Intro to AI Assignment 2A

Harrison Gropper, Justin M Hu, Adideb Nag

November 25, 2022

1 Question 1 - Bayesian Network

1.1 Simultaneously True

The probability that all five variables are true can be denoted as $P(A, B, C, D, E) = P(A) * P(B) * P(C) * P(D|A, B) * P(E|B, C)$

$= 0.20 * 0.50 * 0.80 * 0.10 * 0.30 = 0.0024$

1.2 Simultaneously False

The probability that all five variables are false can be denoted as $P(\neg A, \neg B, \neg C, \neg D, \neg E) = \neg P(A) * \neg P(B) * \neg P(C) * \neg P(D|A, B) * \neg P(E|B, C)$

$= 0.80 * 0.50 * 0.20 * 0.10 * 0.80 = 0.0064$

1.3 P(A) = False, All other Variables are True

The probability that A is false and the other four variables are false can be denoted as $P(\neg A, B, C, D, E) = \neg P(A) * P(B) * P(C) * P(D|A, B) * P(E|B, C)$, where α is $\frac{1}{P(\neg A,B,C,D,E)+P(A,B,C,D,E)} = \frac{1}{0.06}$

$= 0.80 * 0.50 * 0.80 * 0.60 * 0.30 * [\frac{1}{0.06}] = 0.96$

2 Question 2 - Variable Elimination

2.1 Variable Elimination

Let, B = burglary, E = earthquake, A = alarm, J = John calls, M = Mary calls

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

$= \alpha * \sum_E \sum_A P(B, E, A, J = T, m = T)$

$= \alpha * \sum_E \sum_A P(B) * P(E) * P(A|B, E) * P(J = T|A) * P(M = T|A)$

$= \alpha * P(B) \sum_E P(E) \sum_A P(A|B, E) * P(J = T|A) * P(M = T|A)$

$= \alpha < 0.00059224, 0.0014919 > \approx < 0.284, 0.716 >$, where $\alpha = 479.535324$

So our answer is $P(B|J = True, M = True) = 0.284$.

2.2 Operations

Arithmetically there are $5 + 1 + 2(1) + 4(3) = 20$ operations.

With a tree there would be a total of 15 operations by counting.

2.3 Complexity of Enumeration and Variable Elimination

The time complexity of enumeration is $O(2^n)$ since a depth first search is performed on the enumeration tree. The time complexity of variable elimination is $O(n)$ because the Bayesian network is a poly-tree and proper elimination order can be used to reduce time complexity.

3 Question 3 - Rejection Sampling and Likelihood Weighting

3.1 Computing Probabilities by Enumeration

$P(D|C) = \frac{P(\neg B)P(D|\neg B,C)+P(D|C,B)P(B)}{\sum_d P(D|C,B)P(B)+P(\neg B)P(D|\neg B,C)}$

$= \frac{0.10*0.50+0.90*0.75}{0.90+0.10}$

$= 0.725$.

$P(B|C) = \frac{P(B)P(C|\neg A,B)}{P(P(C|\neg A\neg B))P(\neg B)+P(B)P(C|\neg A,B)}$

$= \frac{0.90*0.50}{0.90*0.50+1*0}$

$= \frac{0.45}{0.45+0}$

$= 1$.

$P(D|\neg A, B) = \frac{P(\neg C|\neg A,B)P(D|\neg C,B)+P(C|\neg A,B)P(D|B,C)}{P(C|\neg A,B)+P(\neg C|\neg A,B)}$

$= 0.50 * 0.10 + 0.75 * 0.501$

$= 0.425$

3.2 Rejection Sampling and Likelihood Weighting

Approximating these values... Using code from file: Question3Simulation.py

I notice that only one of our probabilities is 100% accurate because its true probability is also 100%.

Because the other probabilities are not equal to 1, we have good approximations for them. But know we could get better approximations by using more samples.

1,000 Samples:

Results:

$P(B|C) = 1.0$

$P(D|C) = 0.7423887587822015$

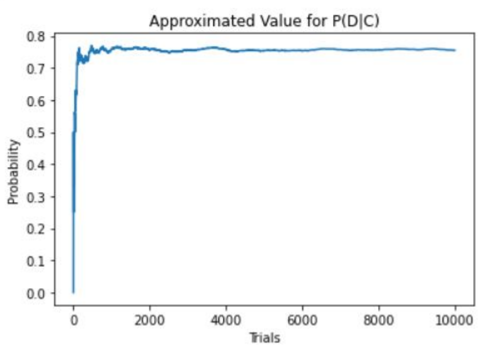
$P(D|\neg A, B) = 0.3986562150055991$

```
# Finding P(B|C)
numerator = 0
denominator = 0
for b,c in zip(probs["B"],probs["C"]):

    if b == c and c == True:
        numerator += 1
    if c == True:
        denominator += 1
print("P(B|C) = " + str(numerator/denominator))

# Finding P(D|C)
numerator = 0
denominator = 0
for d,c in zip(probs["D"],probs["C"]):

    if d == c and c == True:
        numerator += 1
    if c == True:
        denominator += 1
print("P(D|C) = " + str(numerator/denominator))

# Finding P(D|¬A,B)
numerator = 0
denominator = 0
for d,a,b in zip(probs["D"],probs["A"],probs["B"]):

    if d == True and a == False and b == True:
        numerator += 1
    if a == False and b == True:
        denominator += 1
print("P(D|¬A,B) = " + str(numerator/denominator))
```

```
P(B|C) = 1.0
P(D|C) = 0.7423887587822015
P(D|¬A,B) = 0.3986562150055991
```

3.3 Focusing on $P(D|C)$

```
plt.plot(to_graph)
plt.title("Approximated Value for P(D|C)")
plt.xlabel("Trials")
plt.ylabel("Probability")
plt.show()
```



As the number of trials increases we converge to our optimal value of 0.725. If we were run this experiment for an infinite amount of time, then we would exactly get 0.725.

3.4 Bayesian Network Query

Given Query: $P(C|\neg B)$

Rejection sampling is noticeably worse for this query because the evidence variable is of a low probability, leading to fewer samples being calculated and therefore a more inaccurate probability.

4 Question 4 - Interplanetary Search and Rescue Expert

4.1 Filtering

Initial State Distribution
We know the rover is guaranteed to start at position A, so $X_1 = A$.
Every other position has zero probability of being equal to X_1

X_1	$P(X_1)$
A	1
¬A	0

Transition Model
The rover cannot move backwards so we know the probabilities that include the given position cannot come before the non given position.
There is no way to get to position D, E, or F in three days
If we ever reach position C, then that means that is the third day and we cannot move back to C.

X_t	X_{t+1}	$P(X_{t+1}\|X_t)$
A	B	0.80
B	C	0.80
C	D	0.80
A	A	0.20
B	B	0.20

Sensor Model There are only three days that pass, so it is impossible to reach D, E, or F.
There exists one hot space and two cold spaces within a distance of three.

X_t	e_t	$P(e_t\|X_t)$
A	hot	1.00
A	cold	0.00
B	hot	0.00
B	cold	1.00
C	hot	0.00
C	cold	1.00

Because this is a filtering problem we can use the tables above and the following formula to calculate $P(X_3|E_1 = hot_1, E_2 = cold_2, E_3 = cold_3)$ for all six possible positions.
Filtering Formula: P$(X_t|e_{1:t}) = \alpha * P(e_t|X_t) \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1})$, where α is the normalizing constant.

Solving for $X_t = A$, where $t = 3$

$= \alpha * P(e_3 = cold|X_3 = A) \sum_{x_2} P(x_2 = A|e_{1:2} = hot_1, cold_2)$

$= \alpha * 0 \sum_{x_2} P(x_2 = A|e_{1:2} = hot_1, cold_2)$

$= 0.$

Solving for $X_t = B$, where $t = 3$

$= \alpha * P(e_3 = cold|X_3 = B) \sum_{x_2} P(x_2 = B|e_{1:2} = hot_1, cold_2)$

$= 0.80 * 0.20 * \alpha$

$= 0.80 * 0.20 * (\frac{1}{0.16+0.64})$

$= 0.80 * 0.20 * (\frac{1}{0.80})$

$= \frac{0.16}{0.80}$

$= 0.20.$

Solving for $X_t = C$, where $t = 3$

$= \alpha * P(e_3 = cold|X_3 = C) \sum_{x_2} P(x_2 = B|e_{1:2} = hot_1, cold_2)$

$= 0.80 * 0.80 * \alpha$

$= 0.80 * 0.80 * (\frac{1}{0.16+0.64})$

$= 0.80 * 0.80 * (\frac{1}{0.80})$

$= \frac{0.64}{0.80}$

$= 0.80.$

Solving for $X_t = \{$D, E, F$\}$, where $t = 3$
We know we cannot reach these positions because we are interested in the first three days. So all these probabilities are zero.

Our final answer is:

X_3	$P(X_3 \mid hot_1, cold_2, cold_3)$
A	0.00
B	0.20
C	0.80
D	0.00
E	0.00
F	0.00

4.2 Smoothing

Smoothing formula: $P(X_k|e_{1:t}) = \alpha P(X_k|e_{1:k})P(e_{k+1:t}|X_k)$

Solving for $P(X_2 = A|hot_1, cold_2, cold_3)$, so $k = 2$ and $t = 3$.

$= \alpha P(X_2 = A|hot_1, cold_2)P(cold_3|X_2 = A)$

$= \alpha * 0 * P(cold_3|X_2 = A)$

$= 0.$

Solving for $P(X_2 = B|hot_1, cold_2, cold_3)$, so $k = 2$ and $t = 3$.

$= \alpha P(X_2 = B|hot_1, cold_2)P(cold_3|X_2 = B)$

$= \alpha * 1 * 1$

$= \alpha$, where $\alpha = 1$

$= 1.$

We know that for all other positions where X_2 = {C, D, E, F} are not possible because we cannot reach those positions by the second day.

Our final answer is:

X_2	$P(X_2 \vert hot_1, cold_2, cold_3)$
A	0.00
B	1.00
C	0.00
D	0.00
E	0.00
F	0.00

4.3 Most Likely Explanation

For this problem we would like to compute the most likely explanation (argmax) for $P(X_3|hot_1, cold_2, cold_3)$.

We can rule out the positions (probability of zero) where X_3 = {D, E, F} because we cannot reach these positions in three days.

Let's calculate $X_3 = A$

$P(X_3 = A|hot_1, cold_2, cold_3)$

$= P(X_1 = A|hot_1) * P(X_2 = A|cold_2) * P(X_3 = A|cold_3)$

$= 1 * 0 * 0$

$= 0.$

Let's calculate $X_3 = B$

$P(X_3 = B|hot_1, cold_2, cold_3)$

$= P(X_1 = A|hot_1) * P(X_2 = B|cold_2) * P(X_3 = B|cold_3)$

$= 1 * 1 * 0.20$

$= 0.20$

Let's calculate $X_3 = C$

$P(X_3 = C|hot_1, cold_2, cold_3)$

$= P(X_1 = A|hot_1) * P(X_2 = B|cold_2) * P(X_3 = C|cold_3)$

$= 1 * 1 * 0.80$

$= 0.80$

Our final distribution is:

The maximum probability in this table is for position C.

X_3	$P(X_3 \vert hot_1, cold_2, cold_3)$
A	0.00
B	0.20
C	0.80
D	0.00
E	0.00
F	0.00

The sequence related to this probability is starting at position A, moving to B, then moving from B to C.

4.4 Prediction 1

We have to figure out $P(hot_4, hot_5, cold_6|hot_1, cold_2, cold_3)$.

Prediction Formula: $P(X_{t+k+1}|e_{1:t}) = \sum_{x_{t+k}} P(X_{t+k+1}|X_{t+k})P(X_{t+k}|e_{1:t})$

$P(X_{t+k+1}|e_{1:t}) = \sum_{x_{t+k}} P(X_{t+k+1}|X_{t+k})P(X_{t+k}|e_{1:t})$

$= P(X_2 = B|X_1 = A) * P(X_3 = C|X_2 = B) * P(X_4 = D|X_3 = C) * P(X_5 = D|X_4 = D) * P(X_6 = E|X_5 = D)$

$= 1 * P(X_3 = C|X_2 = B) * P(X_4 = D|X_3 = C) * P(X_5 = D|X_4 = D) * P(X_6 = E|X_5 = D)$

$= 1 * 0.80 * 0.80 * P(X_5 = D|X_4 = D) * P(X_6 = E|X_5 = D)$

$= 1 * 0.80 * 0.80 * 0.20 * P(X_6 = E|X_5 = D)$

$= 1 * 0.80 * 0.80 * 0.20 * 0.80$

$= (0.80)^3 * 0.20$

$= 0.1024.$

So, our answer is $P(hot_4, hot_5, cold_6|hot_1, cold_2, cold_3) = 0.1024$.

4.5 Prediction 2

We have to figure out the distribution for $P(X_4|hot_1, cold_2, cold_3)$.

We know there is no way we could get to positions {A, E, F}, so those probabilities are zero. Now we just have to figure out {B, C, D}.

Prediction Formula: $P(X_{t+k+1} = B|e_{1:t}) = \sum_{x_{t+k}} P(X_{t+k+1} = B|X_{t+k} = B)P(X_{t+k} = B|e_{1:t})$

$= P(X_2 = B|X_1 = A) * P(X_3 = B|X_2 = B) * P(X_4 = B|X_3 = B)$

$= 1 * P(X_3 = B|X_2 = B) * P(X_4 = B|X_3 = B)$

$= 1 * 0.20 * 0.20$

$= 0.04.$

Prediction Formula: $P(X_{t+k+1} = C|e_{1:t}) = \sum_{x_{t+k}} P(X_{t+k+1} = C|X_{t+k} = B)P(X_{t+k} = B|e_{1:t})$

$= \big[P(X_2 = B|X_1 = A) * P(X_3 = B|X_2 = B) * P(X_4 = C|X_3 = B)\big] + \big[P(X_2 = B|X_1 = A) * P(X_3 = C|X_2 = B) * P(X_4 = C|X_3 = C)\big]$

$= \big[1 * P(X_3 = B|X_2 = B) * P(X_4 = C|X_3 = B)\big] + \big[1 * P(X_3 = C|X_2 = B) * P(X_4 = C|X_3 = C)\big]$

$= \big[1 * 0.20 * P(X_4 = C|X_3 = B)\big] + \big[1 * 0.80 * P(X_4 = C|X_3 = C)\big]$

$= \big[1 * 0.20 * 0.80\big] + \big[1 * 0.80 * 0.20\big]$

$= 2 * 0.16$

$= 0.32.$

Prediction Formula: $P(X_{t+k+1} = D|e_{1:t}) = \sum_{x_{t+k}} P(X_{t+k+1} = D|X_{t+k} = B)P(X_{t+k} = C|e_{1:t})$

$= P(X_2 = B|X_1 = A) * P(X_3 = C|X_2 = B) * P(X_4 = D|X_3 = C)$

$= 1 * 0.80 * P(X_4 = D|X_3 = C)$

$= 1 * 0.80 * 0.80$

$= 0.64.$

Our final distribution looks like:

X_4	$P(X_4|hot_1, cold_2, cold_3)$
A	0.00
B	0.04
C	0.32
D	0.64
E	0.00
F	0.00

Now we have to figure out the distribution for $P(X_5|hot_1, cold_2, cold_3)$.

We know there is no way we could get to positions {A, F}, so those probabilities are zero. Now we just have to figure out {B, C, D, E}.

Conveniently we can use the above distribution to further build on our predictions for X_5.

Prediction Formula: $P(X_{t+k+1} = B|e_{1:t}) = \sum_{x_{t+k}} P(X_{t+k+1} = B|X_{t+k} = B)P(X_{t+k} = B|e_{1:t})$

$= P(X_4 = B|hot_1, cold_2, cold_3) * P(X_5 = B|X_4 = B)$

$= 0.04 * 0.20$

$= 0.008.$

Prediction Formula: $P(X_{t+k+1} = C|e_{1:t}) = \sum_{x_{t+k}} P(X_{t+k+1} = C|X_{t+k} = B)P(X_{t+k} = C|e_{1:t})$

$= \big[P(X_3 = B|X_2 = B) * P(X_4 = C|X_3 = B) * P(X_5 = C|X_4 = C)\big]+$
$\big[P(X_3 = C|X_2 = B) * P(X_4 = C|X_3 = C) * P(X_5 = C|X_4 = C)\big]+$
$\big[P(X_3 = B|X_2 = B) * P(X_4 = B|X_3 = B) * P(X_5 = C|X_4 = B)\big]$

$= \big[0.20 * 0.80 * 0.20\big]+$
$\big[0.80 * 0.20 * 0.20\big]+$

$\big[0.20 * 0.20 * 0.80)\big]$

$= 3 * 0.032$

$= 0.096$

Prediction Formula: $P(X_{t+k+1} = D|e_{1:t}) = \sum_{x_{t+k}} P(X_{t+k+1} = D|X_{t+k} = B)P(X_{t+k} = D|e_{1:t})$

$= \big[P(X_3 = B|X_2 = B) * P(X_4 = C|X_3 = B) * P(X_5 = D|X_4 = C)\big]+$
$\big[P(X_3 = C|X_2 = B) * P(X_4 = C|X_3 = C) * P(X_5 = D|X_4 = C)\big]+$
$\big[P(X_3 = C|X_2 = B) * P(X_4 = D|X_3 = C) * P(X_5 = D|X_4 = D)\big]$

$= \big[0.20 * 0.80 * 0.80]+$
$\big[0.80 * 0.20 * 0.80]+$
$\big[0.80 * 0.80 * 0.20)\big]$

$= 0.128 * 3$

$= 0.384$

Prediction Formula: $P(X_{t+k+1} = E|e_{1:t}) = \sum_{x_{t+k}} P(X_{t+k+1} = E|X_{t+k} = B)P(X_{t+k} = E|e_{1:t})$

$= P(X_3 = C|X_2 = B) * P(X_4 = D|X_3 = C) * P(X_5 = E|X_4 = D)$

$= 0.80 * 0.80 * 0.80$

$= 0.512$

Our final distribution looks like:

X_5	$P(X_5|hot_1, cold_2, cold_3)$
A	0.00
B	0.008
C	0.096
D	0.384
E	0.512
F	0.00

5 Question 5 - Unknown Cells

5.1 Computing Probability

We have computed the probabilities of the four graphs with a simulation.

```
# given actions and sensor readings
actions = ["Right", "Right", "Down", "Down"]
sensor_readings = ["N","N","H","H"]
allowable_start_positions = [(0,0),(0,1),(0,2),
                             (1,0),(1,1),(1,2),
                             (2,0),(2,2)]
our_world = [["H", "H", "T"],
             ["N", "N", "N"],
             ["N", "B", "H"]]
```

Below is some of the code we used to calculate these probabilities.

```
# we have to keep track of the probabilites
if graph_number == 1 and sensor_readings == ["N"]:

    # incrememnting trials and position place
    trials["Graph 1"] += 1
    probabilities["Graph 1"][(pos_y,pos_x)] += 1

elif graph_number == 2 and sensor_readings == ["N","N"]:

    # incrememnting trials and position place
    trials["Graph 2"] += 1
    probabilities["Graph 2"][(pos_y,pos_x)] += 1

elif graph_number == 3 and sensor_readings == ["N","N","H"]:

    # incrememnting trials and position place
    trials["Graph 3"] += 1
    probabilities["Graph 3"][(pos_y,pos_x)] += 1

elif graph_number == 4 and sensor_readings == ["N","N","H","H"]:

    # incrememnting trials and position place
    trials["Graph 4"] += 1
    probabilities["Graph 4"][(pos_y,pos_x)] += 1
```

To make sure the code was working we made sure that all calculated probabilities sum to 1 following Kolmogorov's axioms.

```
# confirming that all probabilities sum to 1 in each graph
for g in ["Graph 1","Graph 2","Graph 3","Graph 4"]:
    ii = 0
    for i in probabilities[g]:
        ii += probabilities[g][i]
    print(ii, g)
```

```
1.0 Graph 1
1.0 Graph 2
1.0 Graph 3
1.0 Graph 4
```

The rest of the code is located in a file called Question5Simulation.py

Graph 1: actions = {Right}, sensor readings = {N}

0.00129	0.01322	0.02510
0.02370	0.23689	0.44972
0.23696	0.00000	0.01309

Graph 2: actions = {Right, Right}, sensor readings = {N, N}

0.000008	0.00015	0.00214
0.00250	0.04738	0.04738
0.04738	0.00000	0.00078

Graph 3: actions = {Right, Right, Down}, sensor readings = {N, N, H}

0.000001	0.00002	0.00001
0.00001	0.00408	0.00614
0.02165	0.00000	0.96805

Graph 4: actions = {Right, Right, Down, Down}, sensor readings = {N, N, H, H}

0.00000	0.000002	0.00000
0.0000004	0.00023	0.0000004
0.00124	0.00000	0.99848

Our findings are approximate values because we used the simulation to calculate these probabilities over 10 million trials.
Also, the following probabilities make sense throughout the four graphs.
The more moves and information we have with our sensor, the more certain we are of our location.

5.2 Scaling Up

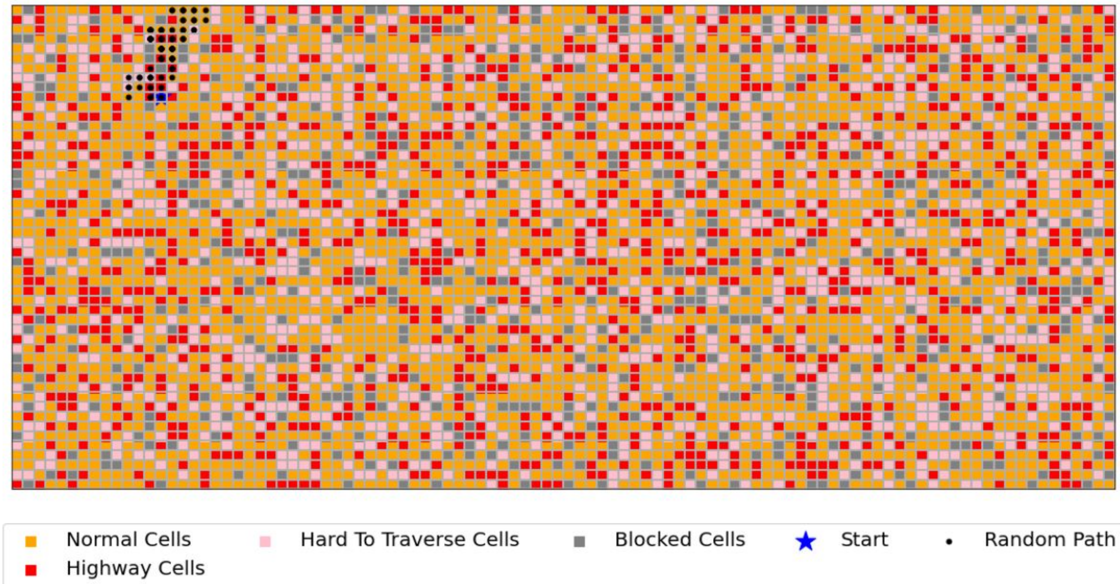
We have used matplotlib to make our graph with different cell types.

Cell types used:

- Normal Cells - orange
- Hard To Traverse Cells - pink
- Blocked Cells - gray
- Highway cells - red

Legend:

- Random Path - black dots
- Start - blue star



The graph above corresponds exactly to groundTruth10.txt in the folder map 10.
Below are some values you will find in that file.

X-position	Y-position	Observations	Action Type
9	13	H	L
9	12	H	D
9	12	H	U
8	12	N	U
7	12	H	U
6	12	T	D
7	12	T	L