AI Assignment 2

Nick Triano, Will Keleher

November 2022

1

a) What is the probability that all five of these Boolean variables are simultaneously true?

We find,

$$P(A = T, B = T, C = T, D = T, E = T)$$

using the following formula,

$$P(x_1, x_2..., x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i)) \tag{1}$$

and compute,

$$P(A)P(B)P(C)P(D \mid A, B)P(E \mid B, C) = 0.2 \cdot 0.5 \cdot 0.8 \cdot 0.1 \cdot 0.3 = 0.0024$$

So, the probability that all 5 Boolean variables are simultaneously true is 0.0024.

b) What is the probability that all five of these Boolean variables are simultaneously false?

We find,

$$P(A = F, B = F, C = F, D = F, E = F)$$

using equation (1) from above we compute,

$$P(A)P(B)P(C)P(D \mid A, B)P(E \mid B, C) = 0.8 \cdot 0.5 \cdot 0.2 \cdot 0.1 \cdot 0.8 = 0.0064$$

c) What is the probability that A is false given that the four other variables are all known to be true?

We use our answer to part a,

$$P(A = T, B = T, C = T, D = T, E = T) = 0.0024$$

along with finding this new probability,

$$P(A = F, B = T, C = T, D = T, E = T)$$

and then normalize to find the probability that A is false given that the four other variables are all known to be true. From equation (1) we have,

$$P(A)P(B)P(C)P(D|A, B)P(E|B, C) = 0.8 \cdot 0.5 \cdot 0.8 \cdot 0.6 \cdot 0.3 = 0.0576$$

Normalizing yields,

$$\alpha 0.0024 + \alpha 0.0576 = 1$$

$$\alpha = \frac{1}{0.0024+0.0576}$$

$$\alpha = 16.\overline{66}$$

$$< \alpha 0.0024, \alpha 0.0576 >=< 0.96, 0.04 >$$

So, the probability that A is false given that the four other variables are all known to be true is 0.96.

2

For this problem, check the Variable Elimination algorithm in your book. Also consider the Bayesian network from the "burglary" example.

a) Apply variable elimination to the query: $P(B \mid J, M)$ and show in detail the calculations that take place. Use your book to confirm that your answer is correct.

Given the formula:

$$P(B \mid J, M) = \alpha P(B) \sum_B P(E) \sum_E P(a|B, e)P(j|a)P(m|a)$$

We can create a matrix for each probability that we will call f_n. For example $f_4(A)$ and $f_5(A)$ would each be:

$$f_4(A) = \begin{bmatrix} 0.9 \\ 0.7 \end{bmatrix} \; f_5(A) = \begin{bmatrix} 0.5 \\ 0.01 \end{bmatrix}$$

With this, we create the equation:

$$P(B|J,M) = \alpha f_1(B) \sum_B f_2(E) \sum_E f_3(A,B,E) X f_4(A) X f_5(A)$$

From here, we employ variable elimination to remove terms from the equation. To remove A we will look to the later summation:

$$\sum f_3(A,B,E) X f_4(A) X f_5(A)$$

Expanding this further, we get:

$$f_3(a,B,E) X f_4(a) X f5(a) + f_3(\neg a,B,E) X f_4(\neg a) X f_5(\neg a)$$

Which becomes:

$$\begin{bmatrix} 0.95 & 0.94 \\ 0.29 & 0.001 \end{bmatrix} \text{ X } (0.9) \text{ X } (0.7) + \begin{bmatrix} 0.05 & 0.06 \\ 0.71 & 0.999 \end{bmatrix} \text{X } (0.05) \text{ X } (0.01)$$

$$\begin{bmatrix} 0.5985 & 0.5922 \\ 0.1827 & 0.00063 \end{bmatrix} + \begin{bmatrix} 0.000025 & 0.00003 \\ 0.000355 & 0.0004995 \end{bmatrix} = \begin{bmatrix} 0.598525 & 0.59223 \\ 0.183055 & 0.0011295 \end{bmatrix}$$

$$\begin{bmatrix} 0.598525 \\ 0.183055 \end{bmatrix} * (0.002) + \begin{bmatrix} 0.59223 \\ 0.0011295 \end{bmatrix} * (0.998)$$

$$\begin{bmatrix} 0.59224259 \\ 0.001493351 \end{bmatrix}$$

$$(0.59224259) * (.001), (0.001493351) * (.999)$$

$$a(0.00059224259), a(0.001491857649)$$

Normalizing gives us,

$$P(B|J,M) = 0.284$$

b) Count the number of arithmetic operations performed (additions, multiplications, divisions), and compare it against the number of operations performed by the tree enumeration algorithm.

$$\begin{bmatrix} 0.95 & 0.94 \\ 0.29 & 0.001 \end{bmatrix} \text{ X } (0.9) \text{ X } (0.7) + \begin{bmatrix} 0.05 & 0.06 \\ 0.71 & 0.999 \end{bmatrix} \text{ X } (0.05) \text{ X } (0.01)$$

11 calculations at step

11 total calculations

$$\begin{bmatrix} 0.598525 \\ 0.183055 \end{bmatrix} * (0.002) + \begin{bmatrix} 0.59223 \\ 0.0011295 \end{bmatrix} * (0.998)$$

5 calculations at step

16 total calculations

$$(0.59224259) * (.001), (0.001493351) * (.999)$$

2 calculations at step
18 total calculations

0.001*
1 calculation at step; 1 total calculation
0.002*0.95*0.9*0.7
3 calculations at step; 4 total calculations
0.002*0.05*0.05*0.01
3 calculations at step; 7 total calculations
0.998*0.94*0.9*0.7
3 calculations at step; 10 total calculations
0.999*0.06*0.05*0.01
3 calculations at step; 13 total calculations
Summations
3 calculations at step; 16 total calculations
Repeat Process for $P(\neg b)$ 16 calculation at step; 32 total calculations

Variable Elimination Total Steps = 18, Enumeration Total Steps = 32.

c) Suppose a Bayesian network has the from of a chain: a sequence of Boolean variables $X_1, ...X_n where Parents(X_i) = (X_i - 1)$for $i = 2, ..., n$. What is the complexity of computing $P(X1|Xn = true)$using enumeration? What is the complexity with variable elimination?

The enumeration process will take $O(2^n)$ time to complete the problem where n is the amount of variables in the total problem.

While variable elimination can prove to be a faster algorithm than enumeration, in this example, given that the network is a chain, it will still take $O(2^n)$ time.

3

a) Compute the probabilities of $P(d \mid c)$, $P(b \mid c)$, and $P(d \mid \neg a, b)$ by enumeration.

We use the following equation to compute these probabilities,

$$P(X \mid e) = \alpha P(X, e) = \alpha \sum_{y} P(X, e, y) \tag{2}$$

$P(d \mid c)$:

$$\alpha P(d, c) = \alpha \textstyle\sum_A \sum_B P(A, B, c, d)$$

$$= \alpha \textstyle\sum_A \sum_B P(A)P(B)P(c \mid A, B)P(d \mid B, C)$$

$$= \alpha \textstyle\sum_A P(A) \sum_B P(B)P(c \mid A, B)P(d \mid B, C)$$

$$P(a) = 0,\ P(\neg a) = 1$$

$$= \alpha \textstyle\sum_B P(B)P(c \mid \neg a, B)P(d \mid B, c)$$

$$= \alpha[P(b)P(c \mid \neg a, b)P(d \mid b, c) + P(\neg b)P(c \mid \neg a, \neg b)P(d \mid \neg b, c)]$$

$$= \alpha(0.9 \cdot 0.5 \cdot 0.75 + 0.1 \cdot 0 \cdot 0.5) = \alpha 0.3375$$

$$P(\neg d \mid c) = \alpha 0.1125$$

$$\textit{normalize} < \alpha 0.3375, \alpha 0.1125 >=< 0.75, 0.25 >$$

$$\text{So, } P(d \mid c) = 0.75$$

$P(b \mid c)$:

$$(b, c) = \alpha \textstyle\sum_A \sum_D P(A)P(D \mid b, c)P(c \mid A, b)$$

$$\text{center}= \alpha[P(\neg d \mid b, c)P(c \mid \neg a, b) + (d \mid b, c)P(c \mid \neg a, b)]$$

$$= \alpha[0.9(0.5 * 0.75 + 0.5 * 0.25) = \alpha 0.45$$

Interestingly, when computing for $P(\neg b)$ to normalize α, we get:

$$\alpha[P(c \mid \neg a, \neg b)P(d \mid \neg b, c) + P(c \mid \neg a, \neg b)P(\neg d \mid \neg b, c)]$$

Which is equal to:

$$\alpha 0.1(0 + 0)$$

As $P(c \mid \neg a, \neg b)$ is equal to 0 and appears in both of the calculations. With this we have:

$$normalize < \alpha 0.45, \alpha 0 >=< 1, 0 >$$

Meaning, if C=True, then B can never be false.

$P(d \mid \neg a, b)$:

$$\alpha P(d, \neg a, b) = \alpha \sum_C P(a, b, C, d)$$

$$= \alpha P(\neg a)P(b) \sum_C P(C \mid \neg a, b)P(d \mid b, C)$$

$$= \alpha \cdot 1 \cdot 0.9 \cdot [0.5 \cdot 0.75 + 0.5 \cdot 0.1]$$

$$= \alpha 0.3825$$

$$P(\neg d \mid \neg a, b) = \alpha 0.5175$$

$$normalize < \alpha 0.5175, \alpha 0.3825 >=< 0.425, 0.575 >$$

$$\text{So, } P(d \mid \neg a, b) = 0.425$$

b) Now employ rejection sampling and likelihood weighting to approximate the same three conditional probabilities. Use 1000 samples, and document your results. How do the approximations and the actual values compare?

$P(d \mid c)$:

Rejection Sampling: $(0.7660044150110376, 0.23399558498896247)$
Likelihood Weighting: $(0.7363128491620111, 0.2636871508379888)$
Actual Values: $(0.75, 0.25)$

$P(b \mid c)$:

Rejection Sampling: $(1.0, 0.0)$
Likelihood Weighting: $(1.0, 0.0)$
Actual Values: $(1, 0)$

$P(d \mid \neg a, b)$:

Rejection Sampling: $(0.4087346024636058, 0.5912653975363942)$
Likelihood Weighting: $(0.4140000000000015, 0.5859999999999985)$
Actual Values: $(0.425, 0)$

All estimated values are close to their actual values within 10^{-1}.

c) Next focus on $P(d \mid c)$. We know that the accuracy of the approximation depends on the number of samples used. For each of the sampling methods, plot the probability as a function of the number of samples used. Do you notice a large divergence in the convergence rates among the two methods?

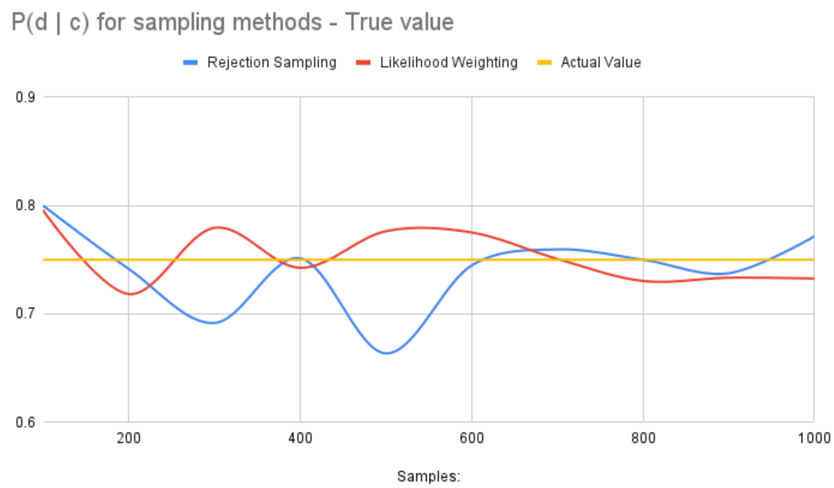

Figure 1: The plotted probability of $P(d \mid c)$ using two sampling methods: Rejection Sampling and Likelihood Weighting *Note: This is just the True value of the normalized vector*

We ran both methods on increments of 100, from 100 to 1000 samples and recorded the results. Rejection Sampling is much more volatile than Likelihood Weighting. In fact, we happened to get the exact probability at 400 samples, but then at 500 samples we get something way off from the actual value. As samples increase they both converge to the actual value, and after about 800 samples Likelihood Weighting maintains a steady rate of convergence, but Rejection Sampling still has some fluctuations.

d) Construct a query on this Bayesian Network such that the convergence and effectiveness of rejection sampling is noticeably worse than that of likelihood weighting. List the query you are using, and provide the probability plot as a function of samples used. Why is it the case that rejection sampling is noticeably worse for this query?

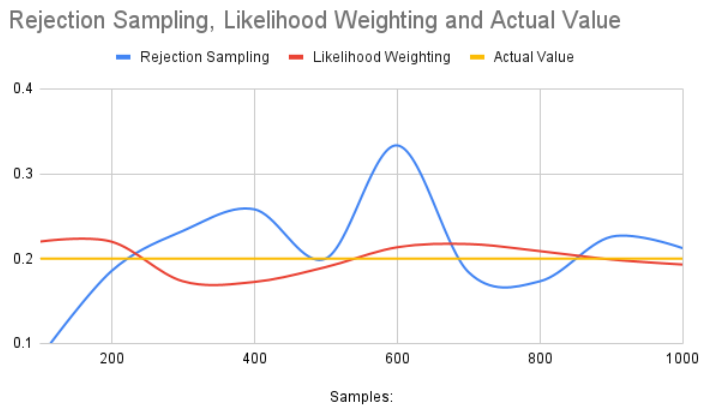

Figure 2: $P(d \mid \neg b)$ calculated with Rejection Sampling and Likelihood Weighting.

We chose this probability because it rejects the most samples without rejecting all of them. This creates a smaller sample size that Rejection Sampling has to calculate the probability from, and creates a huge variation in what it calculates, especially at the lower total sample sizes as shown in the graph.

4

a) Filtering: Three days have passed since the rover fell into the ravine. The observations were $(E_1 = hot, E_2 = cold, E_3 = cold)$. What is $P(X_3 \mid hot_1, cold_2, cold_3)$, the probability distribution over the rover's position on day 3, given the observations? (This is a probability distribution over the six possible positions).

We use the following Filtering formulation to solve this problem.

$$P(X_{t+1} \mid E_{1:t+1}) = \alpha \underbrace{P(E_{1:t+1} \mid X_{t+1})}_{O.M.} \sum_{X_{t+1}} \underbrace{P(X_{t+1} \mid X_t)}_{T.M.} \underbrace{P(X_t \mid E_{1:t})}_{Filtering} \tag{3}$$

And start with,

$$P(X_3 \mid E_{1:3}) = \alpha P(E_{1:3} \mid X_3) \textstyle\sum_{X_3} P(X_3 \mid X_2) P(X_2 \mid E_{1:2})$$

This requires us to solve the sub problem,

$$P(X_2 \mid E_{1:2}) = \alpha P(E_{1:2} \mid X_2) \textstyle\sum_{X_2} P(X_2 \mid X_1) P(X_1 \mid E_1)$$

Since X_1 is our starting state we initialize $P(X_1 \mid E_1) = 1$. And solve the sub problem, *Note: We do not show calculations that end in a probability of 0*

$$P(X_2 \mid E_{1:2}) = \alpha P(E_{1:2} \mid X_2) \sum_{X_2} P(X_2 \mid X_1)$$

$$P(X_2 = B \mid E_1 = hot, E_2 = cold) = \alpha \cdot 1(0.8 \cdot 1) = \alpha 0.8$$

After normalizing,

$$\alpha 0.8 = 1- > \alpha = \frac{1}{0.8} = 1.25- > 0.8 \cdot 1.25 = 1$$

$$\text{Rover's position on Day 2: } \{0, 1, 0, 0, 0, 0\}$$

Now we can solve our problem.

$$P(X_3 \mid E_{1:3}) = \alpha P(E_{1:3} \mid X_3) \sum_{X_3} P(X_3 \mid X_2) P(X_2 \mid E_{1:2})$$

$$X_3 = B : \alpha \cdot 1(0.2 \cdot 1) = \alpha 0.2$$

$$X_3 = C : \alpha \cdot 1(0.8 \cdot 1) = \alpha 0.8$$

After normalizing,

$$\alpha 0.8 + \alpha 0.2 = 1 \rightarrow \alpha = \frac{1}{0.8+0.2} = 1 \rightarrow 0.8 \cdot 1 = 0.8, 0.2 \cdot 1 = 0.2$$

Rover's position on Day 3: $\{0, 0.2, 0.8, 0, 0, 0\}$

b) Smoothing: What is P(X2 — hot1, cold2, cold3), i.e., the probability distribution over the rover's position on day 2, given the observations for the first three days? (Again, provide a probability distribution over all six possible positions).

$$P(X_2 \mid hot_1, cold_2, cold_3) = \alpha \cdot \underbrace{P(cold_3 \mid X_2)}_{Prediction} \cdot \underbrace{P(X_2 \mid hot_1, cold_2)}_{Filtering}$$

We note our vector from the previous filtering problem,

$$P(X_2 \mid hot_1, cold_2) = \{0, 1, 0, 0, 0, 0\}$$

And solve the prediction sub problem,

$$P(cold_3 \mid X_2) = \sum_{X_3} P(cold_3 \mid X_3) P(X_3 \mid X_2) P(O_4 \mid X_3)$$

Setting $P(O_4 \mid X_3) = 1$ and solving yields, *Note: We ignore calculations that result in 0*

$$X_2 = B : (0.2 \cdot 1 + 0.8 \cdot 1) \cdot 1 = 1$$

This gives us our distribution,

$$P(X_2 \mid hot_1, cold_2, cold_3) = \{0, 1, 0, 0, 0, 0\}$$

c) Most Likely Explanation: What is the most likely sequence of the rover's positions in the three days given the observations (E1 = hot, E2 = cold, E3 = cold)?

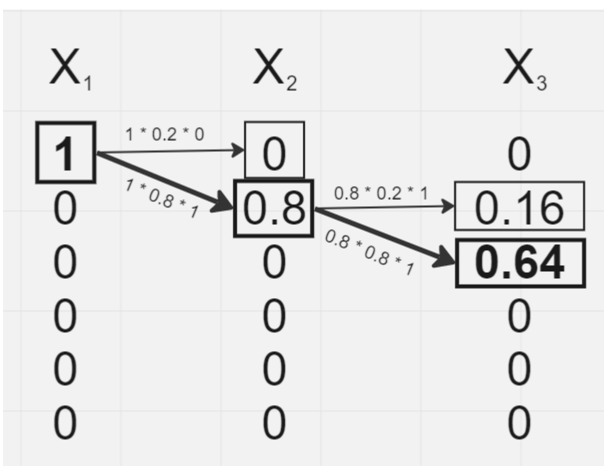

Figure 3: Most Likely Explanation Table. Computed using the Viterbi Algorithm.

The most likely sequence of the rover's positions in the three days given the observations is $(X_1 = A, X_2 = B, X_3 = C)$.

d) Prediction: What is $P(hot4, hot5, cold6 \mid hot1, cold2, cold3)$? i.e., the probability of observing the sequence hot4 and hot5 and cold6 in days 4,5,6 respectively, given the previous observations in days 1,2, and 3? (This is a single value, not a distribution).

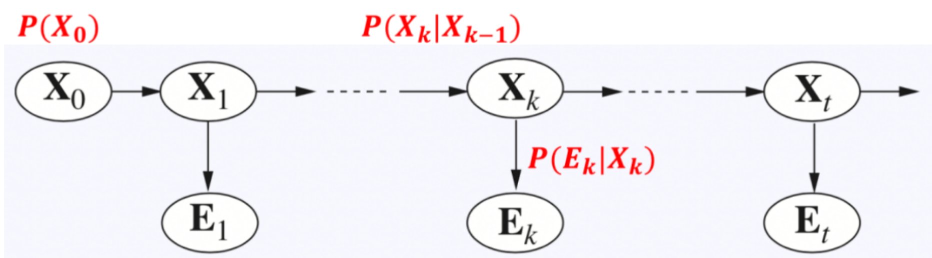

Figure 4: Dynamic Bayesian Network from lecture slides

Evidence is independent from previous evidence. Evidence only depends on the state at that time step. Because of this we can reduce the problem down to the following,

$$P(hot4, hot5, cold6)$$

Given evidence hot_4 we compute the probability of moving to D, a hot cell.

$$P(X_4 = D \mid X_3) = 0.8 \cdot 0.8$$

Also, given evidence hot_5 we compute the probability staying on cell D.

$$P(X_5 = D \mid X_4 = D) = 0.2$$

Now, given evidence $cold_6$ we compute the probability moving to cell E.

$$P(X_6 = E \mid X_5 = D) = 0.8$$

Multiplying these together to get our answer to the query.

$$P(hot4, hot5, cold6 \mid hot1, cold2, cold3) = 0.8 \cdot 0.8 \cdot 0.2 \cdot 0.8 = 0.1024 = 10.24\%$$

e) Prediction: You decide to attempt to rescue the rover on day 4. However, the transmission of E4 seems to have been corrupted, and so it is not observed. What is the probability distribution for the rover's position 4 given the same evidence, $P(X4 \mid hot1, cold2, cold3)$? The same thing happens again on day 5. What is the probability distribution for the rover's position?

We use the Prediction formula to solve this problem.

$$P(X_{t+1} \mid E_{1:t}) = \sum_{X_t} P(X_{t+1} \mid X_t) P(X_t \mid E_{1:3})$$

Applying the formula to this problem yields,

$$P(X_4 \mid E_{1:3}) = \sum_{X_3} P(X_4 \mid X_3) P(X_3 \mid E_{1:3})$$

Our prior distribution is

$$P(X_3 \mid E_{1:3}) = \{0, 0.2, 0.8, 0, 0, 0\}$$

So we compute for all values of X_4,

$$\sum_{X_3} P(X_4 \mid X_3) P(X_3 \mid E_{1:3}) = \{0, 0.2 \cdot 0.2, 0.2 \cdot 0.8 + 0.8 \cdot 0.2, 0.8 \cdot 0.8, 0, 0\}$$

And we have the probability distribution for the rover's position on day 4,

$$P(X_4 \mid E_{1:3}) = \{0, 0.04, 0.32, 0.64, 0, 0\}$$

We perform the same computations on $P(X_5 \mid E_{1:3})$ using the distribution we just calculated to get the rover's position on day 5.

$$P(X_5 \mid E_{1:3}) = \{0, 0.008, 0.096, 0.384, 0.512, 0\}$$

5

Step A:

0.002	0.023	0.023
0.042	0.421	0.421
0.042	0	0.023

Distribution i = 1

0.00003	0.0005	0.003
0.008	0.157	0.824
0.008	0	0.0003

Distribution i = 2

< 10^-5	< 10^-4	< 10^-4
< 10^-4	0.0012	0.0063
0.006	0	0.9918

Distribution i = 3

< 10^-5	< 10^-4	< 10^-5
< 10^-5	< 10^-4	0.0003
< 10^-4	0	0.9994

Distribution i = 4

Figure 6: Four 3 x 3 maps with sets of eight probabilities indicating where we are inside this grid world after each action/sensing pair

Step B:

```
current map=
0
starting location =
(6, 1)
['H', 'N', 'H', 'N', 'H', 'N', 'N', 'T', 'H', 'H']
['T', 'B', 'T', 'B', 'T', 'T', 'N', 'N', 'N', 'H']
['N', 'T', 'N', 'N', 'N', 'H', 'N', 'N', 'B', 'N']
['H', 'T', 'N', 'N', 'H', 'T', 'T', 'H', 'H', 'N']
['N', 'T', 'N', 'N', 'N', 'N', 'N', 'T', 'N', 'T']
['H', 'N', 'H', 'H', 'N', 'H', 'T', 'N', 'H', 'H']
['N', 'H', 'T', 'N', 'B', 'N', 'N', 'B', 'T', 'N']
['B', 'N', 'N', 'N', 'T', 'T', 'N', 'N', 'N', 'N']
['N', 'H', 'N', 'T', 'H', 'H', 'N', 'N', 'T', 'N']
['T', 'N', 'T', 'N', 'H', 'N', 'N', 'H', 'N', 'N']
Start State: (6, 1)
Next State: (6, 2), Sensor Reading: T, Action: R
Next State: (7, 2), Sensor Reading: N, Action: D
Next State: (7, 1), Sensor Reading: N, Action: L
Next State: (7, 1), Sensor Reading: N, Action: L
Next State: (7, 1), Sensor Reading: N, Action: D
current map=
1
starting location =
(3, 7)
['T', 'T', 'N', 'T', 'N', 'N', 'B', 'H', 'T', 'N']
['H', 'H', 'N', 'T', 'N', 'T', 'N', 'N', 'N', 'T']
['B', 'T', 'T', 'B', 'N', 'T', 'B', 'N', 'N', 'T']
['N', 'B', 'T', 'N', 'N', 'N', 'N', 'N', 'H', 'H']
['N', 'T', 'N', 'T', 'T', 'N', 'N', 'N', 'T', 'H']
['N', 'H', 'H', 'N', 'N', 'N', 'N', 'N', 'H', 'N']
['H', 'N', 'N', 'N', 'B', 'H', 'T', 'N', 'T', 'T']
['B', 'H', 'N', 'N', 'N', 'H', 'N', 'H', 'N', 'H']
['B', 'H', 'H', 'T', 'B', 'B', 'H', 'T', 'N', 'T']
['T', 'N', 'H', 'T', 'H', 'N', 'H', 'H', 'H', 'T']
Start State: (3, 7)
Next State: (2, 7), Sensor Reading: N, Action: U
Next State: (3, 7), Sensor Reading: N, Action: D
Next State: (3, 8), Sensor Reading: H, Action: R
Next State: (4, 8), Sensor Reading: T, Action: D
Next State: (4, 7), Sensor Reading: N, Action: L
current map=
2
starting location =
(0, 9)
['T', 'N', 'N', 'N', 'N', 'N', 'H', 'H', 'N', 'N']
['B', 'H', 'N', 'N', 'N', 'B', 'B', 'T', 'N', 'N']
['N', 'H', 'N', 'T', 'T', 'N', 'N', 'T', 'H', 'T']
['N', 'H', 'T', 'N', 'H', 'N', 'H', 'N', 'N', 'T']
['N', 'H', 'T', 'H', 'N', 'H', 'N', 'T', 'N', 'N']
['H', 'B', 'T', 'H', 'N', 'N', 'N', 'B', 'N', 'H']
['H', 'N', 'T', 'H', 'N', 'T', 'N', 'N', 'B', 'T']
['H', 'H', 'H', 'T', 'N', 'N', 'T', 'N', 'H', 'H']
['N', 'N', 'B', 'H', 'N', 'N', 'N', 'N', 'N', 'N']
['B', 'T', 'N', 'T', 'N', 'N', 'N', 'N', 'H', 'H']
Start State: (0, 9)
Next State: (0, 9), Sensor Reading: N, Action: U
Next State: (0, 8), Sensor Reading: N, Action: L
Next State: (0, 9), Sensor Reading: N, Action: R
Next State: (0, 8), Sensor Reading: N, Action: L
Next State: (0, 7), Sensor Reading: H, Action: L
```

Figure 6: This is a print statement for our map program scaled down to be a 10 by 10 map. This shows what our maps look like. In further figures, we will examine what our return files look like.

```
(6, 1)
[(6, 1), (6, 2), (7, 2), (7, 1), (7, 1), (7, 1)]
['R', 'D', 'L', 'L', 'D']
['T', 'N', 'N', 'N', 'N']
```

```
(3, 7)
[(3, 7), (2, 7), (3, 7), (3, 8), (4, 8), (4, 7)]
['U', 'D', 'R', 'D', 'L']
['N', 'N', 'H', 'T', 'N']
```

```
(0, 9)
[(0, 9), (0, 9), (0, 8), (0, 9), (0, 8), (0, 7)]
['U', 'L', 'R', 'L', 'L']
['N', 'N', 'N', 'N', 'H']
```

Figure 7: This is what our output files look like, again scaled down to only 5 actions and 5 sensor readings so it is easier to see.

Our format follows the one in the assignment. From top to bottom we have the starting coordinate 0 indexed from the top left of our matrix. An array of coordinates in (row, col) format. An array of characters corresponding to actions, and then an array of characters corresponding to the readings on the sensor. Our files that we generated show 100 actions, sensors, and coordinates from maps which are 100 by 50.

5.1 Part B

Example heat maps: White squares have a probability less than 0.0002. Higher probabilities have an increasing value of red. Green squares show ground truth path up to that iteration.

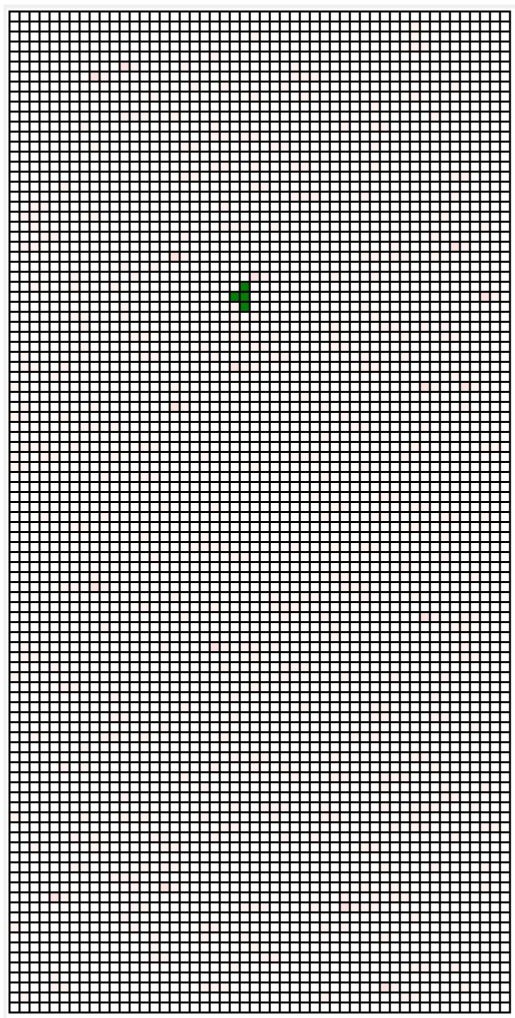

Figure 8: Example heat map after 10 iterations

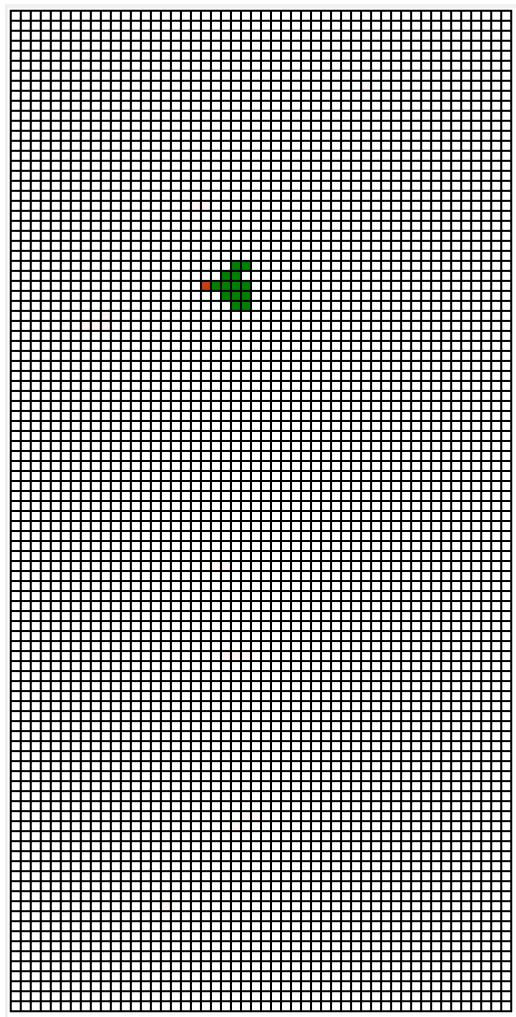

Figure 9: Example heat map after 50 iterations

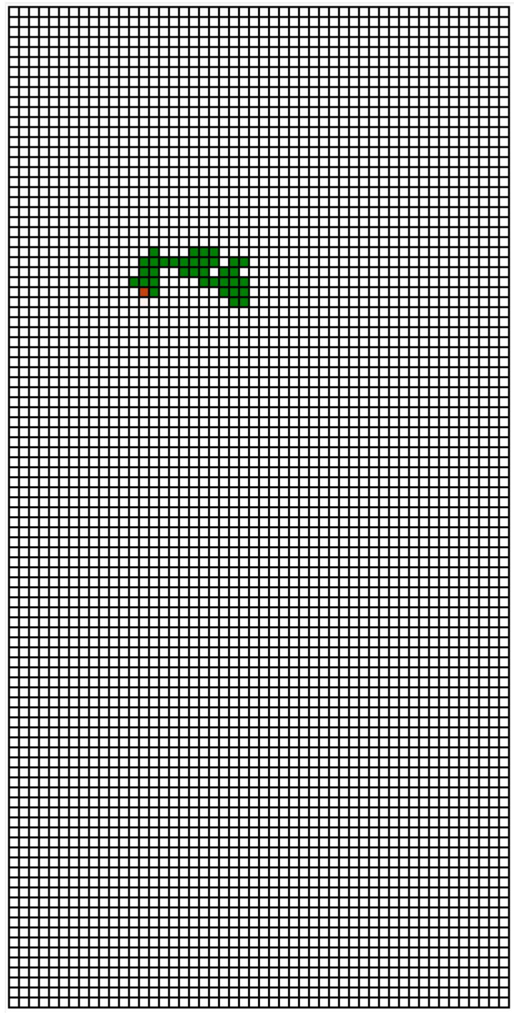

Figure 10: Example heat map after 100 iterations

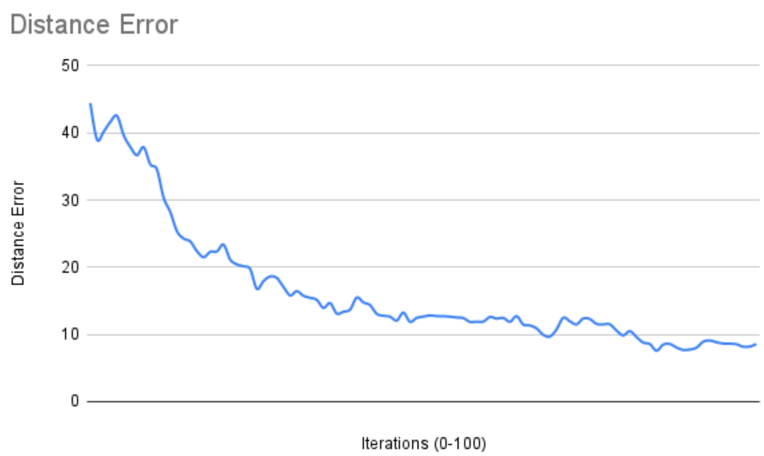

Figure 11: Generated plot of the average error over all 100 experiments as the number of readings increases.

Computed error as distance inside the grid world between the true location of the agent and the maximum likelihood estimation (i.e., the cell with the highest probability according to the filtering algorithm) as the number of readings increases. For the computation of the maximum likelihood estimation, ties were broken randomly.

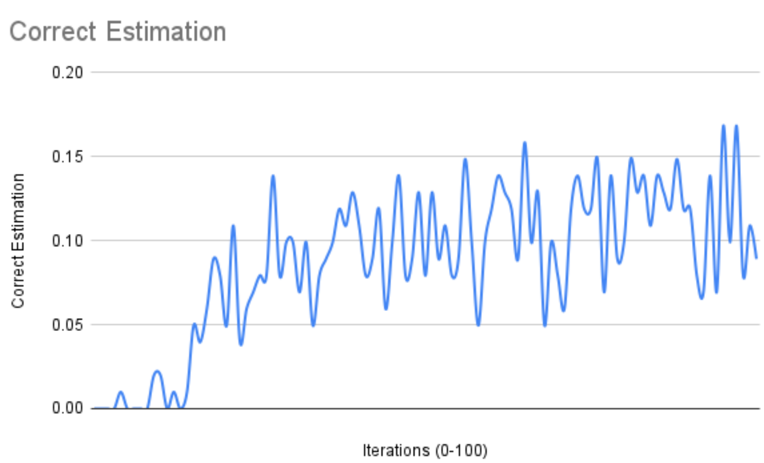

Figure 12: Generated plot of the average probability of the ground truth cell over 100 experiments as the number of readings increases.

At every iteration for all 100 experiments we tracked if the agent's true location was correctly guessed by the algorithm. We put a 1 (true location was estimated location) or a 0 for this case. We then averaged the results across all experiments at the given iteration. We, can see a general increasing function of the graph that highly fluctuates between 5% and 15% correct state estimation at any given iteration.

Our filtering algorithm needed two optimizations to make it run in a competent amount of time. Our first optimization was to use Dynamic Programming. We used Memoization to store the sub filtering problems in an array and then check whether we solved the necessary sub step before computing the current one. Then we can look up the necessary probability in constant time. Another optimization we performed was eliminating checking improbable locations to be at. For example, if the action was down, and we were currently checking where we were given we went down and ended up at (5,5). We would only check our transition model with the previous locations of (5,5) and (6,5), because any other locations are improbable and unnecessary to compute.

6

a

Assume you are interested in buying a used vehicle C_1. You are also considering of taking it to a qualified mechanic and then decide whether to buy it or not. The cost of taking it to the mechanic is $100. C_1 can be in good shape (quality q+) or bad one (quality q-). The mechanic might help to indicate what shape the vehicle is in. C_1 costs $3,000 to buy and its market value is $4,000 if in good shape; if not, $1,400 in repairs will be needed to make it in good shape. Your estimate is that C_1 has a 70% chance of being in good shape. Assume that the utility function depends linearly on the vehicle's monetary value.

Initial cost = $3000 - IC

Good Condition Market Value = $4000 - GCMV

Good Condition Chance = 0.7 - GCC

Bad Condition Repair Cost = $1400 - BCRC

Bad Condition Chance = .3 - BCC

Bad Condition Net = $2600 BCN

$$(GCMV \cdot GCC + BCN \cdot BCC) - IC$$

(4000*0.7+2600*0.3)-3000= $3,580

b

We also have the following information about whether the vehicle will pass the mechanic's test:

P(pass(c1)| $q + (c1)$) = 0.8

P(pass(c1)| $q(c1)$) = 0.35

Use Bayes' theorem to calculate the probability that the car will pass/fail the test and hence the probability that it is in good/ bad shape given what the mechanic will tell you.

P(pass(c1)) = $GCC * P(pass(c1)|q+) + BCC * P(pass(c1)|q-) =$

0.665

P(fail(c1))=0.335

P(q+) =0.7

P(q-) =0.3

P(pass(c1)| $q+$) = 0.8

P(fail(c1) mid q+)=0.2

P(pass(c1)| $q-$) = 0.35

P(fail(c1)| $q-$) = 0.65

Baye's Theorem

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

$$P(q+ \mid Pass) = \frac{P(Pass \mid q+) \cdot P(q+)}{P(Pass)}$$

$$P(q+ \mid Pass) = \frac{0.8 \cdot 0.7}{0.665}$$

$P(q+ \mid Pass) = 0.8421$
$P(q- \mid Pass) = 0.1579$
$P(q+ \mid Fail) = 0.4179$
$P(q- \mid Fail) = 0.5821$

c

What is the best decision given either a pass or a fail? What is the expected utility in each case?

Given a passing grade by the mechanic, you have an 84% chance that the car is in good condition and given a failing grade by the mechanic, you still have almost 50% odds that the car is in good condition. With this information, given a passing grade from the mechanic, the best decision would be to buy the car. The estimated profit from purchasing the car given a passing grade from the mechanic is $778.94 while the estimated profit from a failing grade is surprisingly above 0, totaling $185.06. While a failing grade from the mechanic might seem to be an indication not to buy the car, the potential gains being $600 higher than the potential losses combined with a nearly 50% chance that the mechanic is simply wrong, lead to the estimated profits still being positive. In both situations, you should buy the car.

d

What is the value of optimal information for the mechanic's test? Will you take C1 to the mechanic or not?

Not taking the car to the mechanic, your estimated profits are still:

$$(GCMV \cdot GCC + BCN \cdot BCC) - IC$$

(4000*0.7+2600*0.3)-3000= $3,580

Taking the car to the mechanic, your estimated profits become:

$$P(q+ \mid Pass) = 0.8421$$
$$P(q- \mid Pass) = 0.1579$$
$$P(q+ \mid Fail) = 0.4179$$
$$P(q- \mid Fail) = 0.5821$$

Expected Value of Test:

$$-100 + (1000 \cdot P(q+ \mid Pass) - 400 \cdot P(q- \mid Pass) + 1000 \cdot (q+ \mid Fail) - 400 \cdot (q- \mid Fail)) = 964$$

Final expected profit with the mechanic \$964.

Final expected profit without mechanic \$ 580.

7

To initiate the Value Iteration process, we used state utility values of 0 for all states except s3. We chose this approach because initializing a value iteration process requires a bit of guesswork and we only knew that s3 had a reward function of 1. With this said, we gave it the only initial value to speed the process along and have s3 start with a value around where we expected it to end up. For gamma, we chose the value of 0.1 because it incentives the agent to stay in states near s3, where the reward exists.

The results from running our implementation of Value Iteration were as follows:

```
0) [0, 0, 1, 0]

for state s1 best policy is a1
for state s2 best policy is a2
for state s3 best policy is a3
for state s4 best policy is a1
1 ) [0.0, 0.08000000000000002, 1.0, 0.09000000000000001]

for state s1 best policy is a2
for state s2 best policy is a2
for state s3 best policy is a3
for state s4 best policy is a1
2 ) [0.007200000000000015, 0.0816, 1.009, 0.09090000000000001]

for state s1 best policy is a2
for state s2 best policy is a2
for state s3 best policy is a3
for state s4 best policy is a1
3 ) [0.007416000000000001, 0.082352, 1.00909, 0.091719]

for state s1 best policy is a2
for state s2 best policy is a2
for state s3 best policy is a3
for state s4 best policy is a1
4 ) [0.007485840000000005, 0.08237424000000002, 1.0091719, 0.09173529000000001]

for state s1 best policy is a2
for state s2 best policy is a2
for state s3 best policy is a3
for state s4 best policy is a1
5 ) [0.007488540000000002, 0.0823812368, 1.009173529, 0.09174282389999999]

The final utilities are [0.007488540000000002, 0.0823812368, 1.009173529, 0.09174282389999999]
Iterations: 5
Time: 0.0059833526611328125
```

For our implementation, we iterated through each state and each action for each state to calculate a potential utility for each state that action would potentially take the agent to. After checking every state, we then calculate the change that every state underwent during that time step and compare the state that underwent the largest amount of change with the convergence criteria of 0.00001.

The optimum policy is shown in the 4 statements above the 5_{th} iteration in the figure above. The optimum utilities for each state is shown in the final array in the figure above.

Optimum Utilities: $s_1 = 0.0075, s_2 = 0.0824, s_3 = 1.0092, s_4 = 0.0917$

Optimal Policy: $s_1 = a_2, s_2 = a_2, s_3 = a_3, s_4 = a_1$

For our algorithm to conclude, it took 5 iterations and 0.00598 seconds.