1. (12 points) Linear Naive Bayes

Recall that a Naive Bayes classifier with observed random variables F_i $(i = 1, \ldots, n)$ and the query variable Y uses the classification rule:

$$\arg\max_y P(y|f_1, \ldots, f_n) = \arg\max_y P(y) \prod_{i=1}^{n} P(f_i|y)$$

And a linear classifier (for example, perceptron) uses the classification rule:

$$\arg\max_y \sum_{i=0}^{n} w_{y,i} \cdot f_i \qquad \text{where } f_0 = 1 \text{ is a bias feature}$$

(a) (8 pt) Consider a Naive Bayes classifier with binary-valued features, i.e. $f_i \in [0, 1]$. Prove that it is also a linear classifier, by defining weights $w_{y,i}$ (for $i = 0, \ldots, n$) such that both decision rules above are equivalent. The weights should be expressed in terms of the Naive Bayes probabilities — $P(y)$ and $P(f_i|y)$. You may assume that all the Naive Bayes probabilities are non-zero.

(b) (4 pt) For the training set below with binary features F_1 and F_2 and label Y, either name a smoothing method that would estimate a naive Bayes model that would correctly classify all training set data, or state that it is impossible (i.e., there is no smoothing method that would give appropriate probabilities).

F_1	F_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

2. (15 points) Blind Connect Three

In Connect Three, players alternate dropping pieces into one of four columns. A player wins by having three consecutive pieces of their color either horizontally, vertically, or diagonally. Assume columns have infinite height. A dropped piece *always* occupies the lowest open space in that column.

You are playing the game blind-folded against a random opponent. You can't see the opponent's moves. However, you can *always* hear the opponent's piece sliding into place. When the opponent drops his piece along the edge of the board, it makes a *zing* sound; and when he drops it in one of the center two columns, it makes a *zang* sound. On the other hand, you know exactly the move which you have made. When a player wins, the referee stops the game and announces the winner.

We'll assume that you are representing your belief with a particle filter where each particle is a complete description of the board. Also, assume that you are the first player and that you are playing white. The only observations you get are the *zing* and the *zang* of the opponent pieces falling down.

(a) (8 pt) For each of the following parts, answer *True* if its possible that the two particles are both present together in your particle filter immediately after a resampling step. If you answer *False*, *you must justify your answer for full credit.*

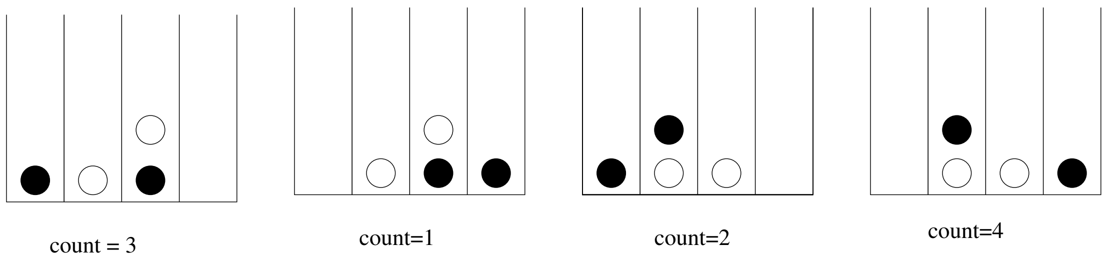
i. (*true* or *false*)

ii. (*true* or *false*)

iii. (*true* or *false*)

iv. (*true* or *false*)

For the next three sub-parts assume that you have exactly 10 particles in your particle filter, and after two rounds of moves you have the following four particles and counts of particles as shown below. Also, your utility is 1 if you win on the next move, and 0 othewise.



(b) (1 pt) What is your belief that column 1 is empty?

(c) (2 pt) What is your MEU, and which move(s) gives you this expected utility?

(d) (4 pt) What is the value of peeking through the blind-fold?

3. (15 points) MDP: Walk or Jump?

Consider the following MDP. It has states $\{0, 1, 2, 3, 4\}$ with 4 the starting state. In every state there are two actions: walk (W) and jump (J). The Walk action decreases the state by one. The jump action has probability 1/2 of decreasing the state by two, and probability 1/2 of leaving the state unchanged. Actions will not decrease the state below zero: you will remain in state 0 no matter which action you take, and Jumping from state 1 leads to state 0 with probability 1/2 and state 1 with probability 1/2. The reward gained when taking an action is the distance travelled squared: $R(s, a, s') = (s - s')^2$. The discount ratio is $\gamma = 1/2$.

(a) (2 pt) Draw the transition graph for this MDP.

(b) (3 pt) Compute $V(2)$. What is the optimal action to take in state 2?

(c) (3 pt) Give an expression for $V(4)$ in terms of $V(3)$ and $V(2)$.

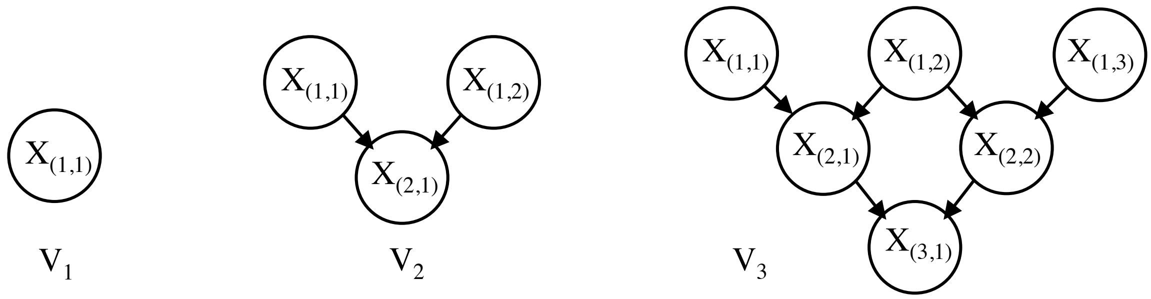
Suppose that the MDP was connected in a ring: Walking from state 0 will take you to state 4, Jumping from 1 will take you to either 4 or leave you in 1, and Jumping from 0 will take you to either 3 or leave you in 0.

(d) (4 pt) Find $V^*(2)$.

(e) (3 pt) Determine the optimal policy for this MDP.

4. (20 points) Very Big V Structures

Let V_n be a Bayes net with nodes $X_{(i,j)}$ for all $i + j \leq n + 1$, where both $i \geq 1$ and $j \geq 1$, and where the parents of $X_{(i,j)}$ are $X_{(i-1,j)}$ and $X_{(i-1,j+1)}$. Nodes $X_{(1,j)}$ have no parents. V_1, V_2, and V_3 are shown below.



(a) (3 pt) For any V_n, give general conditions in terms of i, j, k, and ℓ that make $X_{(i,j)}$ independent of $X_{(k,\ell)}$ (i.e., $X_{(i,j)} \perp\!\!\!\perp X_{(k,\ell)}$), assuming $k > i$.

(b) (3 pt) For any V_n, give conditions in terms of i, j, k, and ℓ that make $X_{(1,i)} \perp\!\!\!\perp X_{(1,j)} \mid X_{(k,\ell)}$ for $j > i$.

(c) (4 pt) How many variables are referenced in the largest factor needed to compute $P(X_{(10,1)})$ in V_{10} if variables are eliminated in an order that minimizes the size of the largest factor.

Hints: $P(X_{(1,1)})$ *references one variable,* $P(X_{(1,1)}, X_{(2,1)})$ *references two, etc. Include factors created just before summing out a variable.*

For the remaining parts of this question, assume each $X_{(i,j)}$ takes values *true* and *false*.

(d) (2 pt) Given the factors below for two variables in V_3, fill in the table for $P(X_{(1,1)}|\neg x_{(3,1)})$ or state that there is not enough information.

$X_{(1,1)}$	$P(X_{(1,1)})$
$x_{(1,1)}$	1/3
$\neg x_{(1,1)}$	2/3

$X_{(1,1)}$	$X_{(3,1)}$	$P(X_{(3,1)}\|X_{(1,1)})$
$x_{(1,1)}$	$x_{(3,1)}$	1/3
$x_{(1,1)}$	$\neg x_{(3,1)}$	2/3
$\neg x_{(1,1)}$	$x_{(3,1)}$	0
$\neg x_{(1,1)}$	$\neg x_{(3,1)}$	1

$X_{(1,1)}$	$X_{(3,1)}$	$P(X_{(1,1)}\|\neg x_{(3,1)})$
$x_{(1,1)}$	$\neg x_{(3,1)}$	
$\neg x_{(1,1)}$	$\neg x_{(3,1)}$	

(e) (3 pt) In V_4, if $P(X_{(1,h)} = true) = \frac{1}{3}$ for all h, and $P(X_{(i,j)}|X_{(i-1,j)}, X_{(i-1,j+1)})$ is defined below for all $i > 1$, what is the joint probability of all variables when $X_{(i,j)}$ is *true* if and only if $i + j$ is even.

$X_{(i,j)}$	$X_{(i-1,j)}$	$X_{(i-1,j+1)}$	$P(X_{(i,j)}\|X_{(i-1,j)}, X_{(i-1,j+1)})$
true	*true*	*true*	1
true	*false*	*true*	1/2
true	*true*	*false*	1/2
true	*false*	*false*	1/3

(f) (3 pt) Given the definition of V_4 above, formulate a CSP over variables $X_{(i,j)}$ that is satisfied by only and all assignments with non-zero joint probability.

(g) (2 pt) Does there exist a binary CSP that is equivalent to the CSP you defined?

(h) (4 pt) Circle all the following statements that are true

i. It is impossible for such a CSP to have a satisfying assignment

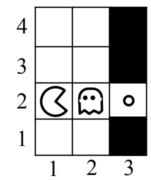
ii. All satisfying assignments of the CSP have joint probability greater than $\frac{1}{32}$ in the original Bayes net

iii. All assignments with joint probability greater than $\frac{1}{32}$ will satisfy the CSP

iv. For any variable, at most one of its values can be part of a satisfying assignment for the CSP

5. (20 points) The Last Dot

Pacman is one dot away from summer vacation. He just has to outsmart the ghost, starting in the game state to the right. Pacman fully observes every game state and will move first. The ghost only observes the starting state, the layout, its own position, and whether or not Pacman is in a middle row (2 and 3) or an edge row (1 and 4). The ghost probabilistically tracks what *row* pacman is in, and it moves around in column 2 to block Pacman.



Starting state

Details of the ghost agent:

- The ghost knows that Pacman starts in row 2. It *assumes* Pacman changes rows according to:
  - In rows 1 and 4, Pacman stays in the same row with probability $\frac{9}{10}$ and moves into the adjacent center row otherwise.
  - In rows 2 and 3, Pacman moves up with probability $\frac{1}{2}$, down with probability $\frac{3}{8}$, and stays in the same row with probability $\frac{1}{8}$.
- Each turn, the ghost *observes* whether Pacman is in a middle row (2 and 3) or an edge row (1 and 4).
- Each turn, the ghost moves up, moves down or stops
- The ghost moves toward the row that most probably contains Pacman according to its model and observations. If the ghost believes Pacman is most likely in its current row, then it will stop.

(a) (2 pt) According to the ghost's *assumed*transition and emission models, what will be the ghost's belief distribution over Pacman's row R if Pacman moves up twice, starting from the game state shown?

Row 1	Row 2	Row 3	Row 4

(b) (2 pt) If Pacman stops twice in a row (starting from the start state), **where** will the ghost be according to its policy? **With what probability** does the ghost believe it is in the same row as Pacman?

Ghost position	Ghost belief probability

(c) (2 pt) If Pacman moves down, then up repeatedly, alternating between positions (1,1) and (1,2), what will the ghost's belief distribution be about Pacman's row after n moves, as n goes to infinity? *Hint: you may want to consider multiple cases.*

(d) (4 pt) If Pacman moves up, then down repeatedly, alternating between positions (1,3) and (1,2), what will the ghost's belief distribution be about Pacman's row after n moves as n goes to infinity? *Hint: if* $ax^2 + bx + c = 0$*, then* $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

(e) (4 pt) Describe the **state space** for a problem that Pacman would need to solve to safely reach the food in as few steps as possible, and name an **algorithm** that could solve this problem.

(f) (4 pt) Precisely describe the path or policy that Pacman should follow in order to reach the food in as few steps . *Briefly* justify your answer.

(g) (2 pt) Describe how Pacman's problem, needed algorithms & optimal path or policy would change if he only observed the start state, but could not observe subsequent states while playing?