

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:

$$f(n) = g(n) + \epsilon h(n)$$

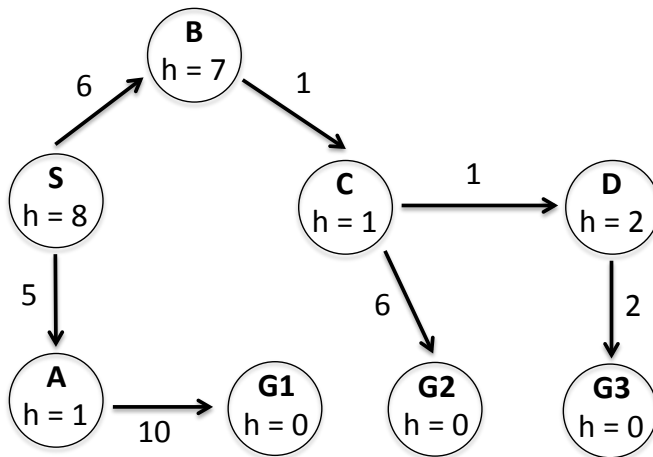
where $\epsilon \geq 1$ is a parameter given to the algorithm. In general, the larger the value of ϵ , the faster the search is, and the higher cost of the goal found.

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.

```

1: function WEIGHTED-A*-TREE-SEARCH(problem, fringe,  $\epsilon$ )
2:   loop do
3:     if fringe is empty then return failure
4:     node  $\leftarrow$  REMOVE-FRONT(fringe)
5:     if GOAL-TEST(problem, STATE[node]) then return node
6:     for child-node in child-nodes do
7:       fringe  $\leftarrow$  INSERT(child-node, fringe,  $\epsilon$ )
    
```

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



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

<i>node</i>	Goal?	<i>fringe</i>
-	-	$\{S : g = 0, f = 16\}$
<i>S</i>	No	$\{S \rightarrow A : g = 5, f = 7; S \rightarrow B : g = 6, f = 20\}$

(b) After running weighted A* with weight $\epsilon \geq 1$ 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.

- $g(G) \leq \epsilon C^*$
 $g(G) \leq C^* + \epsilon$
 $g(G) \leq C^* + 2\epsilon$
 $g(G) \leq 2^\epsilon C^*$
 $g(G) \leq \epsilon^2 C^*$

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) $\epsilon = 1$.

Algorithm:

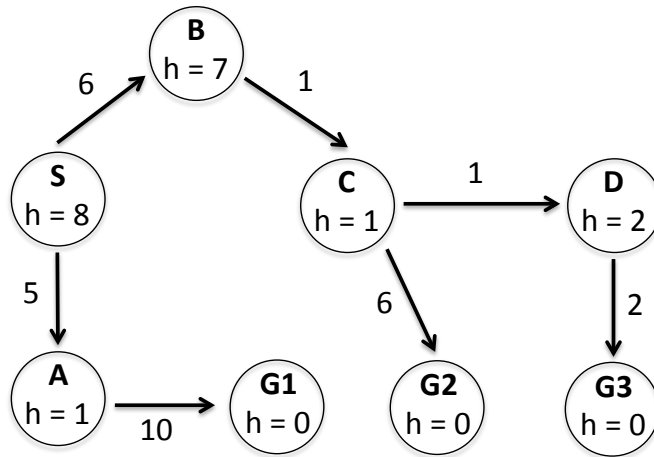
(ii) $\epsilon = 0$.

Algorithm:

(iii) $\epsilon \rightarrow \infty$ (i.e., as ϵ becomes arbitrarily large).

Algorithm:

(d) Here is the same graph again:



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

<i>node</i>	Goal?	<i>fringe</i>

(ii) You'll notice that weighted A* with $\epsilon = 1$ repeats computations performed when run with $\epsilon = 2$. Is there a way to reuse the computations from the $\epsilon = 2$ search by starting the $\epsilon = 1$ search with a different fringe? Let F denote the set that consists of both (i) all nodes the fringe the $\epsilon = 2$ 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 $\{d_1, d_2, \dots, d_K\}$. 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 k th across word. For example, in the crossword above, $A_1 = DAM$, $D_1 = DEN$, $D_2 = ARE$, and so on. Let $D_1[i]$ denote the letter in the i th 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

1	2	3	4	
5				
6				
7				

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.

1	2	3	4	
5				
6				
7				

1	2	3	4	
5				
6				
7				

1	2	3	4	
5				
6				
7				

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, $X_1 = D$, $X_2 = A$, and so on.

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

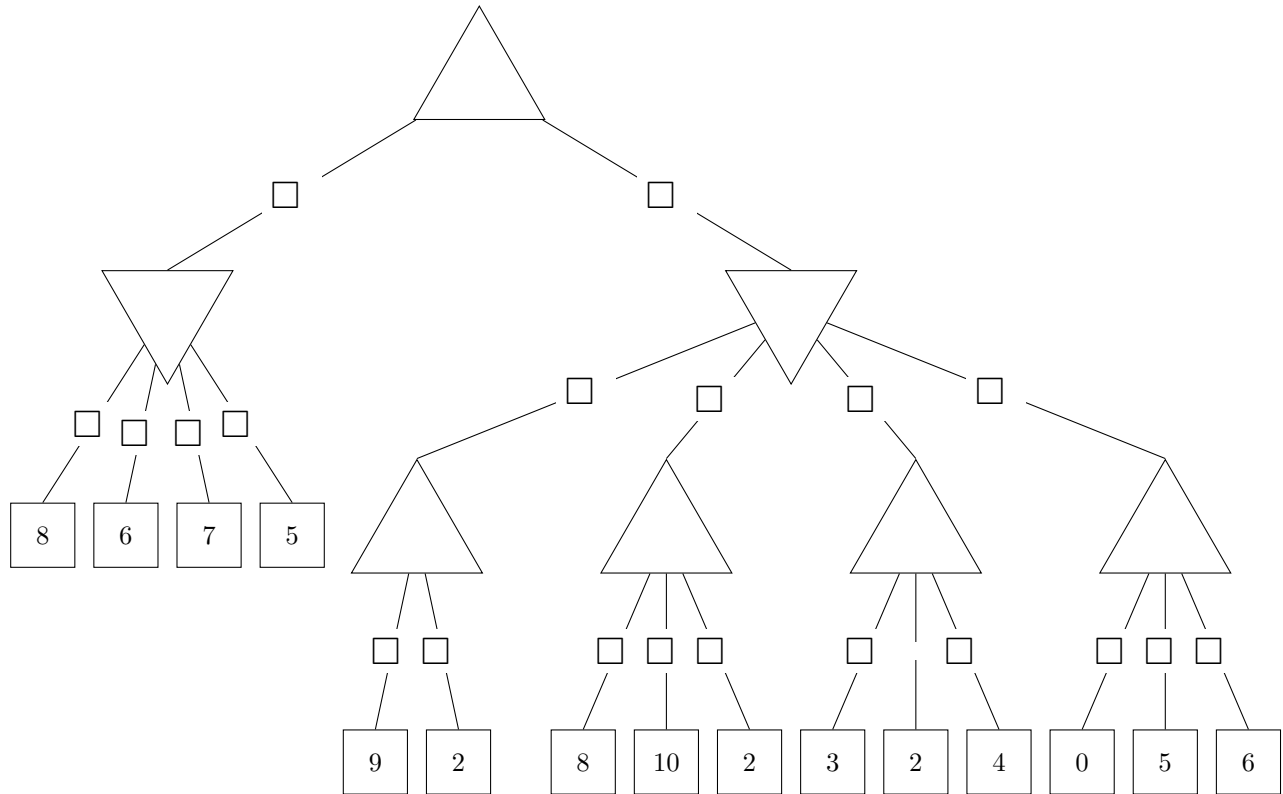
(h) Assume that in your implementation of backtracking search, you use the least constraining value heuristic. Assume that X_1 is the first variable you choose to instantiate. For the crossword puzzle used in parts (c)-(f), what letter(s) might your search assign to X_1 ?

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.

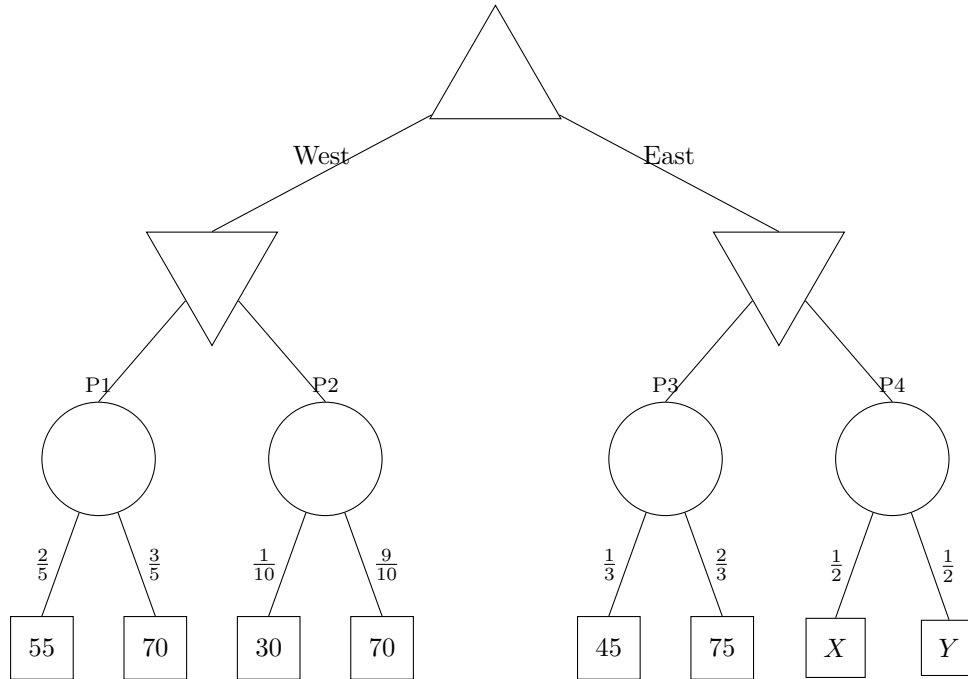
No nodes can be pruned

(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.

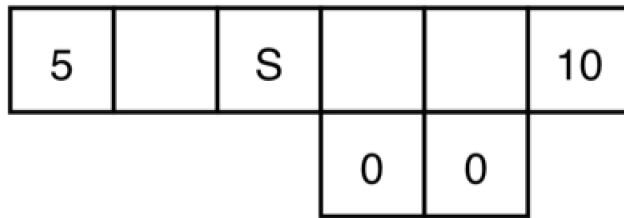


- (i) Fill in values for the nodes that do not depend on X and Y .
- (ii) What conditions must X and Y satisfy for Pacman to move East? What about to definitely reach the P4? Keep in mind that X and Y denote numbers of food pellets and must be **whole numbers**: $X, Y \in \{0, 1, 2, 3, \dots\}$.

To move East:

To reach P4:

4 . Discount MDPs

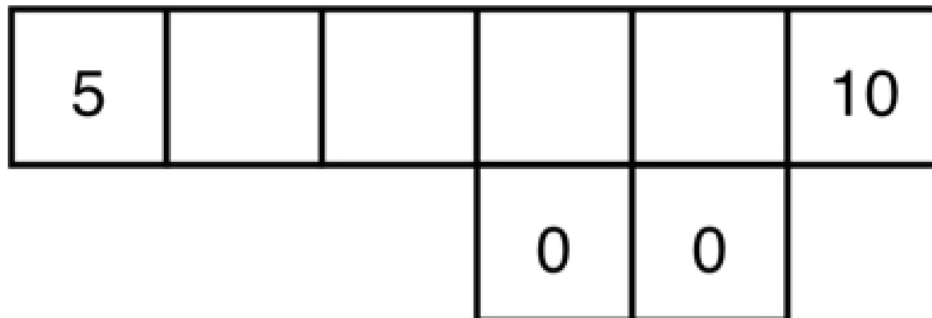


Consider the above gridworld. An agent is currently on grid cell S , and would like to collect the rewards that lie on both sides of it. If the agent is on a numbered square, its only available action is to Exit, and when it exits it gets reward equal to the number on the square. On any other (non-numbered)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

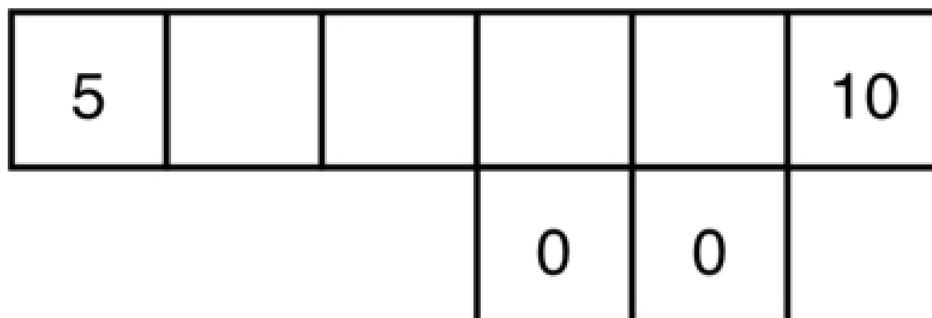
If the agent is in a square with an adjacent square downward, it does not always move successfully: when the agent is in one of these squares and takes a move action, it will only succeed with probability p . With probability $1 - p$, the move action will fail and the agent will instead move downwards. 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 grid). 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 $f_2(s)$: 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:

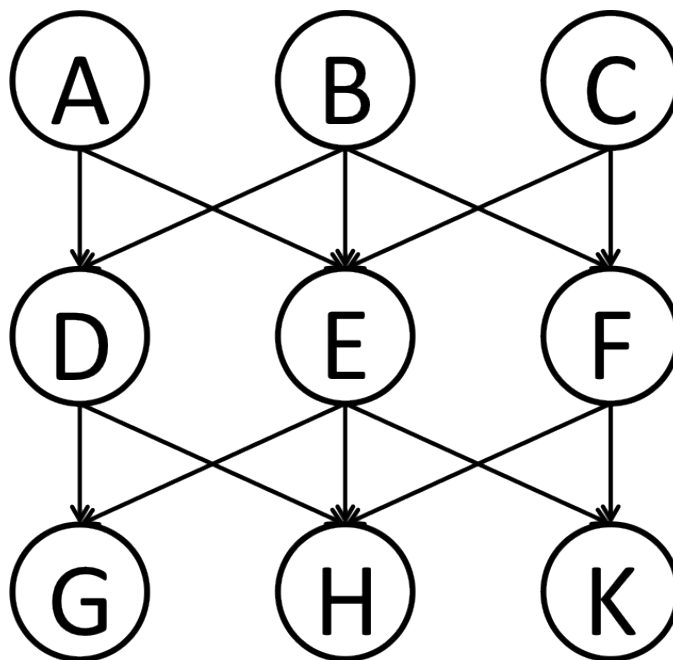
$$\begin{aligned} \mathbf{P}(A = \text{true}) &= \frac{1}{2} \\ \mathbf{P}(B = \text{true} \mid A = \text{true}) &= \frac{1}{4} \\ \mathbf{P}(B = \text{true}) &= \frac{3}{4} \end{aligned}$$

(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

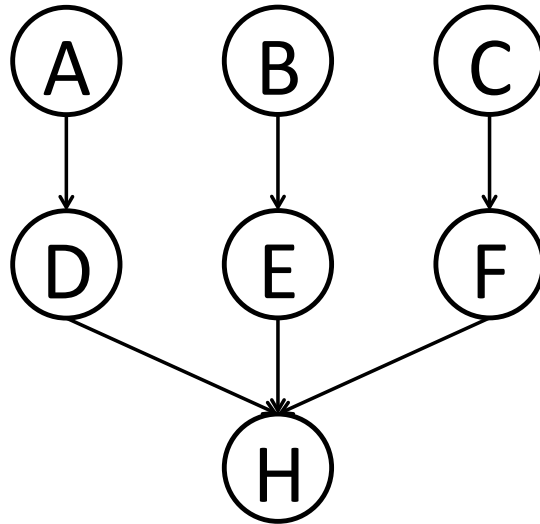
The next step in the variable elimination is to eliminate B .

- (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 i th hidden state is the location of the ghost at timestep i and the i th 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 $p < \underline{\hspace{2cm}}$.
- If $p > \underline{\hspace{2cm}}$.
- 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 $p < \underline{\hspace{2cm}}$.
- If $p > \underline{\hspace{2cm}}$.
- 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?

$$w_A = \underline{\hspace{2cm}} \qquad w_B = \underline{\hspace{2cm}} \qquad w_C = \underline{\hspace{2cm}}$$

(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 individual 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

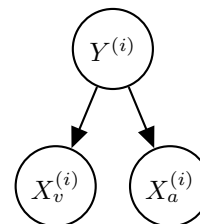
The superscript (i) denotes that the datum is the i th one. Now, the individual with $i = 20$ comes out, and you want to predict the individual's type $Y^{(20)}$ given that you observed $X_v^{(20)} = 1$ and $X_a^{(20)} = 1$.

- (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:

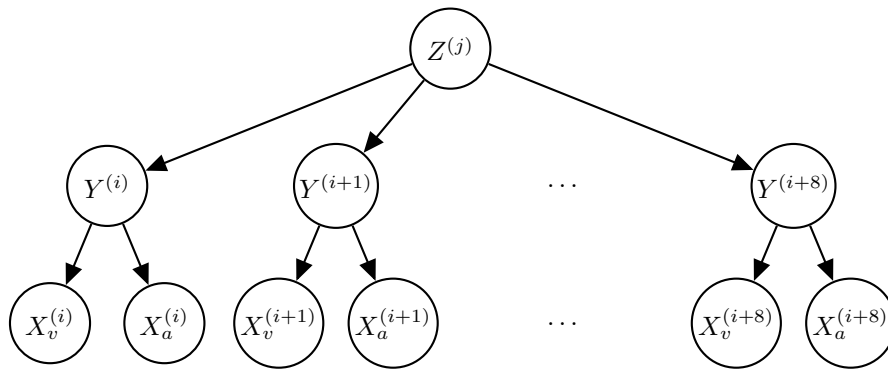
$$P(X_v^{(i)} = x_v | Y^{(i)} = y) = \begin{cases} p_v & \text{if } x_v = y \\ 1 - p_v & \text{if } x_v \neq y \end{cases}$$

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

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

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

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 ?

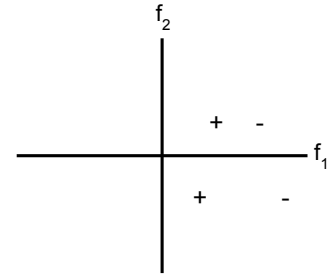
$q_0 =$ _____ $q_1 =$ _____ $r =$ _____

- (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).

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

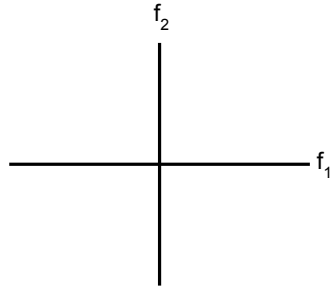
12 . Decision Trees and Other Classifiers

- (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.



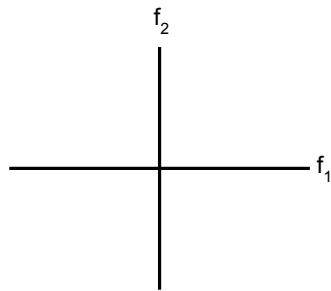
For example, if the conditions were “A depth-1 decision tree can perfectly classify the training data points,” an acceptable answer would be the data points to the right.

- (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

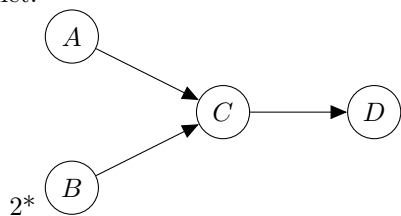
- (b) You are still trying to classify between “+” and “-”, but your two features now can take on only three possible values, $\{-1, 0, 1\}$. You would like to use a Naive Bayes model with the following CPTs:

X	$P(X)$	X	F_1	$P(F_1 X)$	X	F_2	$P(F_2 X)$
-	0.4	-	-1	0.4	-	-1	0.1
+	0.6	-	0	0.5	-	0	0.1
		-	1	0.1	-	1	0.8
		+	-1	0.7	+	-1	0.6
		+	0	0.1	+	0	0.1
		+	1	0.2	+	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: