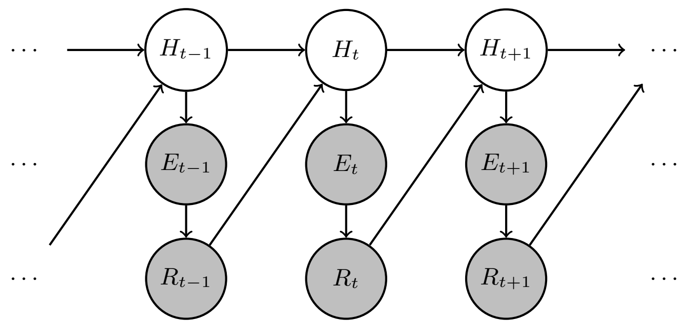
$MEU(\{-s\}) =$

Optimal Action Under $\{-s\}$ =

$VPI(S) =$

5 . HMM: Human-Robot Interaction

In the near future, autonomous robots would live among us. Therefore, it is important for the robots to know how to properly act in the presence of humans. In this question, we are exploring a simplified model of this interaction. Here, we are assuming that we can observe the robot's actions at time t, R_t, and an evidence observation, E_t, directly caused by the human action, H_t. Humans actions and Robots actions from the past time-step affect the Human's and Robot's actions in the next time-step. In this problem, we will remain consistent with the convention that capital letters (H_t) refer to random variables and lowercase letters (h_t) refer to a particular value the random variable can take. The structure is given below:



You are supplied with the following probability tables: $P(R_t \mid E_t)$, $P(H_t \mid H_{t-1}, R_{t-1})$, $P(H_0)$, $P(E_t \mid H_t)$.

Let us derive the forward algorithm for this model. We will split our computation into two components, a **time-elapse update** expression and a **observe update** expression.

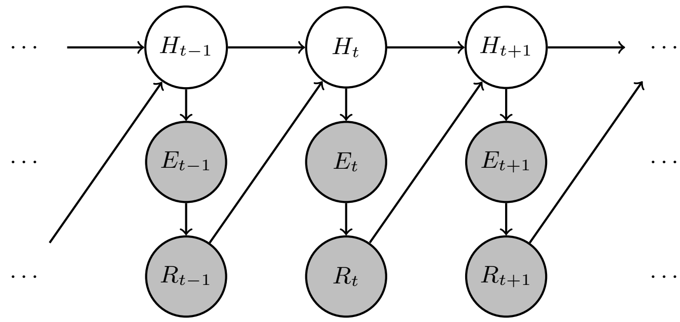
(a) We would like to incorporate the evidence that we observe at time t. Using the time-lapse update expression we will derive separately, we would like to find the **observe update** expression:

$$O(H_t) = P(H_t|e_{0:t}, r_{0:t})$$

In other words, we would like to compute the distribution of potential human states at time t given all observations up to and including time t. In addition to the conditional probability tables associated with the network's nodes, we are given $T(H_t) = P(H_t \mid e_{0:t-1}, r_{0:t-1})$, which we will assume is correctly computed in the time-elapse update that we will derive in the next part. From the options below, select *all* the options that **both** make valid independence assumptions and would evaluate to the observe update expression.

- ☐ $\frac{P(H_t|e_{0:t-1},r_{0:t-1})P(e_t|H_t)P(r_t|e_t)}{\sum_{h_t} P(h_t|e_{0:t-1},r_{0:t-1})P(e_t|h_t)P(r_t|e_t)}$
- ☐ $\frac{P(H_t|e_{0:t-1},r_{0:t-1})P(e_t|H_t)}{\sum_{h_t} P(h_t|e_{0:t-1},r_{0:t-1})P(e_t|h_t)}$
- ☐ $\frac{\sum_{e_t} P(H_t|e_{0:t-1},r_{0:t-1})P(e_t|H_t)}{\sum_{h_t} P(h_t|e_{0:t-1},r_{0:t-1})P(e_t|r_{t-1},H_{t-1})}$
- ☐ $\sum_{r_{t-1}} P(H_t \mid e_{0:t-1}, r_{0:t-1})P(r_{t-1} \mid e_{t-1})$
- ☐ $\sum_{r_t} P(H_t \mid e_{0:t-1}, r_{0:t-1})P(r_t \mid r_{t-1}, e_t)$
- ☐ $\sum_{h_{t+1}} P(H_t \mid e_{0:t-1}, r_{0:t-1})P(h_{t+1} \mid r_t)$

The structure below is identical to the one in the beginning of the question and is repeated for your convenience.



(b) We are interested in predicting what the state of human is at time t (H_t), given all the observations through $t-1$. Therefore, the **time-elapse update** expression has the following form:

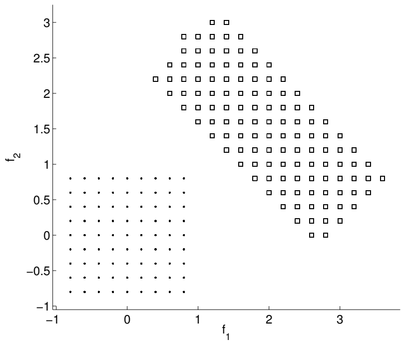
$$T(H_t) = P(H_t|e_{0:t-1}, r_{0:t-1})$$

Derive an expression for the given time-elapse update above using the probability tables provided in the question and the observe update expression, $O(H_{t-1}) = P(H_{t-1}|e_{0:t-1}, r_{0:t-1})$. Write your final expression in the space provided at below. You may use the function O in your solution if you prefer.

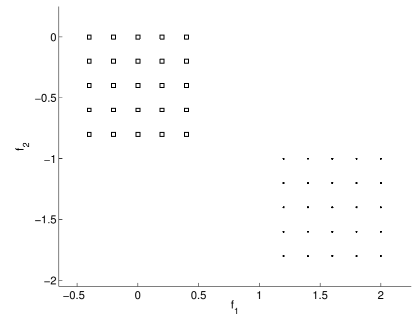
$P(H_t|e_{0:t-1}, r_{0:t-1}) =$ ____

6 . Naïve Bayes Modeling Assumptions

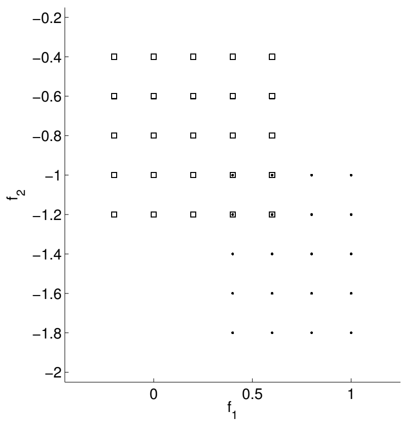
You are given points from 2 classes, shown as rectangles and dots. For each of the following sets of points, mark if they satisfy all the Naïve Bayes modelling assumptions, or they do not satisfy all the Naïve Bayes modelling assumptions. Note that in (c), 4 rectangles overlap with 4 dots.



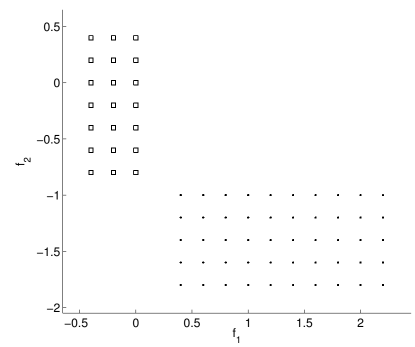
(a) ◯ Satisfies ◯ Does not Satisfy



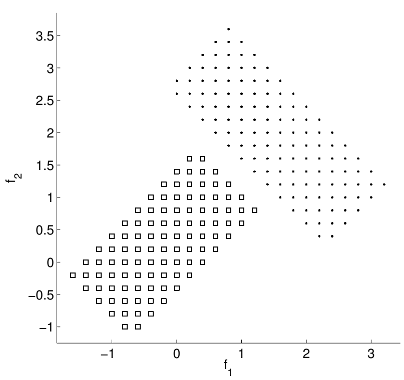
(b) ◯ Satisfies ◯ Does not Satisfy



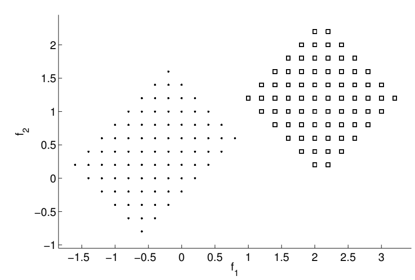
(c) ◯ Satisfies ◯ Does not Satisfy



(d) ◯ Satisfies ◯ Does not Satisfy



(e) ◯ Satisfies ◯ Does not Satisfy



(f) ◯ Satisfies ◯ Does not Satisfy

CS188 Fall 2018 Section 12: Neural Networks and Decision Trees

1 Perceptron $\rightarrow$ Neural Nets

Instead of the standard perceptron algorithm, we decide to treat the perceptron as a single node neural network and update the weights using gradient-based optimization.

In lecture, we covered maximizing likelihood using gradient ascent. We can also choose to **minimize** a loss function that calculates the distance between a prediction and the correct label. The loss function for one data point is $Loss(y, y^*) = \frac{1}{2}(y - y^*)^2$, where y^* is the training label for a given point and y is the output of our single node network for that point.

We will compute a score $z = w_1 x_1 + w_2 x_2$, and then predict the output using an activation function g: $y = g(z)$.

1. 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. We wish to *minimize* the loss, so we will use gradient *descent* (not gradient ascent). What is the update equation for weight w_i given $\frac{\partial Loss}{\partial w_i}$ and learning rate α?

   $$w_i \leftarrow$$

3. For this question, the specific activation function that we will use is

   $$g(z) = 1 \text{ if } z \geq 0 \text{ , or } -1 \text{ if } z < 0$$

   Use gradient descent to update the weights for a single data point. With initial weights of $w_1 = 2$ and $w_2 = -2$, what are the updated weights after processing the data point $(x_1, x_2) = (-1, 2)$, $y^* = 1$?

4. What is the most critical problem with this gradient descent training process with that activation function?

2 Decision Trees

You are a geek who hates sports. Trying to look cool at a party, you join a discussion that you belive to be about football and basketball. You gather information about the two main subjects of discussion, but still cannot figure out what sports they play.

Sport	Position	Name	Height	Weight	Age	College
?	Guard	Charlie Ward	6'02"	185	41	Florida State
?	Defensive End	Julius Peppers	6'07"	283	32	North Carolina

Fortunately, you have brought your CS 188 notes along, and will build some classifiers to determine which sport is being discussed.

You come across a pamphlet from the Atlantic Coast Conference Basketball Hall of Fame, as well as an Oakland Raiders team roster, and create the following table:

Sport	Position	Name	Height	Weight	Age	College
Basketball	Guard	Michael Jordan	6'06"	195	49	North Carolina
Basketball	Guard	Vince Carter	6'06"	215	35	North Carolina
Basketball	Guard	Muggsy Bogues	5'03"	135	47	Wake Forest
Basketball	Center	Tim Duncan	6'11"	260	35	Oklahoma
Football	Center	Vince Carter	6'02"	295	29	Oklahoma
Football	Kicker	Tim Duncan	6'00"	215	33	Oklahoma
Football	Kicker	Sebastian Janikowski	6'02"	250	33	Florida State
Football	Guard	Langston Walker	6'08"	345	33	California

2.1 Entropy

Before we get started, let's review the concept of entropy.

1. Give the definition of entropy for an arbitrary probability distribution $P(X)$.

2. Draw a graph of entropy $H(X)$ vs. $P(X = 1)$ for a binary random variable X.

3. What is the entropy of the distribution of *Sport* in the training data? What about *Position*?

2.2 Decision Trees

Central to decision trees is the concept of "splitting" on a variable.

1. To review the concept of "information gain", calculate it for a split on the *Sport* variable.

2. Of course, in our situation this would not make sense, as *Sport* is the very variable we lack at test time. Now calculate the information gain for the decision "stumps" (one-split trees) created by first splitting on *Position*, *Name*, and *College*. Do any of these perfectly classify the training data? Does it make sense to use *Name* as a variable? Why or why not?

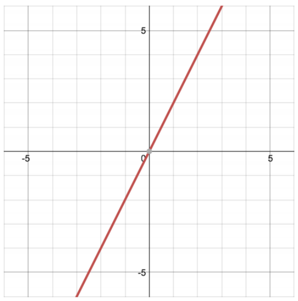
3. Decision trees can represent any function of discrete attribute variables. How can we *best* cast continuous variables (*Height*, *Weight*, and *Age*) into discrete variables?

4. Draw a few decision trees that each correctly classify the training data, and show how their predictions vary on the test set. What algorithm are you following?

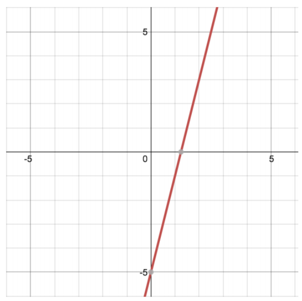
5. You may have noticed that the testing data has a value for *Position* that is missing in training data. What could we do in this case?

3 Neural Network Representations

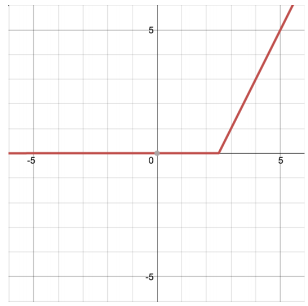
You are given a number of functions (a-h) of a single variable, x, which are graphed below. The computation graphs on the following pages will start off simple and get more complex, building up to neural networks. For each computation graph, indicate which of the functions below they are able to represent.



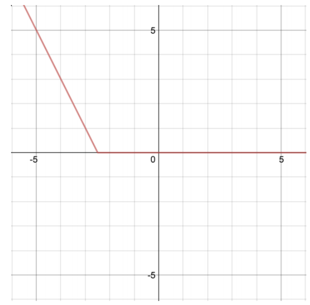
(a) $2x$



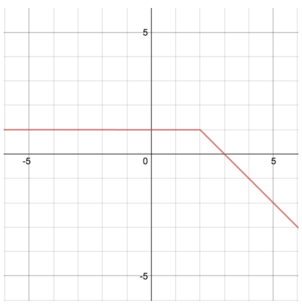
(b) $4x - 5$



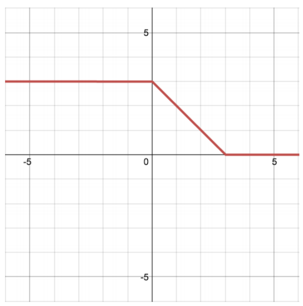
(c) $\begin{cases} 2x-5 & x \geq 2.5 \\ 0 & x < 2.5 \end{cases}$



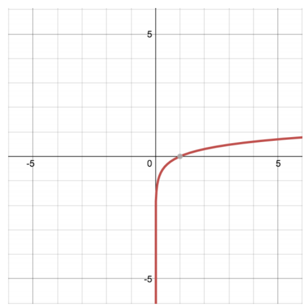
(d) $\begin{cases} -2x-5 & x \leq -2.5 \\ 0 & x > -2.5 \end{cases}$



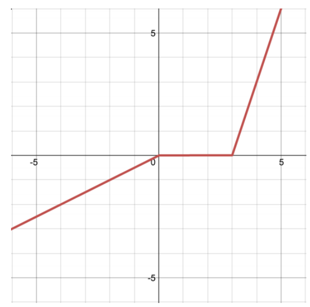
(e) $\begin{cases} -x+3 & x \geq 2 \\ 1 & x < 2 \end{cases}$



(f) $\begin{cases} 3 & x \leq 0 \\ 3-x & 0 < x \leq 3 \\ 0 & x > 3 \end{cases}$

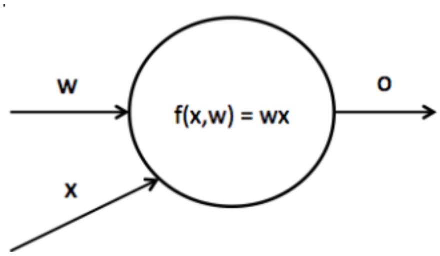


(g) $\log(x)$

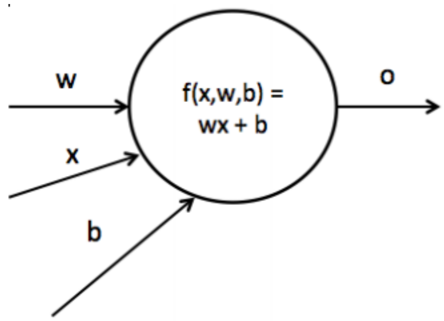


(h) $\begin{cases} 0.5x & x \leq 0 \\ 0 & 0 < x \leq 3 \\ 3x-9 & x > 3 \end{cases}$

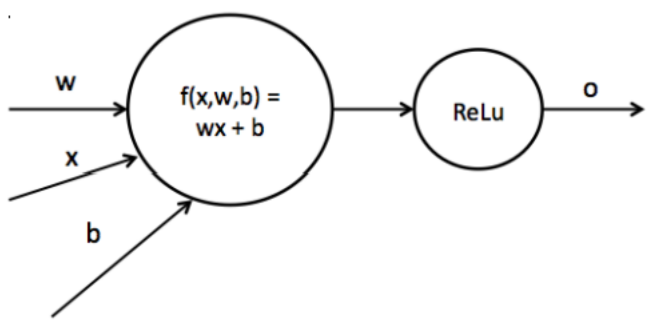
1. Consider the following computation graph, computing a linear transformation with scalar input x, weight w, and output o, such that $o = wx$. Which of the funcions can be represented by this graph? For the options which can, write out the appropriate value of w.



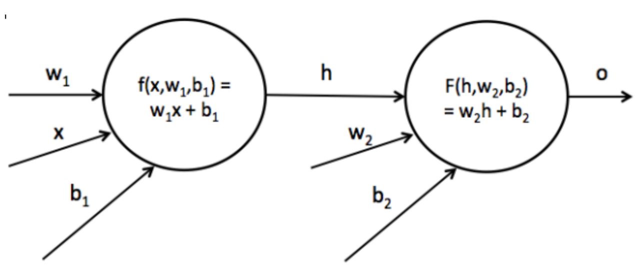
2. Now we introduce a bias term b into the graph, such that $o = wx + b$ (this is known as an *affine* function). Which of the functions can be represented by this network? For the options which can, write out an appropriate value of w, b.



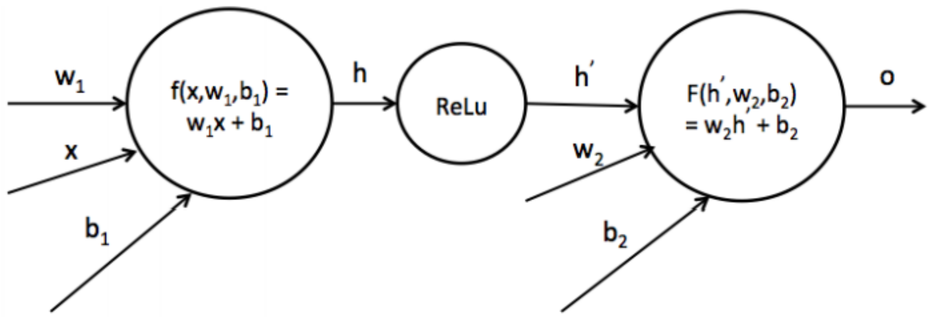
3. We can introduce a non-linearity into the network as indicated below. We use the ReLU non-linearity, which has the form $ReLU(x) = \max(0, x)$. Now which of the functions can be represented by this neural network with weight w and bias b? For the options which can, write out an appropriate value of w, b.



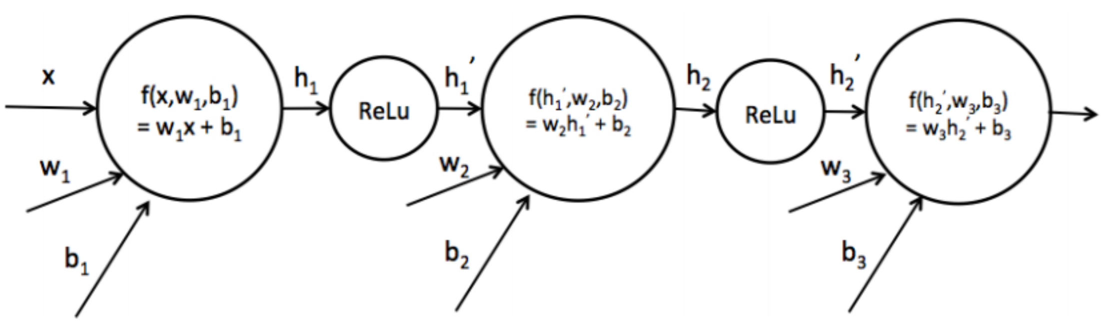
4. Now we consider neural networks with multiple affine transformations, as indicated below. We now have two sets of weights and biases w_1, b_1 and w_2, b_2. We denote the result of the first transformation h such that $h = w_1x + b_1$, and $o = w_2h + b_2$. Which of the functions can be represented by this network? For the options which can, write out appropriate values of w_1, w_2, b_1, b_2.



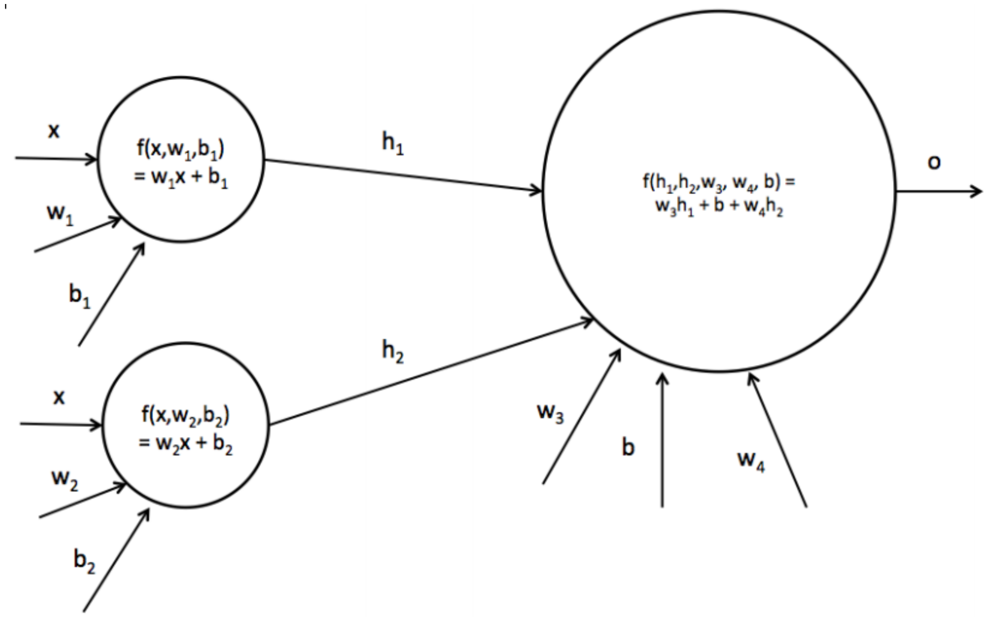
5. Next we add a ReLU non-linearity to the network after the first affine transformation, creating a hidden layer. Which of the functions can be represented by this network? For the options which can, write out appropriate values of w_1, w_2, b_1, b_2.



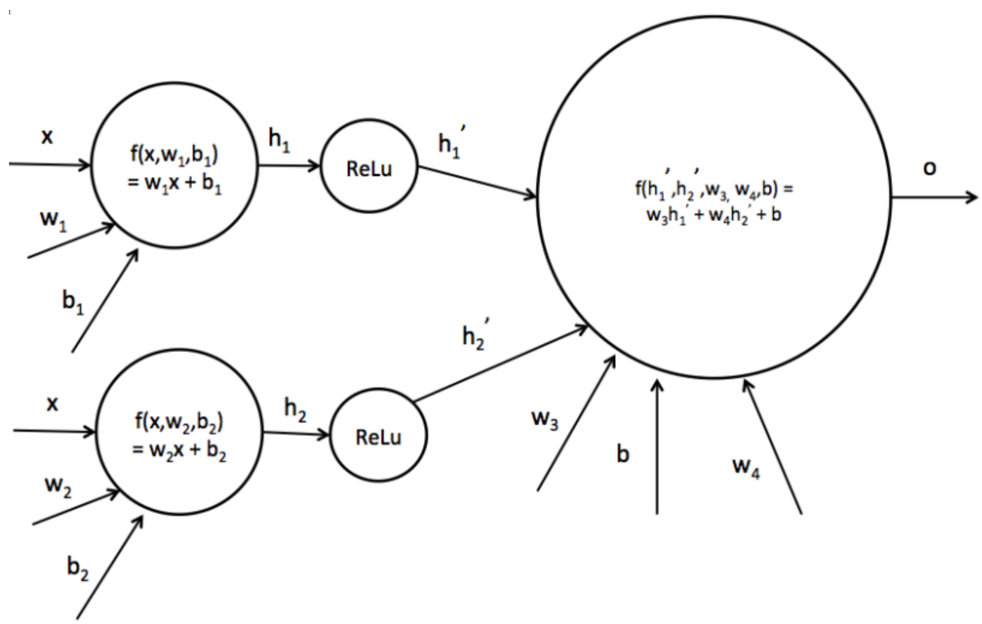
6. Now we add another hidden layer to the network, as indicated below. Which of the functions can be represented by this network?



7. We'd like to consider using a neural net with just one hidden layer, but have it be larger – a hidden layer of size 2. Let's first consider using just two affine functions, with no nonlinearity in between. Which of the functions can be represented by this network?



8. Now we'll add a non-linearity between the two affine layers, to produce the neural network below with a hidden layer of size 2. Which of the functions can be represented by this network?



CS188 Fall 2018 Review: Final Preparation

1 . Bounded suboptimal search: weighted A*

In this class you met A*, an algorithm for informed search guaranteed to return an optimal solution when given an admissible heuristic. Often in practical applications it is too expensive to find an optimal solution, so instead we search for good suboptimal solutions.

Weighted A* is a variant of A* commonly used for suboptimal search. Weighted A* is exactly the same as A* but where the f-value is computed differently:

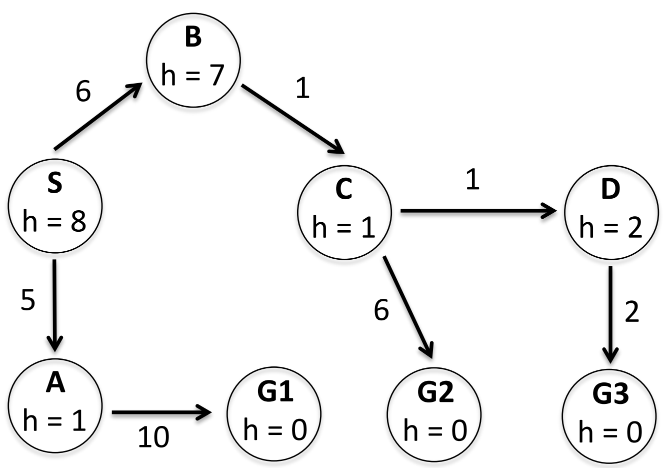
$$f(n) = g(n) + \varepsilon\, h(n)$$

where $\varepsilon \geq 1$ is a parameter given to the algorithm. In general, the larger the value of ε, the faster the search is, and the higher cost of the goal found.

Pseudocode for weighted A* tree search is given below. **NOTE:** The only differences from the A* tree search pseudocode presented in the lectures are: (1) $fringe$ is assumed to be initialized with the start node before this function is called (this will be important later), and (2) now INSERT takes ε as a parameter so it can compute the correct f-value of the node.

```
function WEIGHTED-A*-TREE-SEARCH(problem, fringe, ε)
    loop do
        if fringe is empty then return failure
        node ← REMOVE-FRONT(fringe)
        if GOAL-TEST(problem, STATE[node]) then return node
        for child-node in child-nodes do
            fringe ← INSERT(child-node, fringe, ε)
```

(a) We'll first examine how weighted A* works on the following graph:



Execute weighted A* on the above graph with $\varepsilon = 2$, completing the following table. To save time, you can optionally just write the nodes added to the fringe, with their g and f values.

node	Goal?	*fringe*
-	-	$\{S : g = 0, f = 16\}$
S	No	$\{S \to A : g = 5, f = 7;\ S \to B : g = 6, f = 20\}$

(b) After running weighted A* with weight $\varepsilon \geq 1$ a goal node G is found, of cost $g(G)$. Let C^* be the optimal solution cost, and suppose the heuristic is admissible. Select the strongest bound below that holds, and provide a proof.

○ $g(G) \leq \varepsilon\, C^*$ ○ $g(G) \leq C^* + \varepsilon$ ○ $g(G) \leq C^* + 2\varepsilon$ ○ $g(G) \leq 2^{\varepsilon}\, C^*$ ○ $g(G) \leq \varepsilon^2\, C^*$

Proof: (Partial credit for reasonable proof sketches.)

(c) Weighted A* includes a number of other algorithms as special cases. For each of the following, name the corresponding algorithm.

(i) $\varepsilon = 1$.

Algorithm:

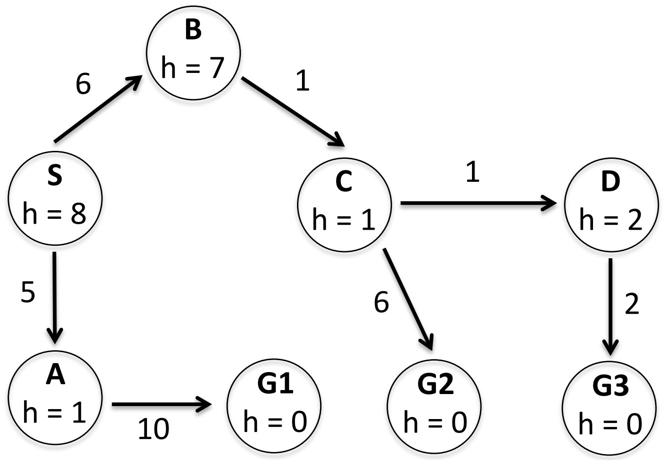
(ii) $\varepsilon = 0$.

Algorithm:

(iii) $\varepsilon \to \infty$ (i.e., as ε becomes arbitrarily large).

Algorithm:

(d) Here is the same graph again:



(i) Execute weighted A* on the above graph with $\varepsilon = 1$, completing the following table as in part (a):

node	Goal?	*fringe*

(ii) You'll notice that weighted A* with $\varepsilon = 1$ repeats computations performed when run with $\varepsilon = 2$. Is there a way to reuse the computations from the $\varepsilon = 2$ search by starting the $\varepsilon = 1$ search with a different fringe? Let F denote the set that consists of both (i) all nodes the fringe the $\varepsilon = 2$ search ended with, and (ii) the goal node G it selected. Give a brief justification for your answer.

◯ Use F as new starting fringe
◯ Use F with goal G removed as new starting fringe
◯ Use F as new starting fringe, updating the f-values to account for the new ε
◯ Use F with goal G removed as new starting fringe, updating the f-values to account for the new ε
◯ Initialize the new starting fringe to all nodes visited in previous search
◯ Initialize the new starting fringe to all nodes visited in previous search, updating the f-values to account for the new ε
◯ It is not possible to reuse computations, initialize the new starting fringe as usual

Justification:

2 . Crossword Puzzles as CSPs

You are developing a program to automatically solve crossword puzzles, because you think a good income source for you might be to submit them to the New York Times ($200 for a weekday puzzle, $1000 for a Sunday).[1] For those unfamiliar with crossword puzzles, a crossword puzzle is a game in which one is given a grid of squares that must be filled in with intersecting words going from left to right and top to bottom. There are a given set of starting positions for words (in the grid below, the positions $1, 2, 3, 4$, and 5), where words must be placed going across (left to right) or down (top to bottom). At any position where words intersect, the letters in the intersecting words must match. Further, no two words in the puzzle can be identical. An example is the grid below, in which the down words (1, 2, and 3) are `DEN`, `ARE`, and `MAT`, while the across words (1, 4, and 5) are `DAM`, `ERA`, and `NET`.

Example Crossword Grid and Solution

1 D	2 A	3 M
4 E	R	A
5 N	E	T

A part of your plan to make crosswords, you decide you will create a program that uses the CSP solving techniques you have learned in CS 188, since you want to make yourself obsolete at your own job from the get-go. Your first task is to choose the representation of your problem. You start with a dictionary of all the words you could put in the crossword puzzle, where the dictionary is of size K and consists of the words $\{d_1, d_2, \ldots, d_K\}$. Assume that you are given a grid with N empty squares and M different entries for words (and there are 26 letters in the English language). In the example above, $N = 9$ and $M = 6$ (three words across and three words down).

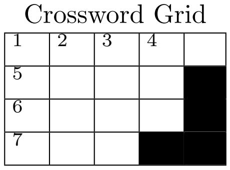
You initially decide to use words as the variables in your CSP. Let D_1 denote the first down word, D_2 the second, D_3 the third, etc., and similarly let A_k denote the kth across word. For example, in the crossword above, $A_1 = \texttt{DAM}$, $D_1 = \texttt{DEN}$, $D_2 = \texttt{ARE}$, and so on. Let $D_1[i]$ denote the letter in the ith position of the word D_1.

(a) What is the size of the state space for this CSP?

(b) Precisely (i.e. use mathematical notation to) describe the constraints of the CSP when we use words as variables.

After defining your CSP, you decide to go ahead and make a small crossword using the grid below. Assume that you use the words on the right as your dictionary.

Crossword Grid



Dictionary Words

`ARCS, BLAM, BEAR, BLOGS, LARD, LARP,`
`GAME, GAMUT, GRAMS, GPS, MDS, ORCS, WARBLER`

[1] `http://www.nytimes.com/2009/07/19/business/media/19askthetimes.html`

(c) Enforce all *unary* constraints by crossing out values in the table below.

D_1	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
D_2	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
D_3	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
D_4	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
A_1	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
A_5	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
A_6	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
A_7	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER

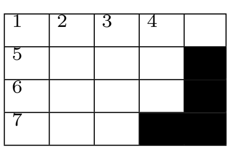
(d) Assume that in backtracking search, we assign A_1 to be GRAMS. Enforce unary constraints, and in addition, cross out all the values eliminated by forward checking against A_1 as a result of this assignment.

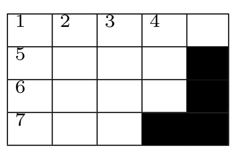
D_1	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
D_2	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
D_3	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
D_4	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
A_1	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
A_5	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
A_6	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
A_7	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER

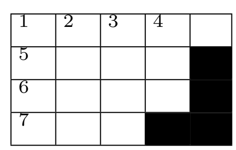
(e) Now let's consider how much arc consistency can prune the domains for this problem, even when no assignments have been made yet. I.e., assume no variables have been assigned yet, enforce unary constraints first, and then enforce arc consistency by crossing out values in the table below.

D_1	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
D_2	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
D_3	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
D_4	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
A_1	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
A_5	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
A_6	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER
A_7	ARCS	BLAM	BEAR	BLOGS	LARD	LARP	GPS	MDS	GAME	GAMUT	GRAMS	ORCS	WARBLER

(f) How many solutions to the crossword puzzle are there? Fill them (or the single solution if there is only one) in below.







Your friend suggests using letters as variables instead of words, thinking that sabotaging you will be funny. Starting from the top-left corner and going left-to-right then top-to-bottom, let X_1 be the first letter, X_2 be the second, X_3 the third, etc. In the very first example, $X_1 = \texttt{D}$, $X_2 = \texttt{A}$, and so on.

(g) What is the size of the state space for this formulation of the CSP?

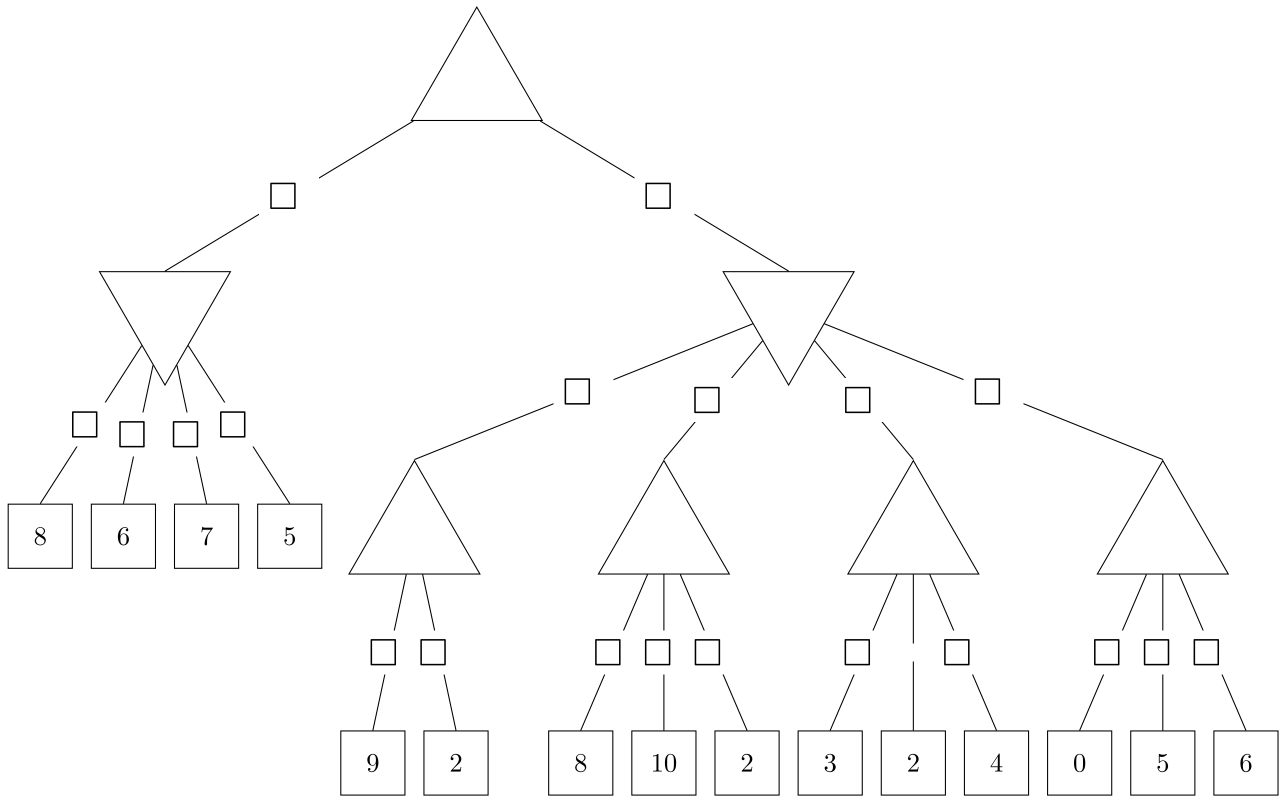
(h) Assume that in your implementation of backtracking search, you use the least constraining value heuristic. Assume that X_1 is the first variable you choose to instantiate. For the crossword puzzle used in parts (c)-(f), what letter(s) might your search assign to X_1?

3 . Game Trees

The following problems are to test your knowledge of Game Trees.

(a) Minimax

The first part is based upon the following tree. Upward triangle nodes are maximizer nodes and downward are minimizers. (small squares on edges will be used to mark pruned nodes in part (ii))



(i) Complete the game tree shown above by filling in values on the maximizer and minimizer nodes.

(ii) Indicate which nodes can be pruned by marking the edge above each node that can be pruned (you do not need to mark any edges below pruned nodes). In the case of ties, please prune any nodes that could not affect the root node's value. Fill in the bubble below if no nodes can be pruned.

○ No nodes can be pruned

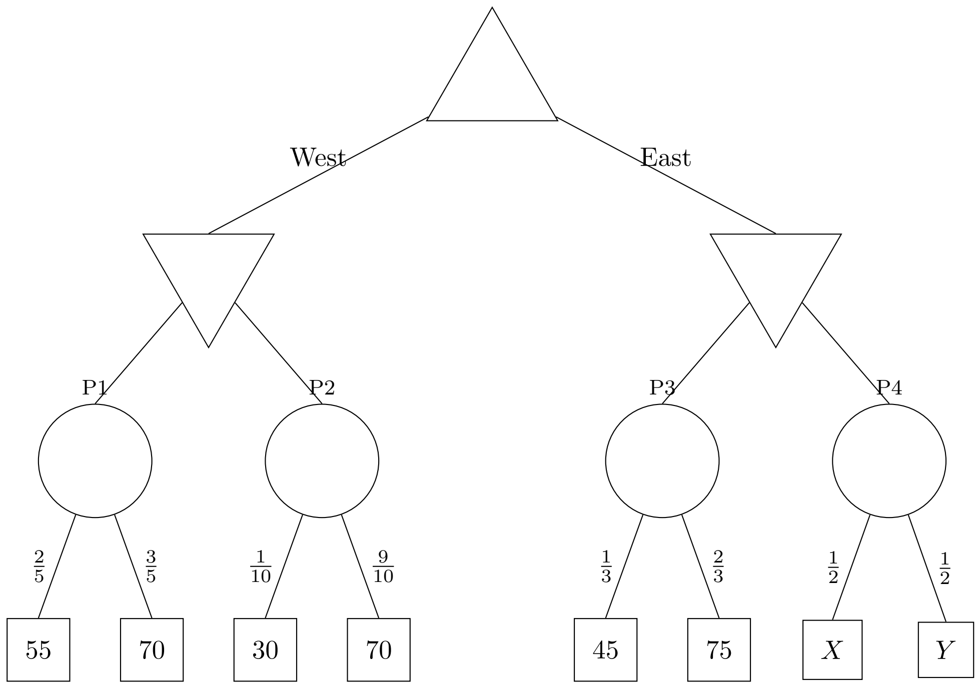
(b) Food Dimensions

The following questions are completely unrelated to the above parts.

Pacman is playing a tricky game. There are 4 portals to food dimensions. But, these portals are guarded by a ghost. Furthermore, neither Pacman nor the ghost know for sure how many pellets are behind each portal, though they know what options and probabilities there are for all but the last portal.

Pacman moves first, either moving West or East. After which, the ghost can block **1** of the portals available.

You have the following gametree. The maximizer node is Pacman. The minimizer nodes are ghosts and the portals are chance nodes with the probabilities indicated on the edges to the food. In the event of a tie, the left action is taken. Assume Pacman and the ghosts play optimally.



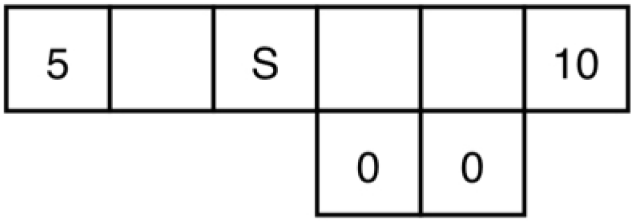
(i) Fill in values for the nodes that do not depend on X and Y.

(ii) What conditions must X and Y satisfy for Pacman to move East? What about to definitely reach the P4? Keep in mind that X and Y denote numbers of food pellets and must be **whole numbers**: $X, Y \in \{0, 1, 2, 3, \dots\}$.

To move East:

To reach P4:

4 . Discount MDPs

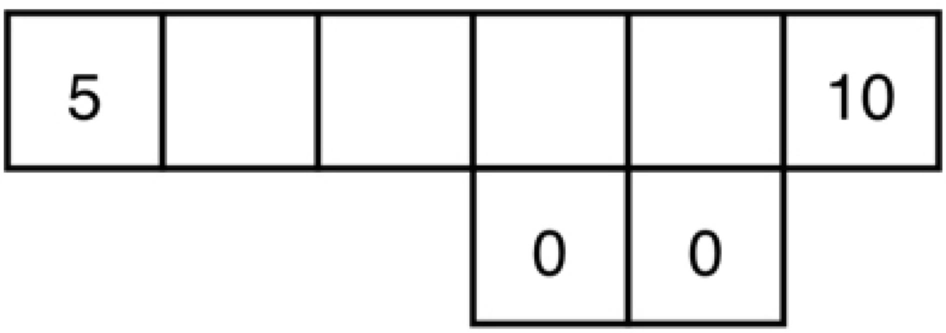


Consider the above gridworld. An agent is currently on grid cell S, and would like to collect the rewards that lie on both sides of it. If the agent is on a numbered square, its only available action is to Exit, and when it exits it gets reward equal to the number on the square. On any other (non-numbered) square, its available actions are to move East and West. Note that North and South are never available actions.

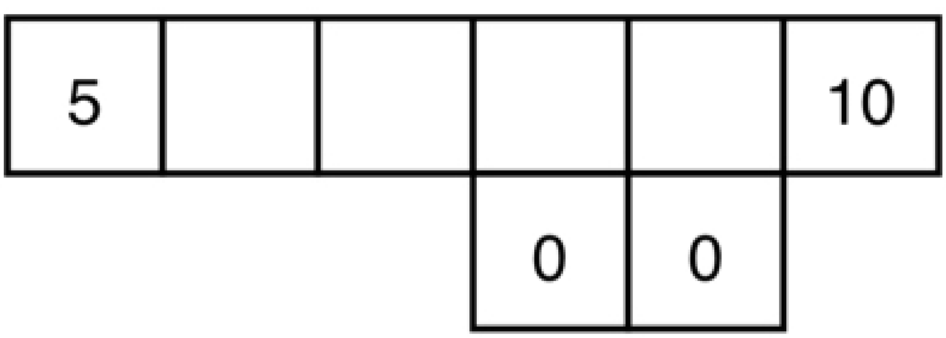
If the agent is in a square with an adjacent square downward, it does not always move successfully: when the agent is in one of these squares and takes a move action, it will only succeed with probability p. With probability $1-p$, the move action will fail and the agent will instead move downwards. If the agent is not in a square with an adjacent space below, it will always move successfully.

For parts (a) and (b), we are using discount factor $\gamma \in [0,1]$.

(a) Consider the policy π_{East}, which is to always move East (right) when possible, and to Exit when that is the only available action. For each non-numbered state x in the diagram below, fill in $V^{\pi_{\text{East}}}(x)$ in terms of γ and p.



(b) Consider the policy π_{West}, which is to always move West (left) when possible, and to Exit when that is the only available action. For each non-numbered state x in the diagram below, fill in $V^{\pi_{\text{West}}}(x)$ in terms of γ and p.



(c) For what range of values of p *in terms of* γ is it optimal for the agent to go West (left) from the start state (S)?

Range: ____________________

(d) For what range of values of p *in terms of* γ is π_{West} the optimal policy?

Range: ____________________

(e) For what range of values of p *in terms of* γ is π_{East} the optimal policy?

Range: ____________________

Recall that in approximate Q-learning, the Q-value is a weighted sum of features: $Q(s,a) = \sum_i w_i f_i(s,a)$. To derive a weight update equation, we first defined the loss function $L_2 = \frac{1}{2}(y - \sum_k w_k f_k(x))^2$ and found $dL_2/dw_m = -(y - \sum_k w_k f_k(x))f_m(x)$. Our label y in this set up is $r + \gamma \max_a Q(s', a')$. Putting this all together, we derived the gradient descent update rule for w_m as $w_m \leftarrow w_m + \alpha\,(r + \gamma \max_a Q(s', a') - Q(s,a))\, f_m(s,a)$.

In the following question, you will derive the gradient descent update rule for w_m using a different loss function:

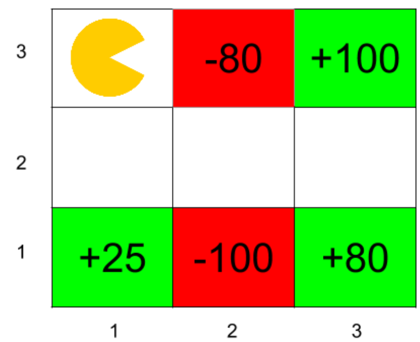
$$L_1 = \left| y - \sum_k w_k f_k(x) \right|$$

(f) Find dL_1/dw_m. Show work to have a chance at receiving partial credit. Ignore the non-differentiable point.

(g) Write the gradient descent update rule for w_m, using the L_1 loss function.

5 . Q-Learning Strikes Back

Consider the grid-world given below and Pacman who is trying to learn the optimal policy. If an action results in landing into one of the shaded states the corresponding reward is awarded during that transition. All shaded states are terminal states, i.e., the MDP terminates once arrived in a shaded state. The other states have the *North, East, South, West* actions available, which deterministically move Pacman to the corresponding neighboring state (or have Pacman stay in place if the action tries to move out of the grad). Assume the discount factor $\gamma = 0.5$ and the Q-learning rate $\alpha = 0.5$ for all calculations. Pacman starts in state (1, 3).



(a) What is the value of the optimal value function V^* at the following states:

$V^*(3,2) =$ ________________ $V^*(2,2) =$ ________________ $V^*(1,3) =$ ________________

(b) The agent starts from the top left corner and you are given the following episodes from runs of the agent through this grid-world. Each line in an Episode is a tuple containing (s, a, s', r).

Episode 1	Episode 2	Episode 3
(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0
(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0
(2,2), S, (2,1), -100	(2,2), E, (3,2), 0	(2,2), E, (3,2), 0
	(3,2), N, (3,3), +100	(3,2), S, (3,1), +80

Using Q-Learning updates, what are the following Q-values after the above three episodes:

Q((3,2),N) = ________________ Q((1,2),S) = ________________ $Q((2,2), E) =$ ________________

(c) Consider a feature based representation of the Q-value function:

$$Q_f(s,a) = w_1 f_1(s) + w_2 f_2(s) + w_3 f_3(a)$$

$f_1(s)$: The x coordinate of the state $\qquad$ $f_2(s)$: The y coordinate of the state

$$f_3(N) = 1,\ f_3(S) = 2,\ f_3(E) = 3,\ f_3(W) = 4$$

(i) Given that all w_i are initially 0, what are their values after the first episode:

$w_1 =$ ________________ $w_2 =$ ________________ $w_3 =$ ________________

(ii) Assume the weight vector w is equal to $(1, 1, 1)$. What is the action prescribed by the Q-function in state $(2, 2)$?

____

6 . Probability

(a) Consider the random variables A, B, and C. Circle all of the following equalities that are **always** true, if any.

1. $\mathbf{P}(A, B) = \mathbf{P}(A)\mathbf{P}(B) - \mathbf{P}(A|B)$

2. $\mathbf{P}(A, B) = \mathbf{P}(A)\mathbf{P}(B)$

3. $\mathbf{P}(A, B) = \mathbf{P}(A|B)\mathbf{P}(B) + \mathbf{P}(B|A)\mathbf{P}(A)$

4. $\mathbf{P}(A) = \sum_{b \in B} \mathbf{P}(A|B = b)\mathbf{P}(B = b)$

5. $\mathbf{P}(A, C) = \sum_{b \in B} \mathbf{P}(A|B = b)\mathbf{P}(C|B = b)\mathbf{P}(B = b)$

6. $\mathbf{P}(A, B, C) = \mathbf{P}(C|A)\mathbf{P}(B|C, A)\mathbf{P}(A)$

Now assume that A and B both can take on only the values true and false ($A \in \{\text{true}, \text{false}\}$ and $B \in \{\text{true}, \text{false}\}$). You are given the following quantities:

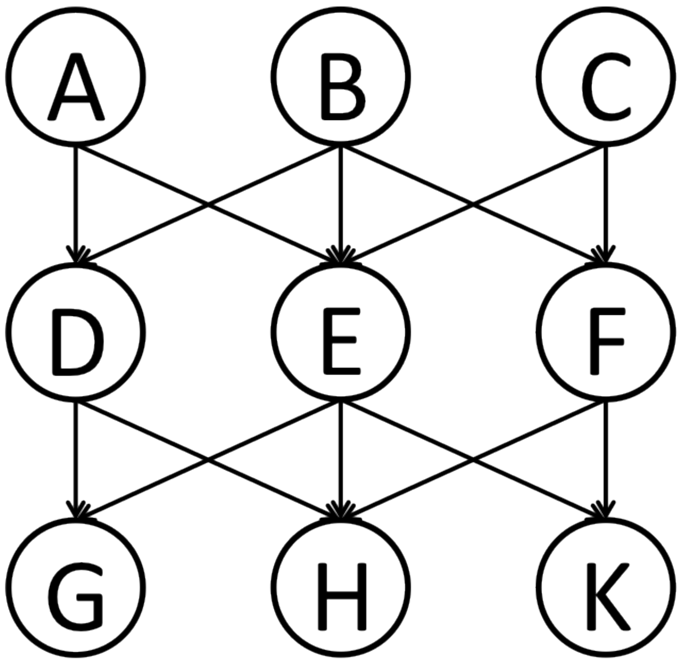
$$\begin{aligned} \mathbf{P}(A = \text{true}) &= \tfrac{1}{2} \\ \mathbf{P}(B = \text{true} \mid A = \text{true}) &= 1 \\ \mathbf{P}(B = \text{true}) &= \tfrac{3}{4} \end{aligned}$$

(b) What is $\mathbf{P}(B = \text{true} \mid A = \text{false})$?

7 . Bayes' Nets: Short Questions

(a) Bayes' Nets: Conditional Independence

Based only on the structure of the (new) Bayes' Net given below, circle whether the following conditional independence assertions are guaranteed to be true, guaranteed to be false, or cannot be determined by the structure alone.*Note: The ordering of the three answer columns might have been switched relative to previous exams!*



1	$A \perp\!\!\!\perp C$	Guaranteed false	Cannot be determined	Guaranteed true
2	$A \perp\!\!\!\perp C \mid E$	Guaranteed false	Cannot be determined	Guaranteed true
3	$A \perp\!\!\!\perp C \mid G$	Guaranteed false	Cannot be determined	Guaranteed true
4	$A \perp\!\!\!\perp K$	Guaranteed false	Cannot be determined	Guaranteed true
5	$A \perp\!\!\!\perp G \mid D, E, F$	Guaranteed false	Cannot be determined	Guaranteed true
6	$A \perp\!\!\!\perp B \mid D, E, F$	Guaranteed false	Cannot be determined	Guaranteed true
7	$A \perp\!\!\!\perp C \mid D, F, K$	Guaranteed false	Cannot be determined	Guaranteed true
8	$A \perp\!\!\!\perp G \mid D$	Guaranteed false	Cannot be determined	Guaranteed true

(b) Bayes' Nets: Elimination of a Single Variable

Assume we are running variable elimination, and we currently have the following three factors:

A	B	$f_1(A,B)$
$+a$	$+b$	0.1
$+a$	$-b$	0.5
$-a$	$+b$	0.2
$-a$	$-b$	0.5

A	C	D	$f_2(A,C,D)$
$+a$	$+c$	$+d$	0.2
$+a$	$+c$	$-d$	0.1
$+a$	$-c$	$+d$	0.5
$+a$	$-c$	$-d$	0.1
$-a$	$+c$	$+d$	0.5
$-a$	$+c$	$-d$	0.2
$-a$	$-c$	$+d$	0.5
$-a$	$-c$	$-d$	0.2

B	D	$f_3(B,D)$
$+b$	$+d$	0.2
$+b$	$-d$	0.2
$-b$	$+d$	0.5
$-b$	$-d$	0.1

The next step in the variable elimination is to eliminate B.

(i) Which factors will participate in the elimination process of B?

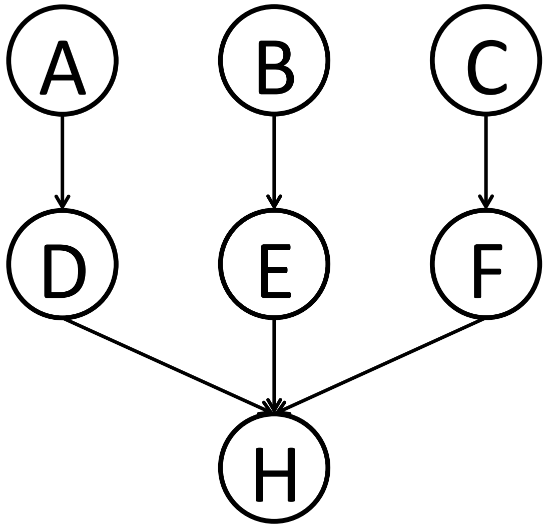
(ii) Perform the join over the factors that participate in the elimination of B. Your answer should be a table similar to the tables above, it is your job to figure out which variables participate and what the numerical entries are.

(iii) Perform the summation over B for the factor you obtained from the join. Your answer should be a table similar to the tables above, it is your job to figure out which variables participate and what the numerical entries are.

(c) Elimination Sequence

For the Bayes' net shown below, consider the query $P(A|H = +h)$, and the variable elimination ordering B, E, C, F, D.

(i) In the table below fill in the factor generated at each step — we did the first row for you.



Variable Eliminated	Factor Generated	Current Factors
(no variable eliminated yet)	(no factor generated)	$P(A), P(B), P(C), P(D|A), P(E|B), P(F|C), P(+h|D, E, F)$
B	$f_1(E)$	$P(A), P(C), P(D|A), P(F|C), P(+h|D, E, F), f_1(E)$
E		
C		
F		
D		

(ii) Which is the largest factor generated? Assuming all variables have binary-valued domains, how many entries does the corresponding table have?

(d) Sampling

(i) Consider the query $P(A| - b, -c)$. After rejection sampling we end up with the following four samples: $(+a, -b, -c, +d), (+a, -b, -c, -d), (+a, -b, -c, -d), (-a, -b, -c, -d)$. What is the resulting estimate of $P(+a| - b, -c)$?

.

(ii) Consider again the query $P(A| - b, -c)$. After likelihood weighting sampling we end up with the following four samples: $(+a, -b, -c, -d), (+a, -b, -c, -d), (-a, -b, -c, -d), (-a, -b, -c, +d)$, and respective weights: $0.1, 0.1, 0.3, 0.3$. What is the resulting estimate of $P(+a| - b, -c)$?

8 . HMM: Where is the key?

The cs188 staff have a key to the homework bin. It is the master key that unlocks the bins to many classes, so we take special care to protect it.

Every day John Duchi goes to the gym, and on the days he has the key, 60% of the time he forgets it next to the bench press. When that happens one of the other three GSIs, equally likely, always finds it since they work out right after. Jon Barron likes to hang out at Brewed Awakening and 50% of the time he is there with the key, he forgets the key at the coffee shop. Luckily Lubomir always shows up there and finds the key whenever Jon Barron forgets it. Lubomir has a hole in his pocket and ends up losing the key 80% of the time somewhere on Euclid street. However, Arjun takes the same path to Soda and always finds the key. Arjun has a 10% chance to lose the key somewhere in the AI lab next to the Willow Garage robot, but then Lubomir picks it up.

The GSIs lose the key at most once per day, around noon (after losing it they become extra careful for the rest of the day), and they always find it the same day in the early afternoon.

(a) Draw on the left the Markov chain capturing the location of the key and fill in the transition probability table on the right. In this table, the entry of row JD and column JD corresponds to $P(X_{t+1} = \text{JD}|X_t = \text{JD})$, the entry of row JD and column JB corresponds to $P(X_{t+1} = \text{JB}|X_t = \text{JD})$, and so forth.

	JD	JB	LB	AS
JD				
JB				
LB				
AS			0.10	

Monday early morning Prof. Abbeel handed the key to Jon Barron. (The initial state distribution assigns probability 1 to $X_0 = \text{JB}$ and probability 0 to all other states.)

(b) The homework is due Tuesday at midnight so the GSIs need the key to open the bin. What is the probability for each GSI to have the key at that time? Let X_0, X_{Mon} and X_{Tue} be random variables corresponding to who has the key when Prof. Abbeel hands it out, who has the key on Monday evening, and who has the key on Tuesday evening, respectively. Fill in the probabilities in the table below.

	$P(X_0)$	$P(X_{\text{Mon}})$	$P(X_{\text{Tue}})$
JD	0		
JB	1		
LB	0		
AS	0		

(c) The GSIs like their jobs so much that they decide to be professional GSIs permanently. They assign an extra credit homework (make computers truly understand natural language) due *at the end of time*. What is the probability that each GSI holds the key at a point infinitely far in the future. Hint:

$$P_\infty(x) = \sum_{x'} P(X_{\text{next day}} = x \mid X_{\text{current day}} = x')P_\infty(x')$$

Every evening the GSI who has the key feels obliged to write a short anonymous report on their opinion about the state of AI. Arjun and John Duchi are optimistic that we are right around the corner of solving AI and have an 80% chance of writing an optimistic report, while Lubomir and Jon Barron have an 80% chance of writing a pessimistic report. The following are the titles of the first few reports:

Monday:	Survey: Computers Become Progressively Less Intelligent (pessimistic)
Tuesday:	How to Solve Computer Vision in Three Days (optimistic)

(d) In light of that new information, what is the probability distribution for the key on Tuesday midnight given that Jon Barron has it Monday morning? You may leave the result as a ratio or unnormalized.

On Thursday afternoon Prof. Abbeel noticed a suspiciously familiar key on top of the Willow Garage robot's head. He thought to himself, "This can't possibly be the master key." (He was wrong!) Lubomir managed to snatch the key and distract him before he inquired more about it and is the key holder Thursday at midnight (i.e., $X_{\text{Thu}} = \text{LB}$). In addition, the Friday report is this:

Thursday:	??? (report unknown)
Friday:	AI is a scam. I know it, you know it, it is time for the world to know it! (pessimistic)

(e) Given that new information, what is the probability distribution for the holder of the key on Friday at midnight?

(f) Prof. Abbeel recalls that he saw Lubomir holding the same key on Tuesday night. Given this new information (in addition to the information in the previous part), what is the probability distribution for the holder of the key on Friday at midnight?

(g) Suppose in addition that we know that the titles of the reports for the rest of the week are:

Saturday:	Befriend your PC now. Soon your life will depend on its wishes (optimistic)
Sunday:	How we got tricked into studying AI and how to change field without raising suspicion (pessimistic)

Will that new information change our answer to (f)? Choose one of these options:

1. Yes, reports for Saturday and Sunday affect our prediction for the key holder on Friday.
2. No, our prediction for Friday depends only on what happened in the past.

9 . Ghostbusters

Suppose Pacman gets a noisy observation of a ghost's location for T moves, and then may guess where the ghost is at timestep T to eat it. To model the problem, you use an HMM, where the ith hidden state is the location of the ghost at timestep i and the ith evidence variable is the noisy observation of the ghost's location at time step i. *Assume Pacman always acts rationally.*

(a) If Pacman guesses correctly, he gets to eat the ghost resulting in a utility of 20. Otherwise he gets a utility of 0. If he does not make any guess, he gets a utility of 0.

Which of the following algorithms could Pacman use to determine the ghost's most likely location at time T? (Don't worry about runtime.)

- ☐ Viterbi
- ☐ Forward algorithm for HMMs
- ☐ Particle filtering with a lot of particles
- ☐ Variable elimination on the Bayes Net representing the HMM
- ☐ None of the above, Pacman should use ________

(b) In the previous part, there was no penalty for guessing. Now, Pacman has to *pay* 10 *utility* in order to try to eat the ghost. Once he pays, he still gets 20 utility for correctly guessing and eating the ghost, and 0 utility for an incorrect guess. Pacman determines that the most likely ghost location at time T is (x, y), and the probability of that location is p.

What is the expected utility of guessing that the ghost is at (x, y), as a function of p? ________

When should Pacman guess that the ghost is at (x, y)?

- ○ Never (he should not guess)
- ○ If $p <$ ________.
- ○ If $p >$ ________.
- ○ Always

(c) Now, in addition to the -10 utility for trying to eat the ghost, Pacman can also pay 5 utility to learn the exact location of the ghost. (So, if Pacman pays the 5 utility and eats the ghost, he pays 15 utility and gains 20 utility for a total of 5 utility.)

When should Pacman pay the 5 utility to find the exact ghost location?

- ○ Never
- ○ If $p <$ ________.
- ○ If $p >$ ________.
- ○ Always

(d) Now, Pacman can try to eat one out of Blinky (B), Inky (I) and Clyde (C) (three of the ghosts). He has some preferences about which one to eat, but he's afraid that his preferences are not rational. Help him out by showing him a utility function that matches his listed preferences, or mark "Not possible" if no rational utility function will work. You may choose any real number for each utility value. **If "Not possible" is marked, we will ignore any written utility function.**

(i) The preferences are $B \prec I$ and $I \prec C$ and $[0.5, B; 0.5, C] \prec I$

$U(B)$	$U(I)$	$U(C)$

○ Not possible

(ii) The preferences are $I \prec B$ and $[0.5, B; 0.5, C] \prec C$ and $[0.5, B; 0.5, C] \prec [0.5, B; 0.5, I]$

$U(B)$	$U(I)$	$U(C)$

○ Not possible

10 . Perceptrons

(a) Consider a multi-class perceptron for classes A, B, and C with current weight vectors:

$w_A = (1, -4, 7)$, $w_B = (2, -3, 6)$, $w_C = (7, 9, -2)$

A new training sample is now considered, which has feature vector $f(x) = (-2, 1, 3)$ and label $y^* = B$. What are the resulting weight vectors after the perceptron has seen this example and updated the weights?

$w_A =$ ________________ $w_B =$ ________________ $w_C =$ ________________

(b) A single perceptron can compute the XOR function.

○ True ○ False

(c) A perceptron is guaranteed to learn a separating decision boundary for a separable dataset within a finite number of training steps.

○ True ○ False

(d) Given a linearly separable dataset, the perceptron algorithm is guaranteed to find a max-margin separating hyperplane.

○ True ○ False

(e) You would like to train a neural network to classify digits. Your network takes as input an image and outputs probabilities for each of the 10 classes, 0-9. The network's prediction is the class that it assigns the highest probability to. From the following functions, select all that would be suitable loss functions to minimize using gradient descent:

- ☐ The square of the difference between the correct digit and the digit predicted by your network
- ☐ The probability of the correct digit under your network
- ☐ The negative log-probability of the correct digit under your network
- ○ None of the above

11 . Naive Bayes: Pacman or Ghost?

You are standing by an exit as either Pacmen or ghosts come out of it. Every time someone comes out, you get two observations: a visual one and an auditory one, denoted by the random variables X_v and X_a, respectively. The visual observation informs you that the individual is either a Pacman ($X_v = 1$) or a ghost ($X_v = 0$). The auditory observation X_a is defined analogously. Your observations are a noisy measurement of the individual's true type, which is denoted by Y. After the indiviual comes out, you find out what they really are: either a Pacman ($Y = 1$) or a ghost ($Y = 0$). You have logged your observations and the true types of the first 20 individuals:

individual i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
first observation $X_v^{(i)}$	0	0	1	0	1	0	0	1	1	1	0	1	1	0	1	1	1	0	0	0
second observation $X_a^{(i)}$	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
individual's type $Y^{(i)}$	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0

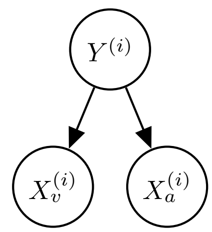
The superscript (i) denotes that the datum is the ith one. Now, the individual with $i = 20$ comes out, and you want to predict the individual's type $Y^{(20)}$ given that you observed $X_v^{(20)} = 1$ and $X_a^{(20)} = 1$.

(a) Assume that the types are independent, and that the observations are independent conditioned on the type. You can model this using naïve Bayes, with $X_v^{(i)}$ and $X_a^{(i)}$ as the features and $Y^{(i)}$ as the labels. Assume the probability distributions take on the following form:

$$P(X_v^{(i)} = x_v | Y^{(i)} = y) = \begin{cases} p_v & \text{if } x_v = y \\ 1 - p_v & \text{if } x_v \neq y \end{cases}$$

$$P(X_a^{(i)} = x_a | Y^{(i)} = y) = \begin{cases} p_a & \text{if } x_a = y \\ 1 - p_a & \text{if } x_a \neq y \end{cases}$$

$$P(Y^{(i)} = 1) = q$$



for $p_v, p_a, q \in [0, 1]$ and $i \in \mathbb{N}$.

(i) What's the maximum likelihood estimate of p_v, p_a and q?

$p_v =$ ________________ $p_a =$ ________________ $q =$ ________________

(ii) What is the probability that the next individual is Pacman given your observations? Express your answer in terms of the parameters p_v, p_a and q (you might not need all of them).

$P(Y^{(20)} = 1 | X_v^{(20)} = 1, X_a^{(20)} = 1) =$ ________

Now, assume that you are given additional information: you are told that the individuals are actually coming out of a bus that just arrived, and each bus carries *exactly* 9 individuals. Unlike before, the types of every 9 consecutive individuals are *conditionally* independent given the bus type, which is denoted by Z. Only after all of the 9 individuals have walked out, you find out the bus type: one that carries mostly Pacmans ($Z = 1$) or one that carries mostly ghosts ($Z = 0$). Thus, you only know the bus type in which the first 18 individuals came in:

individual i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
first observation $X_v^{(i)}$	0	0	1	0	1	0	0	1	1	1	0	1	1	0	1	1	1	0	0	0
second observation $X_a^{(i)}$	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
individual's type $Y^{(i)}$	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0

bus j	0	1	
bus type $Z^{(j)}$	0	1	

(b) You can model this using a variant of naïve bayes, where now 9 consecutive labels $Y^{(i)}, \dots, Y^{(i+8)}$ are *conditionally* independent given the bus type $Z^{(j)}$, for bus j and individual $i = 9j$. Assume the probability distributions take on the following form:

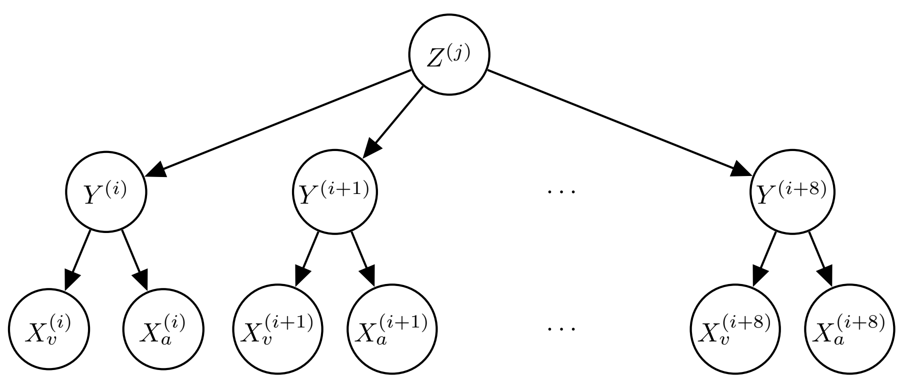
$$P(X_v^{(i)} = x_v | Y^{(i)} = y) = \begin{cases} p_v & \text{if } x_v = y \\ 1 - p_v & \text{if } x_v \neq y \end{cases}$$

$$P(X_a^{(i)} = x_a | Y^{(i)} = y) = \begin{cases} p_a & \text{if } x_a = y \\ 1 - p_a & \text{if } x_a \neq y \end{cases}$$

$$P(Y^{(i)} = 1 | Z^{(j)} = z) = \begin{cases} q_0 & \text{if } z = 0 \\ q_1 & \text{if } z = 1 \end{cases}$$

$$P(Z^{(j)} = 1) = r$$

for $p, q_0, q_1, r \in [0, 1]$ and $i, j \in \mathbb{N}$.



(i) What's the maximum likelihood estimate of q_0, q_1 and r?

$q_0 =$ ________________ $q_1 =$ ________________ $r =$ ________________

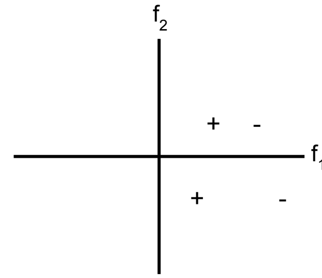
(ii) Compute the following joint probability. Simplify your answer as much as possible and express it in terms of the parameters p_v, p_a, q_0, q_1 and r (you might not need all of them).

$P(Y^{(20)} = 1, X_v^{(20)} = 1, X_a^{(20)} = 1, Y^{(19)} = 1, Y^{(18)} = 1) =$ ________________________

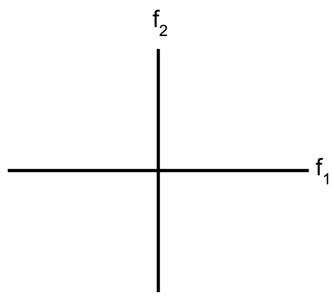
12 . Decision Trees and Other Classifiers

(a) Suppose you have a small training data set of four points in *distinct* locations, two from the "+" class and two from the "–" class. For each of the following conditions, draw a particular training data set (**of exactly four points**: +, +, −, and −) that satisfy the conditions. If this is impossible, mark "Not possible". *If "Not possible" is marked, we will ignore any data points.*

For example, if the conditions were "A depth-1 decision tree can perfectly classify the training data points," an acceptable answer would be the data points to the right.

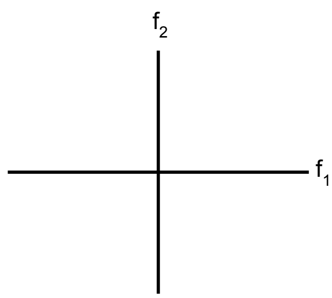


(i) A linear perceptron with a bias term can perfectly classify the training data points, but a linear perceptron without a bias term cannot.



○ Not possible

(ii) A depth-2 decision tree cannot classify the training data perfectly



○ Not possible

(b) You are still trying to classify between "+" and "-", but your two features now can take on only three possible values, $\{-1, 0, 1\}$. You would like to use a Naive Bayes model with the following CPTs:

X	$P(X)$
-	0.4
+	0.6

X	F_1	$P(F_1 \mid X)$
-	-1	0.4
-	0	0.5
-	1	0.1
+	-1	0.7
+	0	0.1
+	1	0.2

X	F_2	$P(F_2 \mid X)$
-	-1	0.1
-	0	0.1
-	1	0.8
+	-1	0.6
+	0	0.1
+	1	0.3

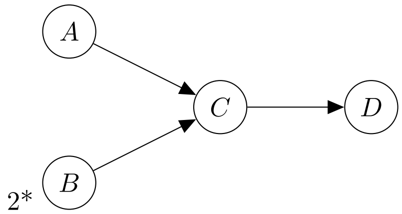
(i) If you observe that $F_1 = -1$ and $F_2 = -1$, how will you classify X using Naive Bayes?
○ $X = -$ ○ $X = +$

(ii) If you observe that $F_1 = 0$ and $F_2 = 0$, how will you classify X using Naive Bayes?
○ $X = -$ ○ $X = +$

(iii) If you observe that $F_1 = 1$ and $F_2 = 1$, how will you classify X using Naive Bayes?
○ $X = -$ ○ $X = +$

13 . Bayes' Net Sampling

Assume you are given the following Bayes' net and the corresponding distributions over the variables in the Bayes' net.



$P(A)$	
+a	0.1
-a	0.9

$P(B)$	
+b	.7
-b	.3

$P(C\|A,B)$			
+c	+a	+b	.25
-c	+a	+b	.75
+c	-a	+b	.6
-c	-a	+b	.4
+c	+a	-b	.5
-c	+a	-b	.5
+c	-a	-b	.2
-c	-a	-b	.8

$P(D\|C)$		
+d	+c	.5
-d	+c	.5
+d	-c	.8
-d	-c	.2

(a) Assume we receive evidence that $A = +a$. If we were to draw samples using rejection sampling, on expectation what percentage of the samples will be **rejected**?

(b) Next, assume we observed both $A = +a$ and $D = +d$. What are the weights for the following samples under likelihood weighting sampling?

Sample	Weight
$(+a, -b, +c, +d)$	
$(+a, -b, -c, +d)$	
$(+a, +b, -c, +d)$	

(c) Given the samples in the previous question, estimate $P(-b| + a, +d)$.

(d) Assume we need to (approximately) answer two different inference queries for this graph: $P(C| + a)$ and $P(C| + d)$. You are required to answer one query using likelihood weighting and one query using Gibbs sampling. In each case you can only collect a relatively small amount of samples, so for maximal accuracy you need to make sure you cleverly assign algorithm to query based on how well the algorithm fits the query. Which query would you answer with each algorithm?

Algorithm	Query
Likelihood Weighting	

Algorithm	Query
Gibbs Sampling	

Justify your answer: