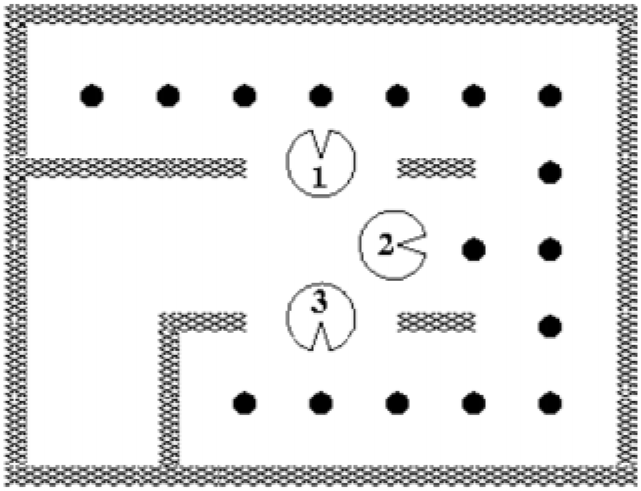
CS221 Practice Midterm #2 Solutions

Summer 2013

Updated 4:00pm, July 24

2 [Deterministic Search] Pacfamily (20 points)

Pacman is trying eat all the dots, but he now has the help of his family! There are initially k dots, at positions $(f_1, \ldots, f_k)$. There are also n Pac-People, at positions $(p_1, \ldots, p_n)$; initially, all the Pac-People start in the bottom left corner of the maze. Consider a search problem in which all Pac-People move simultaneously; that is, in each step each Pac-Person moves into some adjacent position (N, S, E, or W, no STOP). Note that any number of Pac-People may occupy the same position.



(a) Define the state space of the search problem.

The state space consists of the following: a k-tuple of boolean variables E, where $E_i = 1$ if the ith food has been eaten and 0 otherwise, and the n-tuple of Pac-People positions $P = (p_1, \ldots, p_n)$. We assign a cost of 2 for a movement of Pac-People which does not result in a food being eaten, and a cost of 1 for a movement which does.

(b) Give a reasonable upper bound on the size of the state space for a general r by c grid.

To represent just the tuple E we need 2^k states. In addition, there will be a state for each possible arrangement of the Pac-People, and since each Pac-Person can be in one of rc positions, we need $(rc)^n$ states just for the Pac-People positions, so all in all we have an upper (and lower) bound of $2^k(rc)^n$ states in our state space.

(c) What is the goal test?

The goal test in this case is whether or not the tuple E consists entirely of 1's, representing that all of the food has been eaten.

(d) What is the maximum branching factor of the successor function in a general grid?

The maximum branching factor of the successor function in a general grid will be 4^n, since there are n Pac-People and each of them can go in one of four directions, N, S, E, or W. (Note that the minimum may be lower, since different combinations of directions may result in the same state, for example if a Pac-Person is in the upper-left corner then going left and going up results in the same outcome for that Pac-Person).

(e) Circle the admissible heuristics below ($-1/2$ point for each mistake.)

- $h_1(s) = 0$

  This is admissible since the cost of getting to the goal state will never be below 0. (Even with one food left, we will still need to pay a cost of at least 1 to move a Pac-Person to that food.)

- $h_2(s) = 1$

  This is admissible in this case, since the cost of getting to the goal state will always be at least 1. However, if we set our costs differently this may become non-admissible, i.e., if the cost of eating a food was changed to 0.5 then this function would overestimate the cost of reaching the goal state in some cases.

- $h_3(s) = \frac{\text{number of remaining food}}{n}$

  This is admissible in our case since number of remaining food on its own will never overestimate the cost of reaching the goal (since it costs at least 1 to eat each food piece), and dividing by n will only increase the margin by which we underestimate.

- $h_4(s) = \max_i \max_j \mathsf{manhattan}(p_i, f_j)$

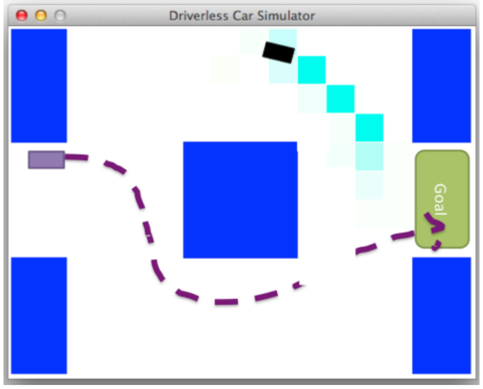
  This function intuitively corresponds to "the distance between the farthest food/Pac-Person pair". This is not admissible since, for example, there could be two Pac-People, one of whom can eat the final food in one move and another who would require 5 moves, and in this case the function will overestimate the cost for reaching the goal as 5 when it is actually 1.

- $h_5(s) = \max_i \min_j \mathsf{manhattan}(p_i, f_j)$

  This function intuitively corresponds to "the farther distance any Pac-Person has to their closest food". This is still not admissible, since it could be the case that one Pac-Person is far away from the final food (say 5 moves away) while another Pac-Person is directly adjacent to it, and this function would predict 5 rather than the actual cost of 1.

3 [Adversarial Search] Expectimax Car (15 points)

The goal of this problem is to extend the self driving car to reason about the future and use that reasoning to make a utility maximizing decision on how to act.



In this problem we are going to assume that the world is the same as in the Driverless Car programming problem. There is a single agent that exists in a closed world and wants to drive to a goal area. For this problem we are going to assume that there is only one other car.

Each heartbeat (there are twenty per second) each car can perform one of four actions Accelerate, TurnLeft, TurnRight, Brake, None. Assume that for each tile we have a probability distribution that the other car will take each of those actions. A car always has a wheel position and a velocity. Even though the car might not be taking the Accelerate action, if it has positive velocity it may continue to move forward.

[important] For this problem assume that cars are manufactured so precisely that if they take an action there is no uncertainty as to how the car will respond. Also assume that you know the start state of both your car and the other.

Formalize the problem of choosing an action as a Markov decision problem:

(a) What variables make up a state?

We will need to include both cars' current velocity, wheel position, what direction it is facing (up, down, left, or right), and location (what tile it is in).

(b) What is the start state?

The start state is the state in which the location of both cars is set to their known start positions, their velocities are set to zero, and their wheel positions are set to "forward", and the direction of the controlled car is set to "right" and of the other car is set to "left".

(c) For each state what are the legal actions?

In a given state, the legal actions are Accelerate, TurnLeft, TurnRight, Brake, and None. (Note that these correspond to the actions for the car we have control over. The "action" taken by the other car will actually just be represented by the probability distribution on the edges in the MDP.)

(d) What is a terminal condition for a given state S? For all terminal conditions specify a utility.

If G is the range of locations comprising the goal area, then one terminal condition is whether the controlled car's location is in G. The utility for this goal should be very large, say 100. Another terminal condition is whether the location of the controlled car is the same as the location of the other car. This corresponds to the cars crashing, and thus should have a very low utility, say 1.

(e) Given a state S from which your agent took action A, what is the successor state distribution?

If $A = Accelerate$, the successor is a state wherein the controlled car's velocity is one higher than the velocity in the current state and its location is determined by location(), which is defined below.

If $A = TurnLeft$, the successor is a state wherein the controlled car's wheel position is the position to the left of its wheel position in the current state ("right" → "forward", "forward" → "left"). If the wheel position is already "left", then the next state has the same wheel position as the current state ("left" → "left"). Location is set by location().

If $A = TurnRight$, the opposite of TurnLeft happens ("left" → "forward", "forward" → "right", "right" → "right"). Location is set by location().

If $A = Brake$, the opposite of Accelerate happens, except that if the controlled car's velocity in the current state is zero then its velocity in the next state remains zero. Location is set by location().

If $A = None$, the controlled car's properties in the next state are the same as its properties in the current state, except its location which is set by location().

location() is set as follows: if the wheel position is "forward" then the controlled car's position is set to be v units in the direction it is facing, where v is its velocity. If the wheel position is "left" or "right" then its position is set to be $v/\sqrt{2}$ units (rounded to the nearest integer) in the direction it is facing and $v/\sqrt{2}$ units (again rounded to the nearest integer) in its relative "left" or "right" direction (respectively). (This is based on me handwaving over some physics things with trig functions that

I don't exactly remember, but you can throw in some cosines and sines and stuff to make it more accurate...)

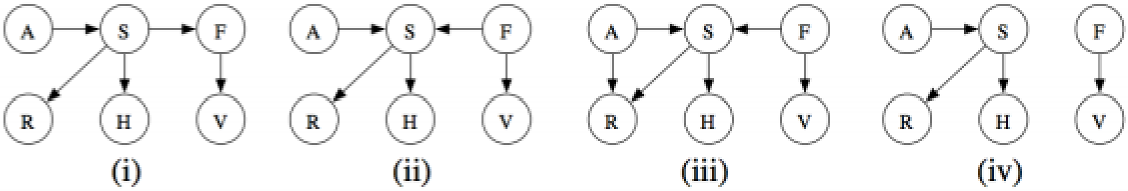
(f) We are going to solve this problem using expectimax. However, we may not want to expand the entire tree. Give a reasonable heuristic utility for a given state S.

One heuristic could be just the **negative of** the maximum of (a) the minimum manhattan distance from the controlled car to the locations comprising the goal area minus the manhattan distance from the controlled car to the other car and (b) zero, in case (a) is below zero (we don't really want a heuristic function to have negative values, even though we technically wouldn't be breaking any rules if it did have them).

4 [Bayes Net] Snuffles (20 points)

Assume there are two types of conditions: (S)inus congestion and (F)lu. Sinus congestion is caused by (A)llergy or the flu. There are three observed symptoms for these conditions: (H)eadache, (R)unny nose, and fe(V)er.

Runny nose and headaches are directly caused by sinus congestion (only), while fever comes from having the flu (only). For example, allergies only cause runny noses indirectly. Assume each variable is boolean.



(i) (ii) (iii) (iv)

(a) Consider the four Bayes Nets shown. Circle the one which models the domain (as described above) best.

(i) is not what we want, since it has an arrow from S to F even though the problem states that sinus congestion is caused by the flu, thus the arrow should point from F to S.

(iii) is no good because we have an arrow from A to R even though the problem states that runny nose is caused *only* by sinus congestion, and not also by allergies as (iii) would model.

(iv) is bad because there does not exist an arrow from F to S, even though the problem states that sinus congestion is caused by the flu.

Thus (ii) is the best model for our domain.

(b) For each network, if it models the domain exactly as above, write correct. If it has too many conditional independence properties, write extra independence and state one that it has but should not have. If it has too few conditional independence properties, write missing independence and state one that it should have but does not have.

This information is also included in the solution to part (a), but for completeness we have:

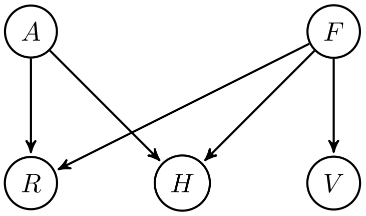
(i): Missing independence, F does not depend on S. Extra independence: S should depend on F.

(ii): Correct

(iii): Missing independence, R should not depend on A.

(iv): Extra independence: S should depend on F.

(c) Assume we wanted to remove the Sinus congestion (S) node. Draw the minimal Bayes Net over the remaining variables which can encode the original model's marginal distribution over the remaining variables.



(d) In the original network you chose, which query is easier to answer: $P(F|r, v, h, a, s)$ or $P(F)$? Briefly justify.

$P(F)$ is easier to answer in the original network because F is not dependent on any other node, so we don't need to use Bayes' rule for incorporating r, v, h, a, s into our probability, we can simply find it via our prior probability or empirical observation proportion of fevers.

(e) Assume the following samples were drawn from prior sampling:

- $a, s, r, \neg h, \neg f, \neg v$
- $a, s, \neg r, h, f, \neg v$
- $a, \neg s, \neg r, \neg h, \neg f, \neg v$
- $a, \neg s, \neg r, h, f, \neg v$
- $a, s, \neg r, h, \neg f, \neg v$

Give the sample estimate of $P(f)$ or state why it cannot be computed.

Given this sample, $P(f) = 2/5$.

Give the sample estimate of $P(f|h)$ or state why it cannot be computed.

$P(f|h) = 2/3$.

5 [Temporal Models] The Jabberwock (25 points)

You have been put in charge of a Jabberwock for your friend Lewis. The Jabberwock is kept in a large tugleywood which is conveniently divided into an $N \times N$ grid. It wanders freely around the N^2 possible cells. At each time step $t = 1, 2, 3, \ldots$, the Jabberwock is in some cell $X_t \in \{1, \ldots, N\}^2$, and it moves to cell X_{t+1} randomly as follows: with probability $1 - \varepsilon$, it chooses one of the (up to 4) valid neighboring cells uniformly at random; with probability ε, it uses its magical powers to teleport to a random cell uniformly at random among the N^2 possibilities (it might teleport to the same cell). Suppose $\varepsilon = \frac{1}{2}$, $N = 10$, and that the Jabberwock always starts in $X_1 = (1, 1)$.

(a) Compute the probability that the Jabberwock will be in $X_2 = (2, 1)$ at time step 2.

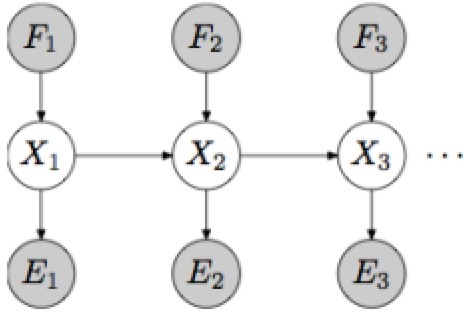
$P(X_2 = (2, 1)) = (1 - \varepsilon)(1/2) + (\varepsilon)(1/N^2)$

What about $P(X_2 = (4, 4))$?

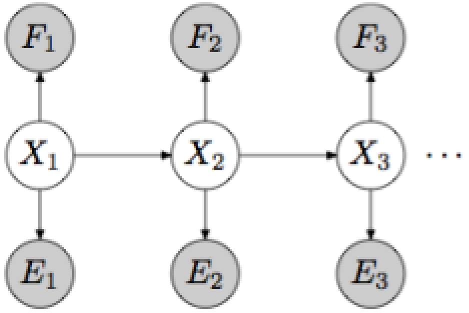
$P(X_2 = (4, 4)) = (1 - \varepsilon)(0) + (\varepsilon)(1/N^2) = \varepsilon/N^2$

(b) At each time step t, you don't see X_t but see E_t, which is the row that the Jabberwock is in; that is, if $X_t = (r, c)$, then $E_t = r$. You still know that $X_1 = (1, 1)$.

You are a bit unsatisfied that you can't pinpoint the Jabberwock exactly. But then you remembered Lewis told you that the Jabberwock teleports only because it is frumious on that time step, and it becomes frumious independently of anything else. Let us introduce a variable $F_t \in \{0, 1\}$ to denote whether it will teleport at time t. We want to add these frumious variables to the HMM. Consider the two candidates:



(A)



(B)

(A)	(B)
→ X_1 independent of $X_3 \mid X_2$	X_1 independent of $X_3 \mid X_2$ ←
→ X_1 independent of $E_2 \mid X_2$	X_1 independent of $E_2 \mid X_2$ ←
X_1 independent of $F_2 \mid X_2$	X_1 independent of $F_2 \mid X_2$ ←
→ X_1 independent of $E_4 \mid X_2$	X_1 independent of $E_4 \mid X_2$ ←
→ X_1 independent of $F_4 \mid X_2$	X_1 independent of $F_4 \mid X_2$ ←
→ E_3 independent of $F_3 \mid X_3$	E_3 independent of $F_3 \mid X_3$ ←
E_1 independent of $F_2 \mid X_2$	E_1 independent of $F_2 \mid X_2$ ←
→ E_1 independent of $F_2 \mid E_2$	E_1 independent of $F_2 \mid E_2$

(c) For each model, circle the conditional independence assumptions above which are true in that model.

(See arrows above)

(d) Which Bayes net is a better representation of the described causal relationships?

(A) is a better representation, because in (B) the "frumious" variable is dependent on the X variables, but in the problem it is stated that the frumious variable is independent on the other variables, as is represented in (A).