

## Problem 1

Prior odds matter when determining if you actually have a disease

$$P(Disease \mid PositiveTest) = \frac{P(PositiveTest \mid Disease) * P(Disease)}{P(PositiveTest)}$$
$$.99 * 1/10000 / (9999/10000 * .01 + 1/10000 * .99) = 0.00980392156$$

## Problem 2

Since the system is markov, you only need to keep track of the last variable to determine the DP structure.

Let  $DP[k][d]$  be

$$\max_{x_1 \dots x_{k-1} \in S^n} P(x_1 \dots x_{k-1}, d)$$

The transition is merely

$$DP[k+1][d] = \max_{v \in S} P(x_{k+1} = d \mid x_k = v) * DP[k][v]$$

After initializing the base case, you fill in a  $m * n$  table with  $m$  calculations per transition, which produces the  $m^2n$  runtime as desired.

## Problem 3

Imagine a three node graph arranged in a cycle with three directed edges with each node having binary outcomes. Let the probability distribution be such that WLOG if node A is 0, then node child(A) is 1 with probability 1, and vice versa. In this case, since your cycle is of odd length, you come to a contradiction after tracing the path one full cycle.

## Problem 4

### 4.1.1

$$P(A \mid B, Y) = \frac{P(A, B, Y)}{P(B, Y)}$$

$P(A, B, Y)$  is unfindable because  $P(B, Y \mid A)$  can't be found without conditional independence or knowing  $P(Y \mid A, B)$

### 4.1.2

Yes, direct application of Bayes Rule

$$P(A \mid B, Y) = \frac{P(A) * P(B, Y \mid A)}{P(B, Y)}$$

### 4.1.3

No, you can't ever find  $P(B, Y)$  because everything you have is conditioned on A and you can't ever separate that away.

### 4.2.1

Yes, because of conditional independence between B and Y given A, you can now do the following

$$P(A | B, Y) = \frac{P(A) * P(B | A) * P(Y | A)}{P(B, Y)}$$

### 4.2.2

Yes, strictly easier than part 4.1.2

### 4.2.3

No, you can't ever find  $P(B, Y)$  because everything you have is conditioned on A and you can't ever separate that away.

## Problem 5

### 5.1.1

In absence of any other information, A and B are independent given neither is the descendent of the other.

$$.8 * .3 = .24$$

### 5.1.2

A gives no information on E because E is a nondescendent of A.

Conditioning on the outputs of B:

$$.7 * .1 + .3 * .9 = .34$$

### 5.2.1

False EHF (V Structure with H known) HFC (Cascade) FCA (Cascade)

### 5.2.2

False GFC (cascade) FCA (cascade) CAD (common parent) ADB (V structure with D known) DBE (common parent)

### 5.2.3

False, there is a cascade from B to H.

## Problem 6

### 6.1

0.5

### 6.2

TODO

### 6.3

$$(.55 + 1)/2 = 0.775$$

### 6.4

Given you are admitted and  $I = 0.95$ , then your creativity ranges from .55 to 1 uniformly. Thus, it is  $1.55/2 = .725$

## Problem 7

### 7.1

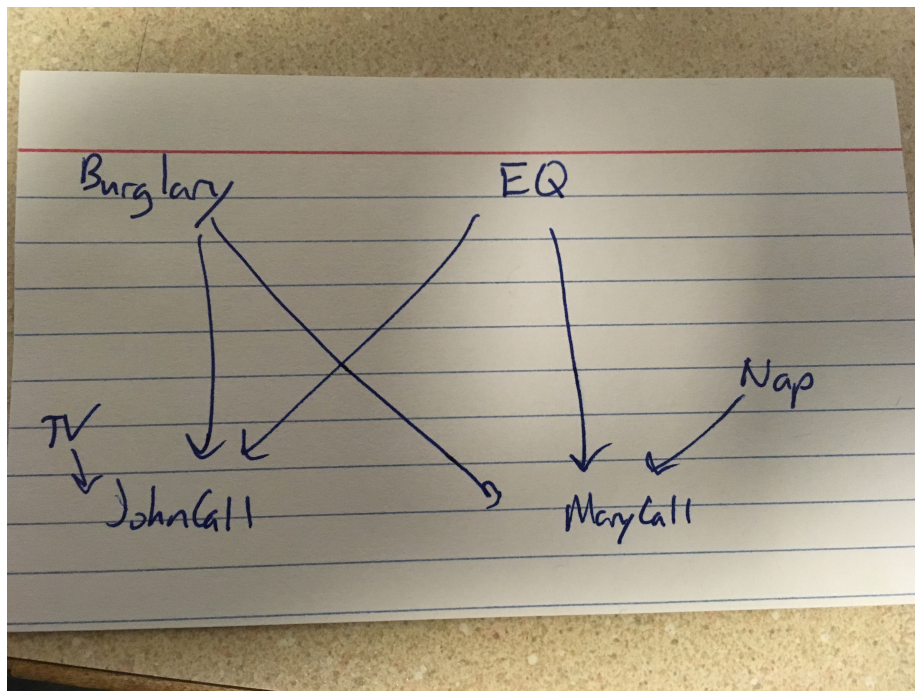


Figure 1: Problem 7 Bayes Net

## 7.2

Because you can't see the probabilities, you have to just go off of the network structure. This means that any valid network is a minimal I map since you can't remove any edge with possibly interfering with the underlying probability distribution. TODO IS THIS RIGHT? BEC

General gist is if you want to remove node X, for each child of X, add all the parents of X (removing duplicates). This works because you are "passing the information" from node X to its children. The exact probability update equation is:

Let C be child of X. Let  $p1..pk$  be the parents of X. Let V be the set of possible values of X

$$P(C = c \mid p1 \dots pk) = \sum_{v \in V} P(C = c \mid v) * P(X = v \mid p1 \dots pk)$$

If I misread the problem, and you indeed can see the probability distributions for each node X, then you need pick the smallest set S such that

$$P(X \mid S) = P(X \mid x1..xn)$$

, then have S be the parents of node X.

## Problem 8

### 8.1

Note that each node only depend on its parents and its descendents. Non descendents tell you nothing about yourself, and ancestors prior to parents are not useful given information about parents.

Thus,

$$P(X \mid par(X), descend(X)) = \frac{P(X, par(X), descend(X))}{P(par(X), descend(X))}$$

### 8.2

Do the hint exactly, pretty much. Notice that

$$P(A, B) = P(A) * P(B \mid A)$$

Decompose the Bayesian net into something like

$$P(A) * P(B) * P(C \mid A, B) \dots$$

Then go down and generate the net by first generating the parents, then their children using their values, until you have generated the whole net.