CS440 Assignment 3

Arnold Trakhtenberg, Yequ Sun Rutgers ID: 171000520, 155004469

April 12, 2017

Question 1

a.

$$\begin{aligned} P(A,B,C,D,E) &= P(A)P(B)P(C)P(D \mid A,B)P(E \mid B,C) \\ &= 0.2 \times 0.5 \times 0.8 \times 0.1 \times 0.3 \\ &= 0.0024 \end{aligned} \tag{1}$$

b.

$$\begin{aligned} P(\neg A,\neg B,\neg C,\neg D,\neg E) &= P(\neg A)P(\neg B)P(\neg C)P(\neg D \mid \neg A,\neg B)P(\neg E \mid \neg B,\neg C) \\ &= 0.8 \times 0.5 \times 0.2 \times 0.1 \times 0.8 \\ &= 0.0064 \end{aligned} \tag{2}$$

c.

$$\begin{aligned} P(\neg A \mid B,C,D,E) &= \alpha P(\neg A)P(B)P(C)P(D \mid \neg A,B)P(E \mid B,C) \\ &= \alpha \times 0.8 \times 0.5 \times 0.8 \times 0.6 \times 0.3 \\ &= \alpha \times 0.0576 \\ &= \frac{0.0576}{P(B,C,D,E)} \\ &= \frac{0.0576}{P(B)P(C)P(E \mid B,C)\sum_A P(a)P(D|a,B)} \\ &= \frac{0.0576}{0.5 \times 0.8 \times 0.3 \times (0.2 \times 0.1 + 0.8 \times 0.6)} \\ &= \frac{0.0576}{0.06} \\ &= 0.96 \end{aligned} \tag{3}$$

Question 2

a. Query - $P(B \mid +j, +m)$
Factors - $P(B),\ P(E),\ P(A \mid B, E),\ P(+j \mid A),\ P(+m \mid A)$

$f_1(A, B) = \sum_e P(e)P(A \mid B, e)$
$f_2(B, +j, +m) = \sum_a P(+j \mid a)P(+m \mid a)f_1(a)$
$f_3(B, +j, +m) = P(B)f_2(B, +j, +m)$

$$\begin{aligned} f_3(B, +j, +m) &= P(B)f_2(B, +j, +m) \\ &= P(B)\sum_a P(+j \mid a)P(+m \mid a)f_1(a) \\ &= P(B)\sum_a P(+j \mid a)P(+m \mid a)\sum_e P(e)P(a \mid B, e) \\ &= 0.001 \times [0.9 \times 0.7(0.002 \times 0.95 + 0.998 \times 0.94) + 0.05 \times 0.01(0.002 \times 0.05 + 0.998 \times 0.06)] \\ &= 0.0006 \end{aligned} \tag{4}$$

b. After variable elimination has been performed, only 7 multiplication and 3 addition operations are needed to evalute our query. Using the tree enumeration algorithm, we would need to perform 11 multiplication and 3 addition operations. Thus, evaluating the query after variable elimination is more efficient. Note: normalization is not taken into account as the normalization factor is the same in both cases.

c. Enumeration:

$$\begin{aligned} \Theta(P(X_1 \mid X_n = true)) &= \Theta(2^{n-2} + \Theta(P(X_1 \mid X_{n-1})) \\ &= \Theta(2^{n-2} + 2^{n-3} + \Theta(P(X_1 \mid X_{n-2})) \\ &= \Theta(\sum_{i=0}^{n-2} 2^i) \\ &= \Theta(2^{n-1} - 1) \\ &= \Theta(2^n) \end{aligned} \tag{5}$$

Variable elimination does not provide a benefit because each branch in the enumeration is unique. Thus, it has the same time complexity as enumeration in this case.

Question 3

a.

$$
\begin{aligned}
P(X \mid MB(X)) &= \alpha P(X, MB(X)) \\
&= P(X, U_1 ... U_m, Y_1 ... Y_n, Z_{1,1} ...) \\
&= P(X \mid U_1 ... U_m) P(Y_1 ... Y_n, Z_{1,1} ...) \\
&= P(X \mid U_1 ... U_m) P(Y_1 \mid Z_{1,1} ... Z_{1,j}) ... P(Y_n \mid Z_{n,1} ... Z_{n,k}) \\
&= P(X \mid U_1 ... U_m) \prod_{Y_i} P(Y_i \mid Z_{i,1} ...)
\end{aligned} \tag{6}
$$

b. $P(R \mid S, W) = P(W \mid S, R) \sum_c P(c) P(R \mid c) P(S \mid c) = 0.99(0.5 \times 0.8 \times 0.1 + 0.5 \times 0.2 \times .5) = 0.0891$
MCMC:

(a) Set the initial state in MCMC to have $Sprinkler = true$, $WetGrass = true$, and assign the remaining unknown values randomly.

(b) Sample a $Rain$ or $Cloudy$, given their Markov Blanket. Store the resulting state (in some efficient manner) and continue sampling until the number of states where $Rain = true$ settles.

There are $2^2 = 4$ possible states, given that we have 2 evidence variables and 2 unknown variables.

c.

$$
\begin{array}{c c}
 & \begin{array}{cccc} tt & tf & ft & ff \end{array} \\
\begin{array}{c} tt \\ tf \\ ft \\ ff \end{array} & \begin{bmatrix} .63 & .22 & .4 & 0 \\ .28 & .39 & 0 & .11 \\ .09 & 0 & .12 & .02 \\ 0 & .39 & .48 & .87 \end{bmatrix}
\end{array}
$$

Note: tf would represent $Rain = true, Cloudy = false$.

Question 4

a. $E(Buy(C_1) = .7(4000 - 3000) + .3(4000 - 3000 - 1400) = 580$

b. $P(q^+ \mid pass) = \dfrac{P(pass \mid q^+)P(q^+)}{P(pass)} = \dfrac{0.8(0.7)}{0.8(0.7) + 0.35(0.3)} = 0.84$
$P(q^- \mid pass) = 1 - P(q^+ \mid pass) = 1 - 0.84 = 0.16$
$P(q^+ \mid \neg pass) = \dfrac{(1 - P(pass \mid q^+)P(q^+)}{P(\neg pass)} = \dfrac{0.2(0.7)}{1 - (0.8(0.7) + 0.35(0.3))} = 0.42$
$P(q^- \mid \neg pass) = 1 - P(q^+ \mid \neg pass) = 1 - 0.42 = 0.58$

c. $E(buy \mid pass) = 0.84(1000) + 0.16(-400) = 780$
$E(buy \mid \neg pass) = 0.42(1000) + 0.58(-400) = 188$
Since the expected utility is postiive whether or not the test passes, we should always buy the car.

d. The mechanic's test is not valuable, because the outcome does not affect our decision. So, I would not take my car to the mechanic.

Question 5

a. Initial

0.125	0.125	0.125
0.125	0.125	0.125
0.125	0	0.125

1st step

0.001315789	0.013157895	0.025
0.023684211	0.236842105	0.45
0.236842105	0	0.013157895

2nd step

7.69231E-06	0.000146154	0.002153846
0.002492308	0.047353846	0.697846154
0.249230769	0	0.000769231

3rd step

1.18441E-06	2.25037E-05	1.84241E-05
2.19115E-05	0.004061918	0.00613522
0.02151117	0	0.968227669

4th step

1.21451E-07	2.30757E-06	1.04958E-07
1.85551E-07	0.000232552	3.58956E-05
0.001226567	0	0.998502265

b. 1st step

0.002 0.016 0.018

0.032 0.289 0.321

0.321 0.000 0.002

2nd step

0.000 0.000 0.000

0.005 0.043 0.476

0.476 0.000 0.000

3rd step

0.000 0.000 0.000

0.000 0.005 0.006

0.058 0.000 0.932

4th step

0.000 0.000 0.000

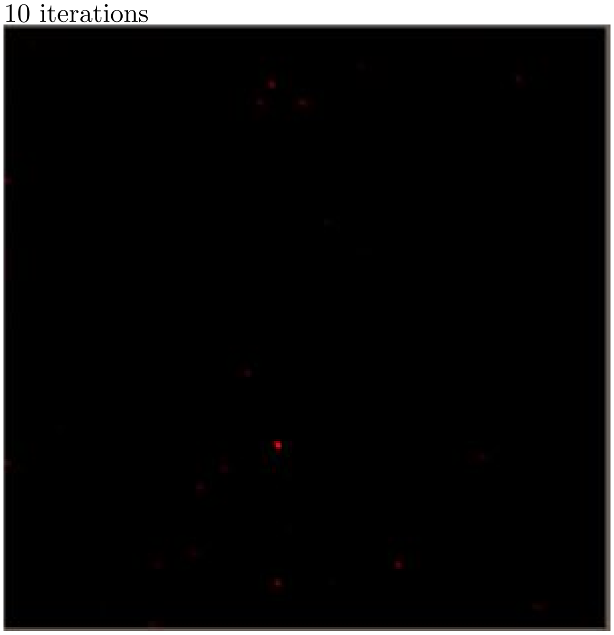
0.000 0.000 0.000

0.000 0.000 1.000

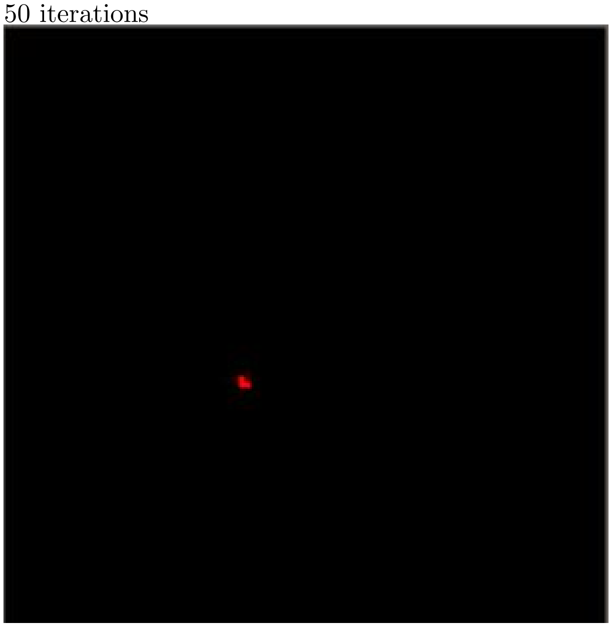
Most Likely Sequence: $(1,2)->(2,3)->(2,3)->(3,3)->(3,3)$

c. The output file is included in the submitted compressed file.
First 3 lines: Width, Height of the map. The number of actions and observations
The big chunk: Actual map terrain data
The next two lines: Action sequence and Observation Sequence
The next two lines: The initial coordinates
The rest: Actual path coordinates

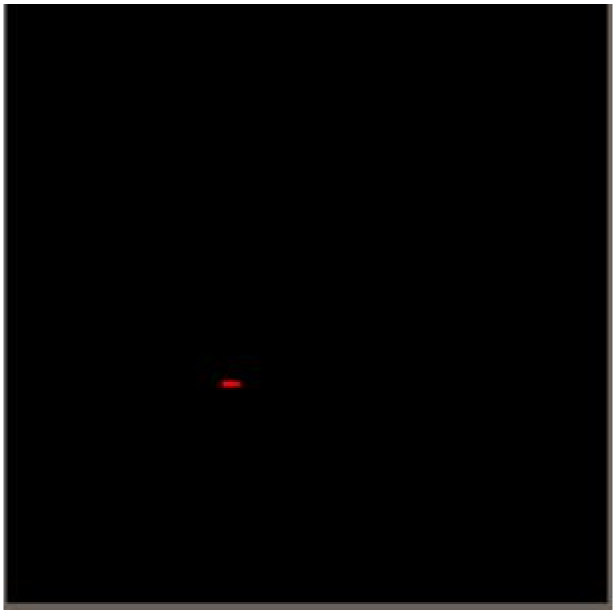
d. 10 iterations



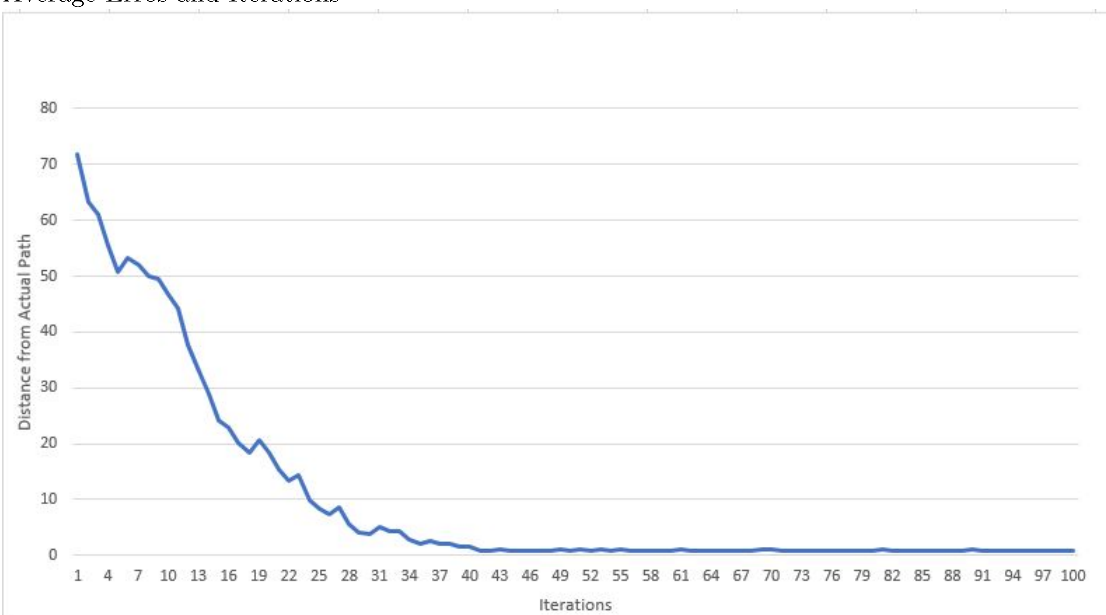
50 iterations



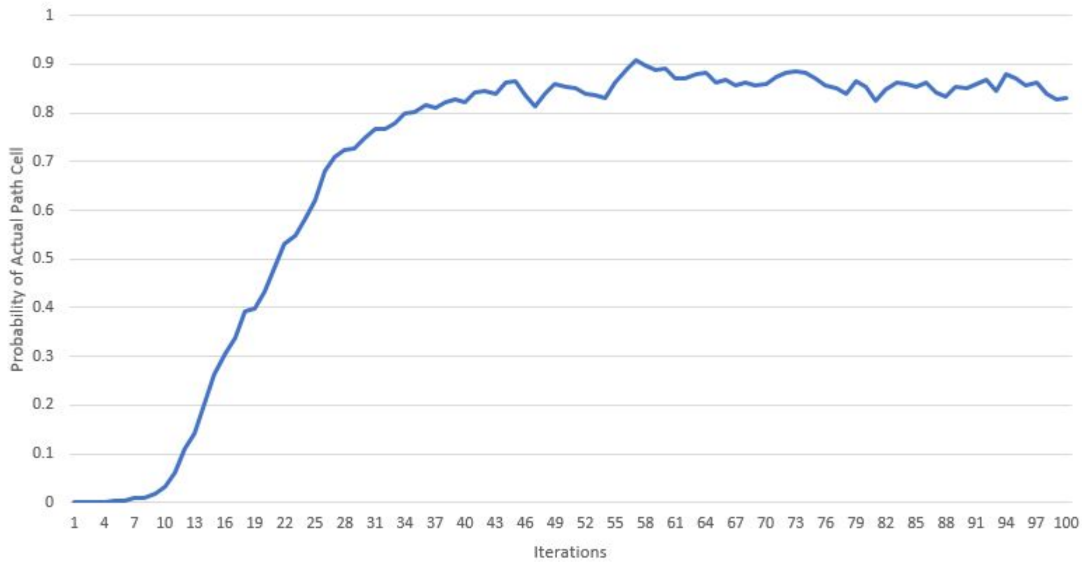
100 iterations



e. Average Erros and Iterations



Averrage Probability on Actual Path and Iterations



Obviously, as the number of iterations increases, the estimation is more and more accurate.

f.

g.

h. In this case, the best way to improve efficiency is to prune the states. According to our transitional model, the agent can only move to a certain direction by 0 or 1 unit of distance. So, we only need to consider neighboring cells. This method introduces no computational errors.