Name: Sidhu Arakkal, Eoin O'Hare, Mohammed Awais Zubair
Date: 12/12/2022
Fall 2022

Assignment 2 (CS 440)

Final Submission

---

1. Bayesian Network

   (a) Allow us to start by restating the question at hand: $P(A, B, C, D, E)$

   Using Joint Probability Distribution we can deduct that we must somehow find the probability for D and E, otherwise known as: $P(D)$ and $P(E)$. In order to do this we take a look at the given values and their corresponding probabilities. Thus, we know that we have $P(A = true)$, $P(B = true)$, and $P(C = true)$. Using this information and the conditional probabilistic tables we can find that $P(D|A, B) = 0.1$ and $P(E|B, C) = 0.3$. We combine this with the known values $P(A) = 0.2$, $P(B) = 0.5$, and $P(C) = 0.8$.

   Thus, leaving us with:

   $P(A, B, C, D, E) = P(D|A, B) * P(E|B, C) * P(A) * P(B) * P(C) = (0.1) * (0.3) * (0.2) * (0.5) * (0.8) = \mathbf{0.0024}$

   (b) We can once again use Joint Probability Distribution to find $P(\neg A, \neg B, \neg C, \neg D, \neg E)$

   We can employ the same strategy and find the values of $P(\neg D|\neg A, \neg B)$ and $P(\neg E||\neg B, \neg C)$, except this time we look for the values that correspond to false for A, B, and C for each respective section in the truth table. This value, however, must be subtracted by 1 in order to get the negation value for both D and E. Thus, we get $P(\neg D|\neg A, \neg B) = 1 - 0.9 = 0.1$ and $P(\neg E||\neg B, \neg C) = 1 - 0.2 = 0.8$

   Similarly, we can find the total value by taking our known values for the probability of A, B, and C and subtracting by them by 1 in order to get the negated values. Here we get: $P(\neg A) = 1 - 0.2 = 0.8$, $P(\neg B) = 1 - 0.5 = 0.5$, and $P(\neg C) = 1 - 0.8 = 0.2$.

   Thus, $P(\neg A, \neg B, \neg C, \neg D, \neg E) = P(\neg D|\neg A, \neg B) * P(\neg E||\neg B, \neg C) * P(\neg A) * P(\neg B) * P(\neg C) = (0.1) * (0.8) * (0.8) * (0.5) * (0.2) = \mathbf{0.0064}$

   (c) Allow us to use summation rules to find $P(\neg A|B, C, D, E)$

   We know that the above probability must equal $P(\neg A, B, C, D, E)$ and this equation can be split up into the following: $P(\neg A, B, C, D, E) = P(\neg A) * P(B) * P(C) * \sum_D P(D|A, B) * \sum_E P(E|B, C)$

   Since we know that B and C are set to be True, $P(B) = 0.5$ and $P(C) = 0.8$ because they correspond to their given values. Next, we know that $A = false$ and $B = true$, so we can see as shown above that $P(D)$ corresponds to $P(D|\neg A, B)$ which is 0.6 according to the truth table. Similarly, we must find the the matching probability for E given that B and C are true, otherwise known as $P(E|B, C) = 0.3$ given through the truth tables. Lastly, we know that that negation of A must be 1 - its true value, which gives us $P(\neg A) = 1 - 0.2 = 0.8$

Combining all this we arrive at:

$P(\neg A|B, C, D, E) = (0.8) * (0.5) * (0.8) * (0.6) * (0.3) = \mathbf{0.0576}$

2. Variable Elimination Algorithm

(a)

$$P(B|j,m) = \alpha * P(B) * \sum_e P(e) \sum_a P(a|b,e)P(j|a)P(m|a)$$

$$f_1(B) = \begin{bmatrix} P(b) \\ P(\neg b) \end{bmatrix} = \begin{bmatrix} 0.001 \\ 0.999 \end{bmatrix}$$

$$f_2(A) = \begin{bmatrix} P(e) \\ P(\neg e) \end{bmatrix} = \begin{bmatrix} 0.002 \\ 0.998 \end{bmatrix}$$

$$f_3(A,B,E) = \begin{bmatrix} f_3(A,B,e) \\ f_3(A,B,\neg e) \end{bmatrix}$$

$$f_3(A,B,e) = \begin{bmatrix} P(a|b,e) & P(a|\neg b,e) \\ P(\neg a|b,e) & P(\neg a|\neg b,e) \end{bmatrix} = \begin{bmatrix} 0.95 & 0.29 \\ 0.94 & 0.001 \end{bmatrix}$$

$$f_3(A,B,\neg e) = \begin{bmatrix} P(a|b,\neg e) & P(a|\neg b,\neg e) \\ P(\neg a|b,\neg e) & P(\neg a|\neg b,\neg e) \end{bmatrix} = \begin{bmatrix} 0.05 & 0.71 \\ 0.06 & 0.999 \end{bmatrix}$$

$$f_4(A) = \begin{bmatrix} P(j|a) \\ P(j|\neg a) \end{bmatrix} = 0.90$$

$$f_5(A) = \begin{bmatrix} P(m|a) \\ P(m|\neg a) \end{bmatrix} = 0.70$$

$$P(B|j,m) = \alpha * f_1(B) * \sum_e f_2(E) \sum_a f_3(A,B,E) f_4(A) f_5(A)$$

$$f_6(B,E) = \sum_a f_3(A,B,E) \times f_4(A) \times f_5(A) = (f_3(a,B,E) \times f_4(a) \times f_5(a)) + (f_3(\neg a,B,E) \times f_4(\neg a) \times f_5(\neg a))$$

$$f_6(B,E) = (0.90 * 0.70 * \begin{bmatrix} 0.95 & 0.29 \\ 0.94 & 0.001 \end{bmatrix}) + (0.05 * 0.01 * \begin{bmatrix} 0.05 & 0.71 \\ 0.06 & 0.999 \end{bmatrix}) = \begin{bmatrix} 0.598525 & 0.183055 \\ 0.59223 & 0.0011295 \end{bmatrix})$$

$$P(B|j,m) = \alpha * f_1(B) * \sum_e f_2(E) * f_6(B,E)$$

$$f_7(B) = \sum_e f_2(E) * f_6(B,E) = (f_2(e) \times f_6(B,e)) + (f_2(\neg e) \times (f_6(B,\neg e))$$

$$f_7(B) = (0.002 * \begin{bmatrix} 0.598525 \\ 0.183055 \end{bmatrix}) + (0.998 * \begin{bmatrix} 0.59223 \\ 0.0011295 \end{bmatrix}) = \begin{bmatrix} 0.592243 \\ 0.001493 \end{bmatrix}$$

$$P(B|j,m) = \alpha * f_1(B) * f_7(B) = \alpha * \begin{bmatrix} 0.001 \\ 0.999 \end{bmatrix} * \begin{bmatrix} 0.59223 \\ 0.001492 \end{bmatrix}$$

$$P(B|j,m) = \alpha * \begin{bmatrix} 0.001 \\ 0.999 \end{bmatrix} * \begin{bmatrix} 0.59223 \\ 0.001492 \end{bmatrix} = \alpha * \begin{bmatrix} 0.0005923 \\ 0.001492 \end{bmatrix}$$

$$P(B|j,m) = \alpha * \begin{bmatrix} 0.0005923 \\ 0.001492 \end{bmatrix} = (0.0005923/(0.0005923+0.001492)), (0.001492/(0.0005923+0.001492))$$

$$P(B|j,m) = < 0.284, 0.716 >$$

(b) For this problem, we calculated the number of arithmetic operations by counting the number of terms in each operation. In order to calculate f_6, you have to multiple 2 scalars with a 2x2 matrix twice to include both a and $\neg a$ to sum out that variable. Therefore, to calculate f_6, there are 2*(3*4)=24 multiplication operations and 2*4 = 8 addition operations to add the resulting 2x2 matrices together. To calculate f_7, you have to multiple one scalar with a 1x2 matrix twice to include both e and $\neg e$ to sum it out. Therefore, there are 2*(2*2) = 8 multiplication operations and 2*2 addition operations to add the resulting 1x2 matrices together. Next, you have to multiple the resulting 1x2 matrix by a scalar, through 2*2 multiplication operations. Finally, you have to normalize the 1x2 matrix to get the probabilities, which requires 2 addition operations and 4 division operations, (each term has to be divided by the sum of the matrix). Therefore, variable elimination of this query took 4 division operations, 36 multiplication operations, and 16 addition operations for a total 56 operations. As discussed

in lecture, tree enumeration of this query requires 40 operations. Therefore, variable eliminate requires 16 more operations.

(c) In Bayesian networks that have the form of a chain, the time complexity of variable elimination with the query P($X_1|X_N = True$)) would be O(n^2). The space complexity would be O(n). The time complexity of tree enumeration with the query P($X_1|X_N = True$)) would be O(n^2) and the space complexity would be O(n).

3. Bayesian Network Sampling Techniques

(a)

$$P(d|c)$$

$$P(d|c) = \alpha * \sum_{e}\sum_{a} P(A)P(B)P(c|A,B)P(d|B,c)$$

$$P(d|c) = P(a)P(b)P(c|a,b)P(d|b,c) + P(\neg a)P(b)P(c|\neg a,b)P(d|b,c)$$

$$+P(a)P(\neg b)P(c|a,\neg b)P(d|\neg b,c) + P(\neg a)P(\neg b)P(c|\neg a,\neg b)P(d|\neg b,c)$$

$$P(a) = 0$$

$$P(c|\ a,\ b) = 0$$

$$P(d|c) = P(\neg a)P(b)P(c|\neg a,b)P(d|b,c)$$

$$P(d|c) = 1 * 0.9 * 0.5 * 0.75 = 0.3375$$

$$P(\neg d|c)$$

$$P(d|c) = \alpha * \sum_{e}\sum_{a} P(A)P(B)P(c|A,B)P(\neg d|B,c)$$

$$P(\neg d|c) = P(a)P(b)P(c|a,b)P(\neg d|b,c) + P(\neg a)P(b)P(c|\neg a,b)P(\neg d|b,c)$$

$$+P(a)P(\neg b)P(c|a,\neg b)P(\neg d|\neg b,c) + P(\neg a)P(\neg b)P(c|\neg a,\neg b)P(\neg d|\neg b,c)$$

$$P(a) = 0$$

$$P(c|\ a,\ b) = 0$$

$$P(\neg d|c) = P(\neg a)P(b)P(c|\neg a,b)P(\neg d|b,c)$$

$$P(d|c) = 1 * 0.9 * 0.5 * 0.25 = 0.1125$$

$$P(D|c) = \alpha * \begin{bmatrix} 0.3375 \\ 0.1125 \end{bmatrix}$$

$$P(D|c) = \begin{bmatrix} 0.75 \\ 0.25 \end{bmatrix}$$

$$P(d|c) = 0.75$$

$$P(b|c)$$

$$P(b|c) = \alpha * \sum_{D}\sum_{A} P(A)P(b)P(c|A,b)P(D|b,c)$$

$$P(b|c) = P(a)P(b)P(c|a,b)P(d|b,c) + P(\neg a)P(b)P(c|\neg a,b)P(d|b,c)$$

$$+P(a)P(b)P(c|a,b)P(\neg d|b,c) + P(\neg a)P(b)P(c|\neg a,b)P(\neg d|b,c)$$

$$P(a) = 0$$

$$P(d|c) = P(\neg a)P(b)P(c|\neg a,b)P(d|b,c) + P(\neg a)P(b)P(c|\neg a,b)P(\neg d|b,c)$$

$$P(d|c) = 1 * 0.9 * 0.5 * 0.75 + 1 * 0.9 * 0.5 * 0.25 = 0.45$$

$$P(\neg b|c)$$

$$P(\neg b|c) = \alpha * \sum_D \sum_A P(A)P(\neg b)P(c|A, \neg b)P(D|\neg b, c)$$

$$P(\neg b|c) = P(a)P(\neg b)P(c|a, \neg b)P(d|\neg b, c) + P(\neg a)P(\neg b)P(c|\neg a, \neg b)P(d|\neg b, c)$$

$$+P(a)P(\neg b)P(c|a, \neg b)P(\neg d|\neg b, c) + P(\neg a)P(\neg b)P(c|\neg a, \neg b)P(\neg d|\neg b, c)$$

$$P(a) = 0$$

$$P(c|\neg a, \neg b] = 0$$

$$P(\neg d|c) = 0$$

$$P(D|c) = \alpha * \begin{bmatrix} 0.45 \\ 0 \end{bmatrix}$$

$$P(D|c) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$P(b|c) = 1$$

$$P(d|\neg a, b)$$

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

$$P(d|\neg a, b) = P(\neg a)P(b)P(c|\neg a, b)P(d|b, c) + P(\neg a)P(b)P(\neg c|\neg a, b)P(d|b, \neg c)$$

$$d|\neg a, b) = 1 * 0.90 * 0.5 * 0.75 + 1 * 0.90 * 0.5 * 0.1 = 0.3825$$

$$P(\neg d|\neg a, b)$$

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

$$P(\neg d|\neg a, b) = P(\neg a)P(b)P(c|\neg a, b)P(\neg d|b, c) + P(\neg a)P(b)P(\neg c|\neg a, b)P(\neg d|b, \neg c)$$

$$\neg d|\neg a, b) = 1 * 0.90 * 0.5 * 0.25 + 1 * 0.90 * 0.5 * 0.9 = 0.5175$$

$$P(D|\neg a, b) = \alpha * \begin{bmatrix} 0.3825 \\ 0.5175 \end{bmatrix}$$

$$P(D|c) = \begin{bmatrix} 0.425 \\ 0.575 \end{bmatrix}$$

$$P(d|\neg a, b) = 0.425$$

(b) With rejection sampling and using a 1000 sample size, the results were [0.76484,0.23516], [1.0,0.0], and [0.41414,0.58586] for P($d|c$), P($b|c$), and P($d|\neg a, b$) respectively. The approximations by the rejection sampling were very close to our exact calculations using enumeration. The approximate probabilities of P($d|c$), P($b|c$), and P($d|\neg a, b$) had an error percentage of 1.97 percent, 0 percent, and 2.56 percent respectively. With likelihood sampling and using a 1000

sample size, the results were [0.75558,0.24442], [1.0,0.0], and [0.426,0.574] for P(($d|c$), P($b|c$), and P($d|\neg a, b$) respectively. The approximations by the rejection sampling were very close to our exact calculations using enumeration. The approximate probabilities of P(($d|c$), P($b|c$), and P($d|\neg a, b$) had an error percentage of 0.744 percent, 0 percent, and 0.235 percent respectively.

(c) For our plots of P($d|c$) using the two methods against the number of samples, we first started with 500 samples and 1000 samples, and then increased the number of samples by 1000 until reaching 25000 samples. The plot of the probability found by rejection sampling against the number of samples is given below:

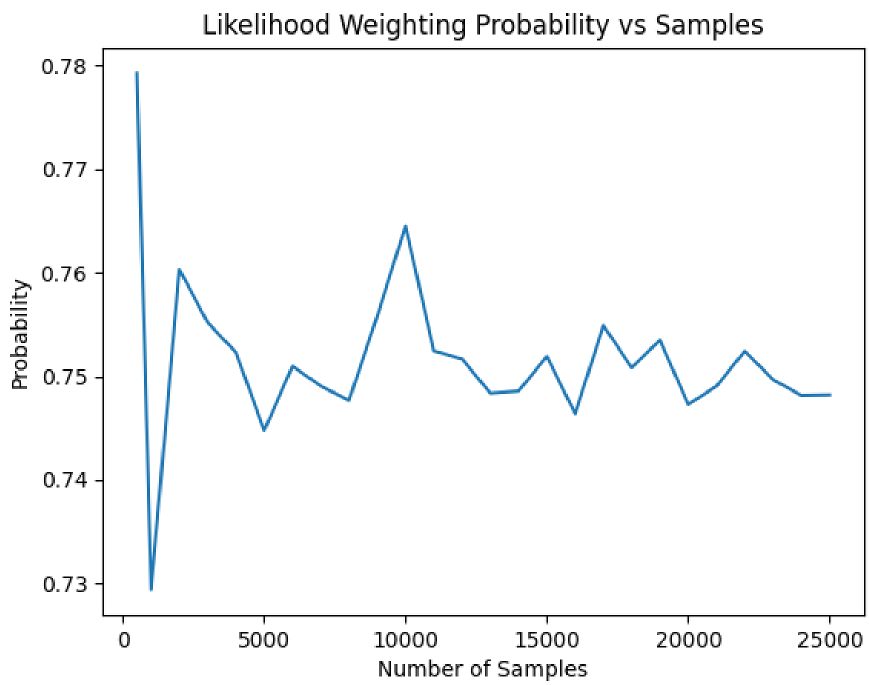

Figure 1: Rejection Sampling Probability vs Samples

The probability of P($d|c$) begins to converge between around 0.755 and 0.745, but still has fluctuation. The high and low peaks of the probability begin to become lower. Increasing the number of samples has a lower convergence rate than increasing samples with likelihood weighting has. The plot of the probability found by likelihood against the number of samples is given below:

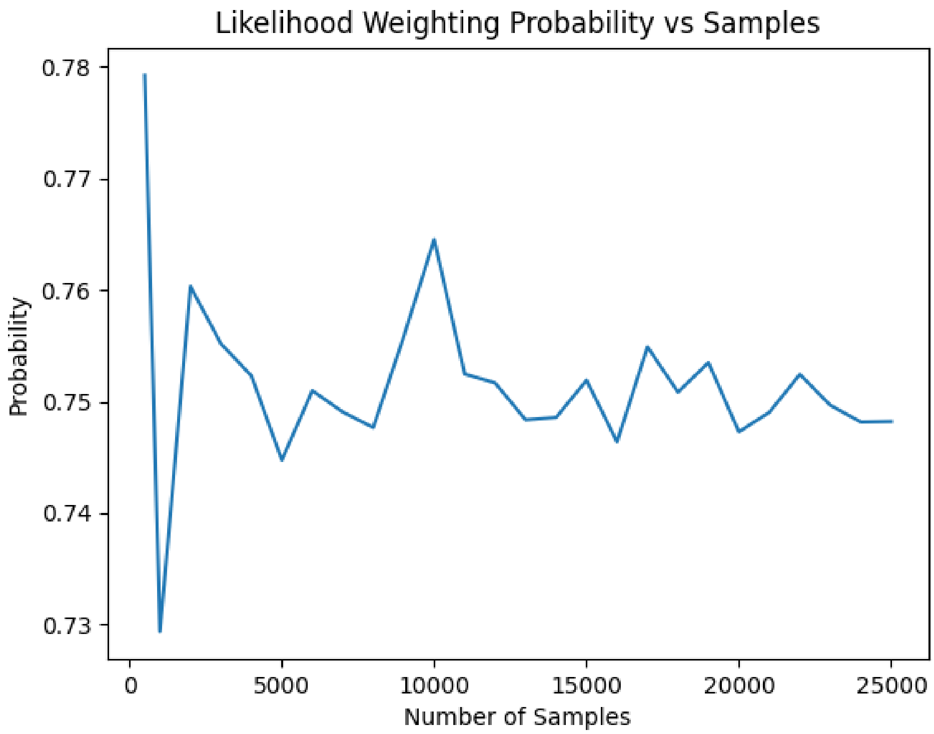

Figure 2: Likelihood Weighting Probability vs Samples

Increasing the number of samples causes the probability of this queue to converge around 0.75 with more accuracy. Therefore, these plots demonstrate how likelihood weighting is more efficient and more accurate with more samples, as many samples are not wasted.

(d) The query we decided to use was P$d|\neg b$. We chose this query as the probability of $\neg b$ is 0.1, so a lot of the samples will be wasted in the rejection sampling. The plot of the rejection sampling and the plot of likelihood sampling are shown below:

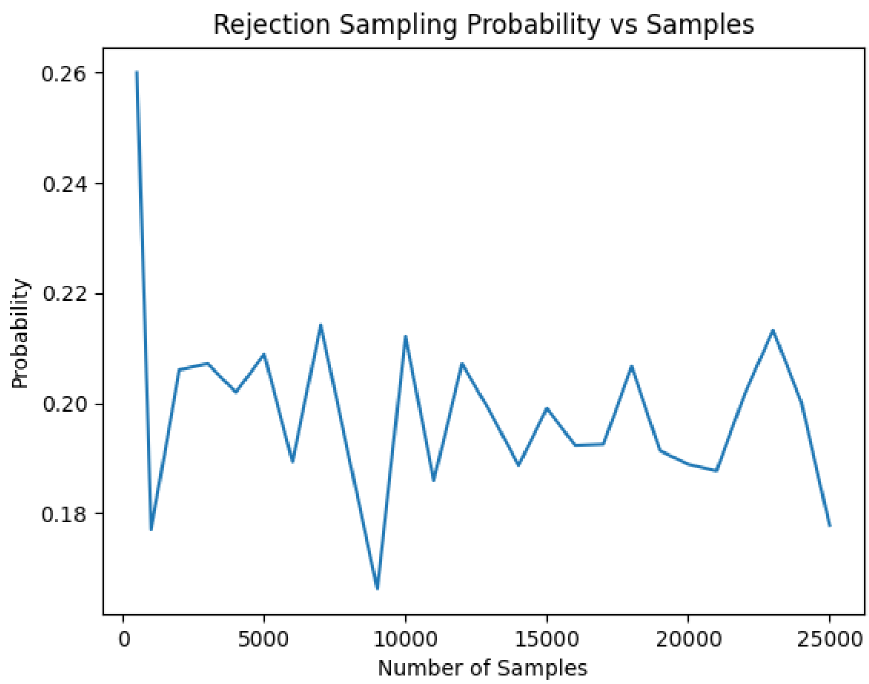

Figure 3: Rejection Sampling Probability vs Samples

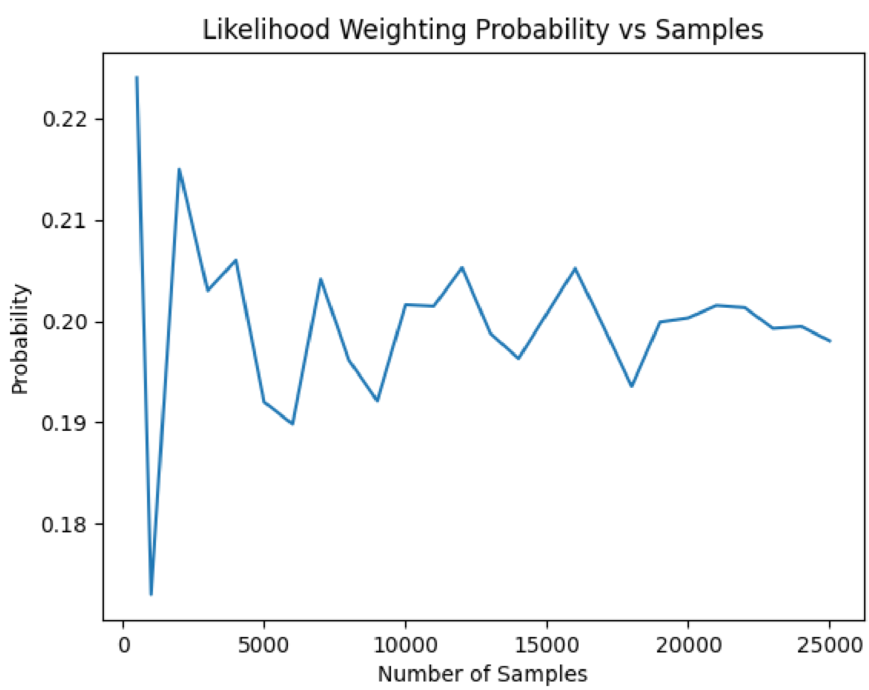

Figure 4: Likelihood Weighting Probability vs Samples

As seen in the plots, the convergence and effectiveness of the rejection sampling is considerably lower than that of the likelihood weighting. Around 20000 samples, the likelihood weighting converges around 0.20 with minimal fluctuation. However, the rejection sampling has a big spike of probability from 0.21 to around 0.18 after 20000 samples. The rejection sampling is noticeably worse for this query due to the probability of b being false being so low. The expected value of samples with b being false with 25000 samples is 2500. With this query, there is a lot of waste with rejection sampling, impacting the convergence and accuracy.

4. Mercury Rover Rescue

(a) In order to find $P(X_3|hot_1, cold_2, hot_3)$ we must use filtering:

Since we know the position $X_1 = A$, we also know that $P(X_1|hot_1) = 1$ must be true. Thus, $P(X_3|hot_1, cold_2) = \alpha * P(cold_2|X2) * \sum_{X_1} P(X_2|X_1)P(X1|hot_1)$

This comes to: $P(X_3|hot_1, cold_2) = \alpha(0,\ 1,\ 1,\ 0,\ 1,\ 1)$ (0.2, 0.8, 0, 0, 0, 0) = α (0, 0.8, 0, 0, 0, 0) = (0, 1, 0, 0, 0, 0). Next we do the same thing for a different set of evidence: $P(X_3|hot_1, cold_2, cold_3) = \alpha * P(cold_3|X3) * \sum_{X_2} P(X_3|X_2)P(X_2|cold_2)$

We can simplify the summation as shown below:
$\sum_{X_2} P(X_3|X_2)P(X_2|cold_2)$ = [((0, 0.2, 0.8, 0, 0, 0)*1) + ((0.2, 0.8, 0, 0, 0, 0)*0) + ((0, 0, 0.2, 0.8, 0, 0)*0) + ...] = (0, 0.2, 0.8, 0, 0, 0)

Lastly, if we combine everything:
$P(X_3|hot_1, cold_2, cold_3) = \alpha P(cold_3|X_3)$ (0, 0.2, 0.8, 0, 0, 0) = α (0, 0.2, 0.8, 0, 0, 0) * (0, 0.2, 0.8, 0, 0, 0) = α (0, 0.2, 0.8, 0, 0, 0) = (0, 0.2, 0.8, 0, 0, 0)

Thus, on X_3 or the third day the rover has a **0.2 probability** of being at position **B** and a **0.8 probability** of being at position **C**. This makes sense because it shows that as we move from A to B or X_1 to X_1, we go from a hot to cold position since a cold position always follows a hot one.

(b) We can find $P(X_2|hot_1, cold_2, cold_3)$ through smoothing by splitting the above:

$P(X_2|hot_1, cold_2, cold_3) = \alpha\ P(X_2|hot_1, cold_2)\ P(cold_3|X_2) = \alpha$ (0,1,0,0,0,0) x $P(cold_3|X_2)$

In order to solve this we need to find $P(cold_3|X_2)$ with summation:
$P(cold_3|X_2) = \sum_{X_3} P(cold_3|X_3)P(X_3|X_2)$ = ((0)*(1)*(0.2,0,0,0,0,0)) + ((1)*(1)*(0.8,0.2,0,0,0,0)) + ((1)*(1)*(0,0.8,0.2,0,0,0)) + ((0)*(1)*(0,0,0.8,0.2,0,0)) + ((1)*(1)*(0,0,0,0.8,0.2,0)) + ((1)*(1)*(0,0,0,0,0.8
= (0.8, 1, 0.2, 0.8, 1, 0.2)

Now allow us to plug this into the equation stated above:
α (0,1,0,0,0,0) x $P(cold_3|X_2) = \alpha$ (0,1,0,0,0,0) x (0.8, 1, 0.2, 0.8, 1, 0.2) = α (0, 1, 0, 0, 0, 0) = **(0,1,0,0,0,0)**

Thus, the above probability distribution over all six days makes sense because for a hot position to be followed by two cold positions the rover must be at position B on day 2.

(c) In order to find the Most Likely Explanation we must evaluate or given evidence. Our first piece of evidence is $E_1 = hot$ and this eludes to the fact on day 1 the position of the rover was at the first volcanic vent (location A). There is no other Most likely explanation for this evidence, so we get **Day 1 = Location A**. Computationally this can be shown below:
$V_1 = P(X_1) =$ (1,0,0,0,0,0)

The next piece of evidence is that $E_2 = cold$:
This is a vague piece of evidence as there are a couple different places the rover can be with

this evidence. However, we know V_1, so we can use that and a max value function to find the next V_2 as shown below:
$V_2(X) = max[V_1(X_1) * P(X_2|X_1) * P(E_2 = cold|X_2)$

From previous parts we know that the the max value for position B is 0.8, thus the following is true:
$V_2(X) = max$((1*0.2*0), (1*0.8*1), ...) max(0,0.8,0,0,0,0) = 0.8
This shows that this piece of evidence is most likely to result with:
Day 2 = Location B

Lastly, the third piece of evidence, $E_3 = cold$, can be evaluated in a similar manner to the previous one. Using the following:
$V_3(X) = max[V_2(X_2) * P(X_3|X_2) * P(E_3 = cold|X_3)$

Since we know that location B cannot exceed (0.2*0.8*1) and position C cannot exceed (0.8*0.8*1) from previous deduction, we get the following:
$V_3(X) = max$(0, 0.16, 0.64, 0, 0, 0)

This reveals the most likely location of the rover with this evidence would be:
Day 3 = Location C

(d) In order to find $P(hot_4, hot_5, cold_6|hot_1, cold_2, cold_3)$ using prediction we must use our previous information of our evidence found on days 1, 2, and 3. We know that the rovers position will reside at either location B or C because of the distribution done in the last step. Now allow us to predict the most likely paths to be taken by the rover given this information. The main path's we will at are the ones that involve C as the starting location on Day 3 which are the following:

i. Path = DDD and Probability = α0.08
ii. Path = DDE and Probability = α0.032
iii. Path = DEE and Probability = α0.032
iv. Path = DEF and Probability = α0.128

Of the above options, the one that makes the most sense in terms of the prediction algorithm would be the DDE path. Here DDE would correspond to $hot_4, hot_5, cold_6$ respectively. Lastly, we are looking for a single value instead of a distribution, so we can arrive at $P(hot_4, hot_5, cold_6|hot_1, cold_2, cold_3) = \alpha 0.032 = \mathbf{0.0533}$, where $\alpha = 1.66$

(e) One way to use prediction to solve $P(X_4, X_5|hot_1, hot_2, hot_3)$ would be to split up the probability distribution and find the sum.

We can start by finding:
$P(X_4|hot_1, hot_2, hot_3) = \sum_{X_3} P(X_3|hot_1, cold_2, cold_3)P(X_4|X_3)$

This evaluates out to:
$P(X_4|hot_1, hot_2, hot_3)$ = (0*(0.2, 0.8, 0, 0, 0, 0) + (0.2*(0, 0.2, 0.8, 0, 0, 0)) + (0.8*(0, 0, 0.2, 0.8, 0, 0)) + (0*(0, 0, 0, 0.2, 0.8, 0)) + (0*(0, 0, 0, 0, 0.2, 0.8)) + (0*(0, 0, 0, 0, 0, 0.2)) = **(0, 0.04, 0.32, 0.64, 0, 0)**

Furthermore, we can extend this methodology for $P(X_5)$:
$P(X_5|hot_1, hot_2, hot_3) = \sum_{X_4} P(X_4|hot_1, cold_2, cold_3)P(X_5|X_4)$

Which evaluates to:
$P(X_5|hot_1, hot_2, hot_3)$ = (0*(0.2, 0.8, 0, 0, 0, 0) + (0.04*(0, 0.2, 0.8, 0, 0, 0)) + (0.32*(0, 0, 0.2, 0.8, 0, 0)) + (0.64*(0, 0, 0, 0.2, 0.8, 0)) + (0*(0, 0, 0, 0, 0.2, 0.8)) + (0*(0, 0, 0, 0, 0, 0.2))
= **(0, 0.008, 0.096, 0.384, 0.512, 0)**

5. Programming Question Analysis

   (a) The steps below outline how to compute the probability of where you are located within the grid given the actions and sensor readings.

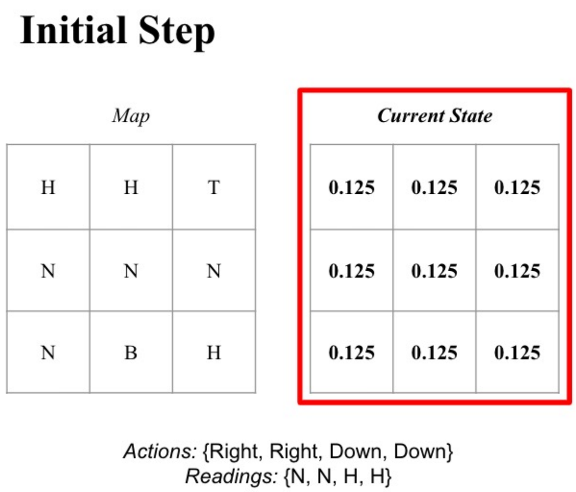

Figure 5: Problem Setup

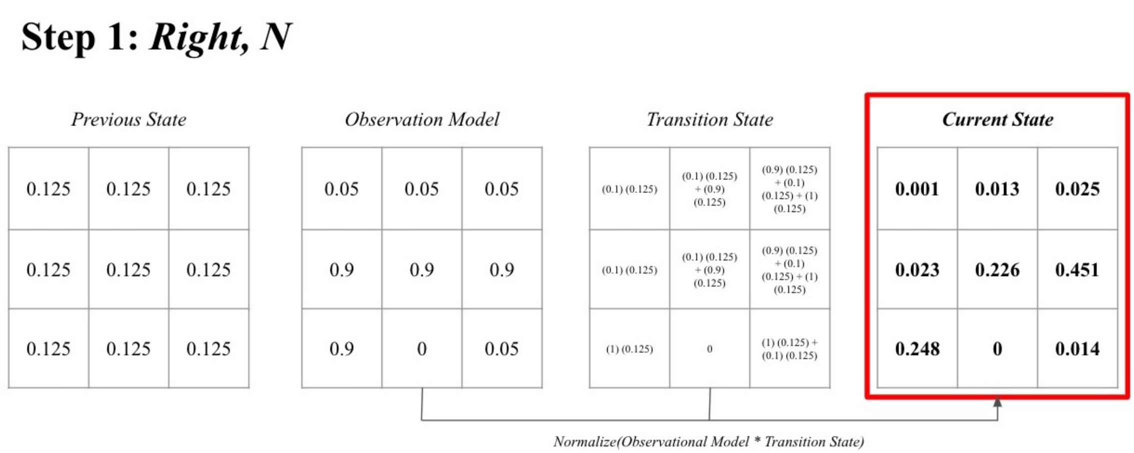

Figure 6: Right-N Action/Sensor Pair

Step 2: *Right, N*

Previous State

0.001	0.013	0.025
0.023	0.226	0.451
0.248	0	0.014

Observation Model

0.05	0.05	0.05
0.9	0.9	0.9
0.9	0	0.05

Transition State

(0.1) (0.001)	(0.1) (0.013) + (0.9) (0.001)	(0.9) (0.013) + (0.1) (0.025) + (1) (0.025)
(0.1) (0.023)	(0.1) (0.226) + (0.9) (0.023)	(0.9) (0.226) + (0.1) (0.451) + (1) (0.451)
(1) (0.248) + (0.1) (0.248)	0	(1) (0.014) + (0.1) (0.014)

Current State

0	**0**	**0.002**
0.002	**0.042**	**0.685**
0.267	**0**	**0.001**

*Normalize(Observational Model * Transition State)*

Figure 7: Right-N Action/Sensor Pair

Step 3: *Down, H*

Previous State

0	0	0.002
0.002	0.042	0.685
0.267	0	0.001

Observation Model

0.9	0.9	0.05
0.05	0.05	0.05
0.05	0	0.9

Transition State

0	0	(0.1) (0.002)
(0.1) (0.002)	(0.1) (0.042) + (1) (0.042)	(0.9) (0.02) + (0.1) (0.685)
(1) (0.267) + (0.9) (0.023) + (0.1) (0.267)	0	(1) (0.001) + (0.9) (0.685) + (0.1) (0.001)

Current State

0	**0**	**0**
0	**0.004**	**0.006**
0.027	**0**	**0.963**

*Normalize(Observational Model * Transition State)*

Figure 8: Down-H Action/Sensor Pair

Step 4: *Down, H*

Previous State

0	0	0
0	0.004	0.006
0.027	0	0.963

Observation Model

0.9	0.9	0.05
0.05	0.05	0.05
0.05	0	0.9

Transition State

0	0	0
0	(0.1) (0.004) + (1) (0.002)	(0.1) (0.006)
(0.1) (0.027) + (1) (0.027)	0	(1) (0.963) + (0.9) (0.006) + (0.1) (0.963)

Current State

0	**0**	**0**
0	**0**	**0**
0.002	**0**	**0.998**

*Normalize(Observational Model * Transition State)*

Figure 9: Down-H Action/Sensor Pair

Final Summary Table

Step 1: Right, N

0.001	0.013	0.025
0.023	0.226	0.451
0.248	0	0.014

Step 2: Right, N

0	0	0.002
0.002	0.042	0.685
0.267	0	0.001

Step 3: Down, H

0	0	0
0	0.004	0.006
0.027	0	0.963

Step 4: Down, H

0	0	0
0	0	0
0.002	0	0.998

Figure 10: Summary Table of Probabilities after Each Step

As seen in *Figure 10*, the four 3x3 maps show the probabilities indicating the probability of where you are located after each step (each action/sensor pair).

(b) For the coding portion of this assignment, a map was generated with randomly assigned terrain types based on the provided probabilities. An example of such a generated map can be seen in *Figure 11* below. The map also shows the randomly generate starting point, as well as a series of one hundred actions that happen from the starting point. Lastly, the ending point is also shown along with a set of one hundred sensor readings that were also generated is also displayed.

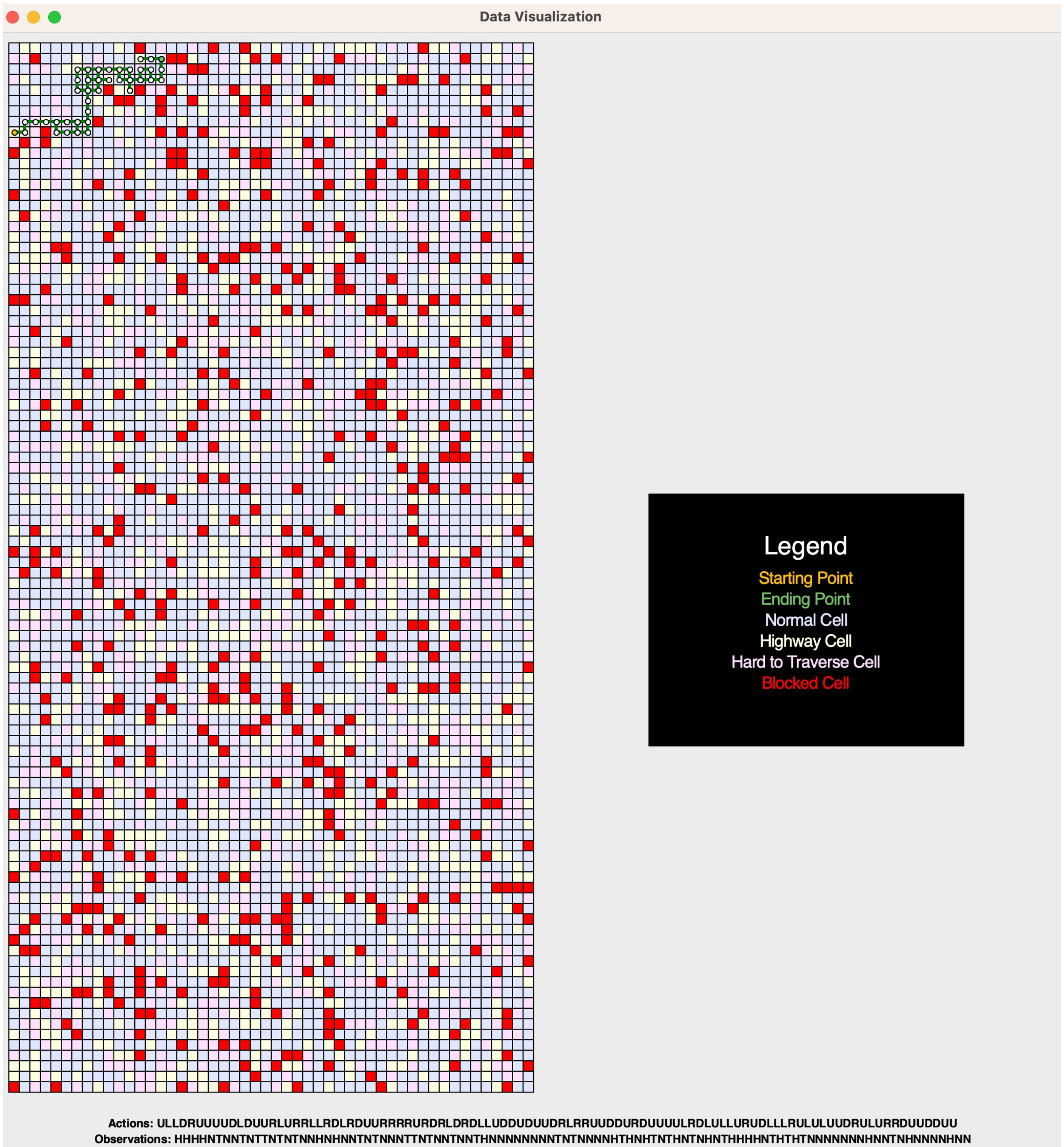

Figure 11: Map Visualization

This visualization is created from a text file that is generated by the program. *Figure 12* shows a sample generated file. Within the text file, Lines 1-2 show the starting point, Lines 3-103 are the one hundred ground truth states, Lines 104-105 show the action (a string with a length of one hundred, made up of each of the four actions), as well as Lines 106-107, show the sensor readings generated (a string with a length of one hundred, made up of each of the three possible sensor readings).

```
Start:
8 0
Path:
8 1
8 1
8 1
8 1
7 1
7 2

1 13
1 12
1 13
1 14
Actions:
ULLDRUUUUDLDUURLURRLLRDLRDUURRRRURDRLDRDLLUDDUDUUDRLRRUUDDURDUUUULRDLULLURUDLLLRULULUUDRULURRDUUDDUU
Sensor Readings:
HHHHNTNNTNTTNTNTNNHNHNNTNTNNNTTNTNNTNNTHNNNNNNNNTNTNNNNHTHNHTNTHNTNHNTHHHHNTHTHTNNNNNNNHNNTNHNNNNHNN
```

Figure 12: Sample Data File

In this way, the program generates and visualizes a map, along with the ground truth states, actions, and sensor readings, for analysis in the next assignment.

(c) We were unable to implement the solution to Part C.

6. Problem 6

(a)

$$NetGain = (4,000 - 3,000) * P(q^+) + (4,000 - 4,400) * P(q^-)$$
$$NetGain = (4,000 - 3,000) * 0.70 + (4,000 - 4,400) * 0.30 = 580$$

The expected net gain is 580 dollars without a test.

(b)

$$P(q^+|Pass) = P(Pass|q^+) * P(q^+)/P(Pass) = \alpha * P(Pass|q^+) * P(q^+)$$
$$\alpha * P(Pass|q^+) * P(q^+) = P(Pass|q^+)P(q^+)/(P(Pass|q^+)P(q^+) + P(Pass|q^-)P(q^-))$$
$$P(q^+|Pass) = 0.70 * 0.80/(0.80 * .70 + 0.35 * 0.30) = 0.8421$$

$$P(q^-|Pass) = P(Pass|q^-) * P(q^-)/P(Pass) = \alpha * P(Pass|q^-) * P(q^-)$$
$$\alpha * P(Pass|q^-) * P(q^-) = P(Pass|q^-)P(q^-)/(P(Pass|q^-)P(q^-) + P(Pass|q^+)P(q^+))$$
$$P(q^+|Pass) = 0.35 * 0.30/(0.35 * 0.30 + 0.80 * 0.70) = 0.1579$$

$$P(q^+|\neg Pass) = P(\neg Pass|q^+) * P(q^+)/P(\neg Pass) = \alpha * P(\neg Pass|q^+) * P(q^+)$$
$$\alpha * P(\neg Pass|q^+) * P(q^+) = P(\neg Pass|q^+)P(q^+)/(P(\neg Pass|q^+)P(q^+) + P(\neg Pass|q^-)P(q^-))$$
$$P(q^+|\neg Pass) = 0.2 * 0.70/(0.2 * 0.70 + 0.65 * 0.30) = 0.4179$$

$$P(q^-|\neg Pass) = P(\neg Pass|q^-) * P(q^-)/P(\neg Pass) = \alpha * P(\neg Pass|q^-) * P(q^-)$$
$$\alpha * P(\neg Pass|q^-) * P(q^-) = P(\neg Pass|q^-)P(q^-)/(P(\neg Pass|q^-)P(q^-) + P(\neg Pass|q^+)P(q^+))$$
$$P(q^-|\neg Pass) = 0.65 * .30/(0.2 * 0.70 + 0.65 * 0.30) = 0.5821$$

(c) For the utility functions of the quality of the car, we used the value of the gain of selling the car. The gain of selling a good shape car with the test is (4,000 - 3,000 - 100), or 900 dollars, and the gain of selling a bad car with the test is (4,000 - 3,000 - 1,400 - 100), or -500 dollars.

$$U(q^+) = 900$$
$$U(q^-) = -400$$
$$EU(Pass) = \sum_q P(q|Pass) * U(q) = P(q^+|Pass)U(q^+) + P(q^-|Pass)U(q^-)$$
$$EU(Pass) = 0.8421 * 900 + 0.1579 * -500 = 678.94$$

$$EU(\neg Pass) = \sum_q P(q|\neg Pass) * U(q) = P(q^+|\neg Pass)U(q^+) + P(q^-|\neg Pass)U(q^-)$$
$$EU(\neg Pass) = 0.4179 * 900 + 0.5821 * -500 = 85.06$$

The best decision given either a pass or a fail is to buy the car, as the expected utility given either a pass or a fail is positive. Therefore, some profit would be expected with either a pass or a fail.

(d) As you would want to sell given either a pass or a fail from the mechanic, the optimal information from the mechanic's test is not that valuable. We would not bring C_1 to the mechanic, as the expected utility is favorable with either result of the test, rendering the test useless while costing 100 dollars.

7. Problem 7

(a) Original Utilities:
The original utilities utilized were essentially an empty array. Although it was not entirely empty, the array of size four would consist of four 0's which would be updated as we iterated through value iteration. This was done by initializing U and U'.

(b) Intermediate Results:

```
1: [0, 0, 0, 0]
        a[1, 2, 3, 1]
20: [3.292985586347631, 3.7707632134854783, 4.393053374685112, 3.8304854168777087]
        a[2, 2, 3, 1]
40: [3.8516467855534433, 4.336648232691622, 4.949908618890849, 4.3972734135838945]
        a[2, 2, 3, 1]
60: [3.9199323799856374, 4.4050406015616295, 5.01816751971359, 4.465679129258628]
        a[2, 2, 3, 1]
80: [3.928239715562185, 4.413349515358644, 5.026474460735022, 4.4739882403332025]
        a[2, 2, 3, 1]
100: [3.9292497734683254, 4.414359596592277, 5.027484512809289, 4.474998324482771]
        a[2, 2, 3, 1]
120: [3.929372574109574, 4.414482397578326, 5.027607313364336, 4.47512112551192]
        a[2, 2, 3, 1]
Iterations: 124


Optimal Utilities: [3.9293795341512, 4.414489357626929, 5.027614273404219, 4.475128085561394]


Optimal Policy: [2, 2, 3, 1]


Computation Time:  1.1920928955078125e-06 seconds
```

Figure 13: Intermediate Results

(c) Implementation:

i. Algorithm To Find Optimal Utilities:
Input(s): set of states (S), transition function ($P(s'|s,a)$), discount factor (γ), and a maximum error threshold allowed in a utility for any given state (ϵ)

Output(s): Optimal Utilities (U)

initialize U' to be an array of zero's with length = $|S|$
while True **do**
  $U = U'$
  $\sigma = 0$
  for $s \in S$ **do**
    $U'[s] = r(s) + \gamma * max_a \sum_{s'} P(s'|s,a) * U[s']$
    $\sigma = max(\sigma|U'[s] - U[s])$
  end for
  if $\sigma <= (\epsilon * (1 - \gamma))/\gamma$ **then**

break
end if
end while
return U

ii. Algorithm To Find Optimal Policies

Input(s): set of states (S), transition function ($P(s'|s,a)$), and optimal utilities for particular state(s) (U)

Output(s): Optimal Policies (π^*)

initialize π^* to be an array of actions with length $|S|$
for $s \in S$ **do**
π^*= $argmax_a \sum_{s'} P(s'|s,a) * U[s']$
end for
return π^*