AI ASSIGNMENT — Uncertainty, Bayesian Nets, HMM and Kalman

Saumil Lakra, 2021097

THEORY

_/_/__

Q1 : (a) Direct sampling

In this sampling, we sample data pts from the given probability distribution. Its strength is because of its simplicity, it is less computation heavy. Its weakness is that it requires full dataset.

Rejection sampling

In this sampling, the generated samples are ~~acep~~ accepted or rejected on the basis of defined criteria. Its strength is that it ~~is~~ can handle complex distribution. Its weakness is that it can be computationally heavy.

Gibbs sampling

In this sampling, we iteratively sample one variable at a time, keeping the rest of the variables constant. It's strength is that due to its sampling strategy, ~~its~~ it is effective for multidimensional distributions. Its weakness is that it performance depends on the inter dependency of the variables.

In the given dataset, direct sampling seems more viable. The data is simple and using this sampling technique can reduce computation.

Q4 Random variables :

A: people travel by air
T: people travel by train
B: people travel for business
L: people travel for leisure
required for (b) & (c)

(b) $P(A|T) = 0.4 = \frac{4 \times 10}{10 \times 10}$

$= \frac{4\cancel{0}}{1\cancel{0}\cancel{0}} \times 3\cancel{0} = 12$ people.

(c) $P(A) = 0.8$

$P(B|A) = 0.2$

$P(A \cap B) = ?$

$P(B/A) = \frac{P(B \cap A)}{P(A)}$

$= \quad P(B \cap A) = P(A \cap B) = P(B|A) \cdot P(A)$

$= 0.2 \times 0.8$

$= 0.160$

(d) Larger sample size decreases error and increases accuracy and precision. Large sample size helps the model to maker better prediction, thus increasing prediction and accuracy

There are some probabilities that are very small in ~~ba~~ value. To increase their significance, ~~[illegible]~~ increasing same size can help.

Example of ~~ba~~ small probabilities ⇒

P(stressed ∩ Air travel) = 0.065

P(travel by bus ∩ Low stress level) = 0.015

Q2

(a) People who read books $\Rightarrow B$
People who read academic journals $\Rightarrow$ ~~J~~ J
People who participate in book clubs $\Rightarrow C$

Above are the random variables.

Statement 1 $\Rightarrow$ $P(B \cup J) = 0.910$

Statement 2 $\Rightarrow$ $P(J|B) = 0.400$
~~$P(\neg J|B) = 0.600$~~ $P(B) = 0.6$

Statement 3 $\Rightarrow$ $P(C|B) = 0.320$

Statement 4 $\Rightarrow$ $P(J \cap \neg B) = 0.227$

Statement 5 $\Rightarrow$ $P(\neg B \cap \neg J) = 0.090$

Statement 6 $\Rightarrow$ $P(J|\neg B) = 0.716$

Statement 7 $\Rightarrow$ $P(C \cap J) = 0.088$

Statement 8 $\Rightarrow$ ~~$P(C \cap J$~~ $P(C \cup J) = 0.631$

Statement 9 $\Rightarrow$ $P(J|C) = 0.400$

Statement 10 $\Rightarrow$ $P(J) = 0.500$

Statement 11 $\Rightarrow$ $P(C|\neg B) = 0.0044$

(b) List of axioms that they ~~the~~ satisfy:
- All ~~prop~~probabilities are greater than 0
- The sum of probabilities is 1
- ~~Disjoint events can be added.~~

In the previous part, we can see that all the ~~prop~~ probabilities are greater than 0.

For the (c) part of Q2, we can see that the probabilities add up to 1.

~~that the~~
in ~~the formula~~

Product Rule: $P(a \wedge b) = P(a|b).P(b)$
$= P(b|a)\,P(a)$

(c)

	C	¬C	Total
$B \cap J$	0.08736	0.186	0.273
$B \cap \neg J$	0.1312	0.279	0.4102
$\neg B \cap J$	0.001	0.226	0.091
$\neg B \cap \neg J$	0	0.090	0.090
TOTAL	0.219	0.781	1

$P(B \cap J \cap C) = P(C|B \wedge J).P(B \wedge J)$ [Product rule from slides]
$= P(C|B).P(B \wedge J)$
$= \cancel{B}\ 0.320 \times P(B \wedge J)$

$P(B \cup J) = P(B) + P(J) - P(B \wedge J)$
$0.91 = 0.682 + 0.5 - P(B \wedge J)$ —①
$P(B \cap J) = 0.683 + 0.5 - 0.91$
$= 0.273$

$P(B \wedge J \wedge C) = 0.320 \times 0.273$
$= \cancel{0.08704}\ 0.08736$

for eq① P(B) is calculated using,
$$P(J|\neg B) = \frac{P(J \cap \neg B)}{P(\neg B)}$$
$$0.716 = \frac{0.227}{P(\neg B)}$$
$$P(B) = 1 - \frac{0.227}{0.716} = 0.6[illegible]$$

$$P(B \cap \neg J \cap C) = P(C \mid B \cap \neg J) \, . \, P(B \cap \neg J)$$
$$= 0.320 \times P(B \cap \neg J)$$

$$P(B \cap \neg J) = P(B) - P(B \cap J)$$
$$= 0.683 - 0.273$$
$$= 0.410$$

$$P(B \cap \neg J \cap C) = 0.320 \times 0.410$$
$$= 0.1312$$

$$P(\neg B \cap J \cap C) = P(C \mid \neg B \cap J) \, . \, P(\neg B \cap J)$$
$$= P(C \mid \neg B) \, . \, P(\neg B \cap J)$$
$$= 0.0044 \times P(J \mid \neg B) \, . \, P(\neg B)$$
$$= 0.0044 \times 0.716 \times (1 - 0.683)$$
$$= 0.0009 \approx 0.001$$

$$P(\neg B \cap \neg J \cap C) = P(C \mid \neg B \cap \neg J) \, . \, P(\neg B \cap \neg J)$$
$$= 0 \times P(\neg B \cap \neg J)$$
$$= 0$$

$$P(B \cap J \cap \neg C) = P(\neg C \mid B \cap J) \, . \, P(B \cap J)$$
$$= (1 - P(C \mid B \cap J) \, . \, P(B \cap J)$$
$$= (1 - 0.320) \times 0.273$$
$$= 0.18564 \approx 0.186$$

$P(B \wedge \neg J \wedge \neg C) = P(\neg C \mid B \wedge \neg J) \times P(B \wedge \neg J)$

$= (1 - P(C \mid B)) \times ((P(B) - P(B \wedge J)$

$= (1 - 0.320) \times (0.683 - 0.273)$

$= 0.2788 \simeq 0.279$

$P(\neg B \wedge J \wedge \neg C) = P(\neg C \mid \neg B \wedge J)\ P(\neg B \wedge J)$

$= \cancel{(1 - P(C \mid \neg B) \times P(J \mid B))}$

$= (1 - P(C \mid \neg B))\ (P(J \mid \neg B) . P(\neg B))$

$= (1 - 0.0044) \times 0.716 \times 0.317$

$\cancel{= 0.2169 \simeq 0.217}$

$= 0.2259 \simeq 0.226$

$P(\neg B \wedge \neg J \wedge \neg C) = 1 - P(B \cup J \cup C)$

$= 0.090$

~~P(...)~~

_/_/_

(d) B and J are conditionally independent of C - ①

J & C are not conditionally independent given B - ②

B & C are not conditionally independent given J - ③

for statement ① proof ⇒

$P(B \cap J | C)$

$P(B \cap J \cap C) = 0.08736 \approx 0.088$

$P(C) = \cancel{0.2198}\ 0.220$

$$P(B \cap J | C) = \frac{P(B \cap J \cap C)}{P(C)} = \frac{0.088}{0.220} = 0.4$$

$$P(B|C) = \frac{P(B \cap C)}{P(C)} = \frac{0.088 + 0.1304}{0.2198}$$

$= \cancel{0.8068}\ 0.9936$

$$P(J|C) = \frac{P(J \cap C)}{P(C)} = \frac{0.088 + 0.0014}{0.2198} = 0.4067$$

$P(B|C).P(J|C) = 0.4030 \approx P(B \cap J|C)$

⇒ B & J are conditionally independent of C.

for statement ② proof ⇒

$$P(J \cap C | B) = P(J|B) \cdot P(C|B)$$

$$P(J \cap C | B) = \frac{P(J \cap C \cap B)}{P(B)} = \frac{0.088}{0.6825} = 0.1297$$

$P(J|B)$ & $P(C|B)$

$$P(J|B) = \frac{0.4034}{0.6825} = 0.5918 \simeq 0.591$$

$$P(C|B) = \frac{0.2184}{0.6825} = 0.3201 \simeq 0.320$$

$$P(J|B) \cdot P(C|B) = 0.1890$$

⇒ J & C are conditionally independent of B

for statement ③ proof ⇒

$$P(B \cap C | J) = P(B|J) \cdot P(C|J)$$

$$P(B \cap C | J) = \frac{P(B \cap C \cap J)}{P(J)} = \frac{0.088}{0.5} = 0.176$$

$$P(B|J) = \frac{0.4034}{0.5} = 0.8068 \simeq 0.807$$

$$P(C|J) = \frac{0.0894}{0.5} = 0.1788 \simeq 0.179$$

⇒ $P(B \cap C | J) \neq P(B|J) \cdot P(C|J)$

⇒ B & C are not conditionaly independent

Q3 (a) Random variables ⇒

Adversarial Perturbations : A (cause)

Backdoor Attack : B (cause)

Misclassification alarm : M (effect)

Bayesian formulation ⇒

~~$P(\text{cause}|\text{effect}) = \frac{P(\text{effect}|\text{cause})\,P(\text{cause})}{P(\text{effect})}$~~ [from slides]

~~$= P(A,B$~~

~~$\Rightarrow P(A|B,M) = P(M|A,B)\,P(A$~~

$$P(Y|X_1, \ldots X_n) = \frac{P(X_1, \ldots, X_n|Y)\,P(Y)}{P(X_1, \ldots, X_n)}$$

[from slides]

$$\Rightarrow \quad P(M|A,B) = \frac{P(A,B|M)\,P(M)}{P(A,B)}$$

(b) Prior : $P(M)$ ⇒ probability that there is a misclassification.

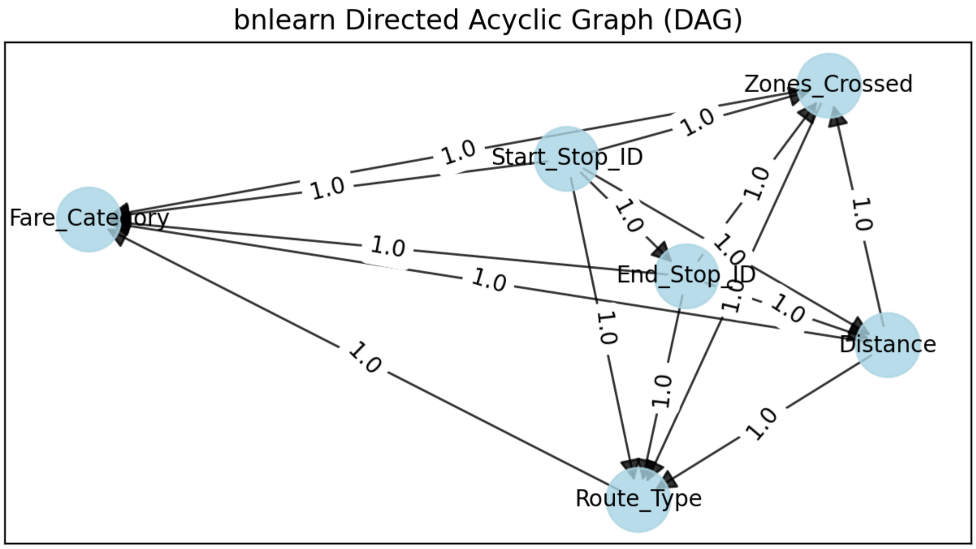
Likelihood : $P(A,B|M)$ ⇒ probability that there is adversarial perturbation (A) & backdoor Attack (B) given there is a misclassification (M)

Posterior Probability : $P(M|A,B)$ ⇒ probability that there is a misclassification (M), given that adversarial perturbations (A), backdoor attack (B) is present.

(c) Since now, we ~~are~~ are able ~~to~~ to detect backdoor attack (B), we can now track the exact effect adversarial perturbation has on the experiment.

P(B) is now know
P(A) can be found given we ~~at~~ have the information about misclassifications.

Q4.
Task 1: Initial Bayesian Network:
(c) Visualization:



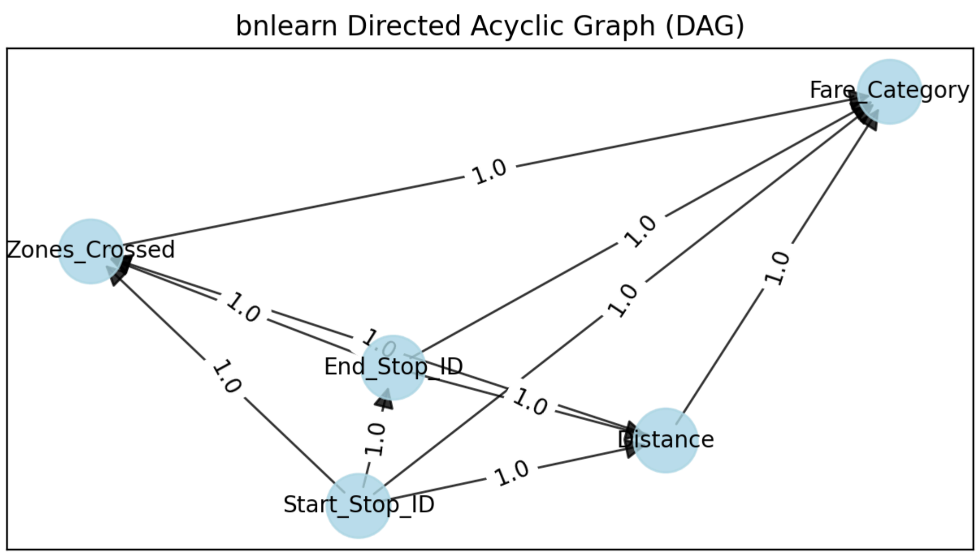
Time taken by make_network is 750.6627984046936 seconds

Total Test Cases: 350
Total Correct Predictions: 350 out of 350
Model accuracy on filtered test cases: 100.00%

Task 2: Pruned initial Bayesian network:

(b) The pruning method that I have used is the chi-square test with alpha value 0.05 (these are the default parameters in the independence test function). Independence test function applies chi-test on the edges of the network and prunes those edges whose p value is less than alpha. In the initial Bayesian network we can see that there is an extra node in the network named 'Route_Type' and all its corresponding edges which gets pruned after chi-square test is applied. Now, the network takes 54.2% less time to fit to the new edges while retaining the accuracy. This makes the model faster and also there is no loss in the accuracy on the validation data. The model now focuses on more significant relationships in the dataset. The model has become more efficient now.

(c) Visualization:



Time taken by make_pruned_network is 406.9261703491211 seconds

Total Test Cases: 350
Total Correct Predictions: 350 out of 350
Model accuracy on filtered test cases: 100.00%

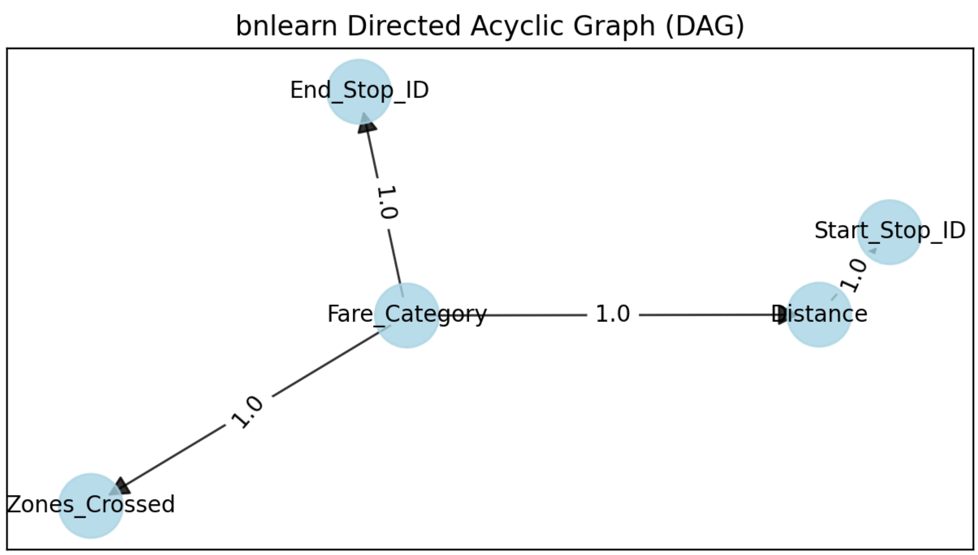
Task 3: Optimize the Bayesian network:
(b) The optimized network takes a lot less time to fit than the previous methods. There is no loss in the accuracy. The optimization technique I have used is hill climbing (hc parameter in the function) which is simple and a fast algorithm.

Time taken to fit data in all the methods:

Time taken by make_network is **750.6627984046936 seconds**
Time taken by make_pruned_network is **406.9261703491211 seconds**
Time taken by make_optimized_network is **9.832030057907104 seconds**

We can see that the optimized network is more efficient than the base_model and the pruned_model.

(c) Visualization:



Time taken by make_optimized_network is 9.832030057907104 seconds

Total Test Cases: 350
Total Correct Predictions: 350 out of 350
Model accuracy on filtered test cases: 100.00%

Q5.

(a) Transition states and emission probabilities are given in the code file.

(c)

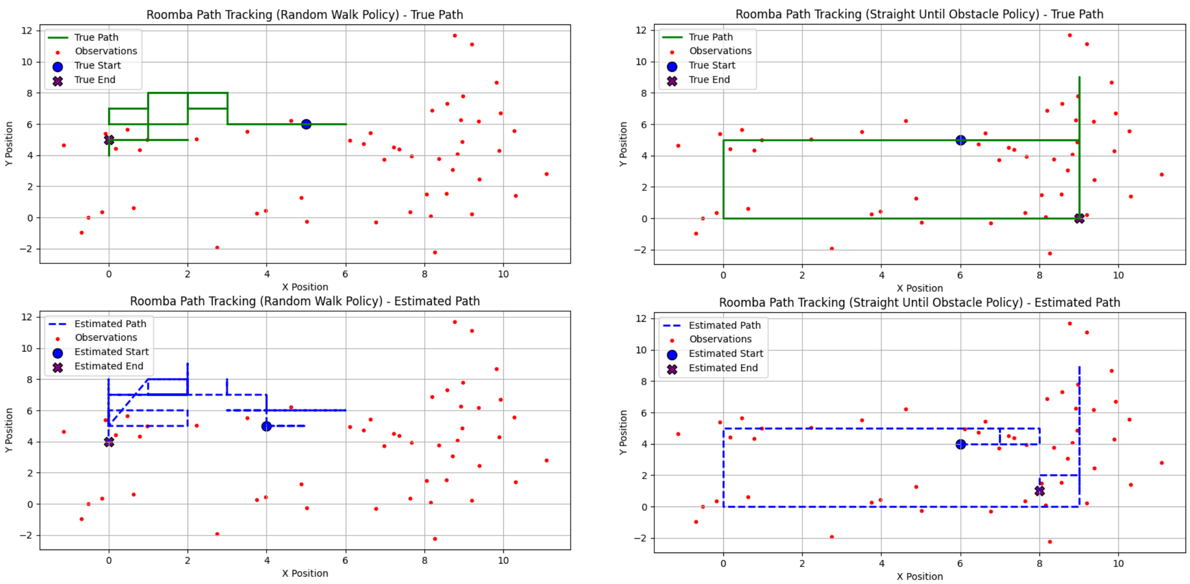
Seed = 111

Processing policy: random_walk

Tracking accuracy for random walk policy: 42.00%

Processing policy: straight_until_obstacle

Tracking accuracy for straight until obstacle policy: 78.00%



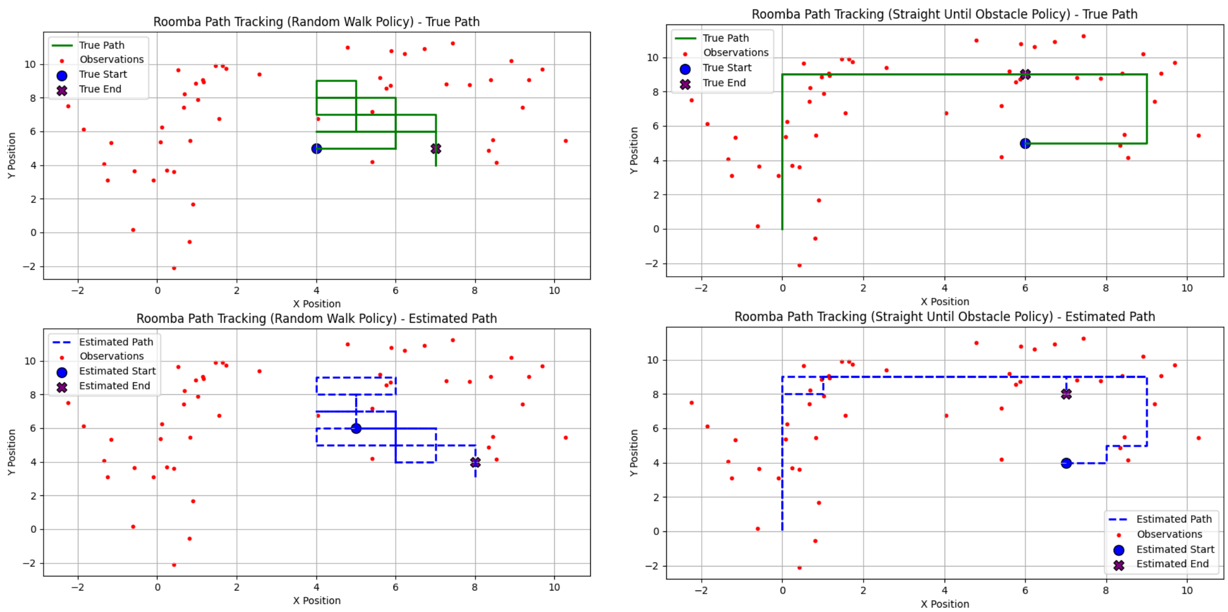
Seed = 150

Processing policy: random_walk

Tracking accuracy for random walk policy: 52.00%

Processing policy: straight_until_obstacle

Tracking accuracy for straight until obstacle policy: 76.00%



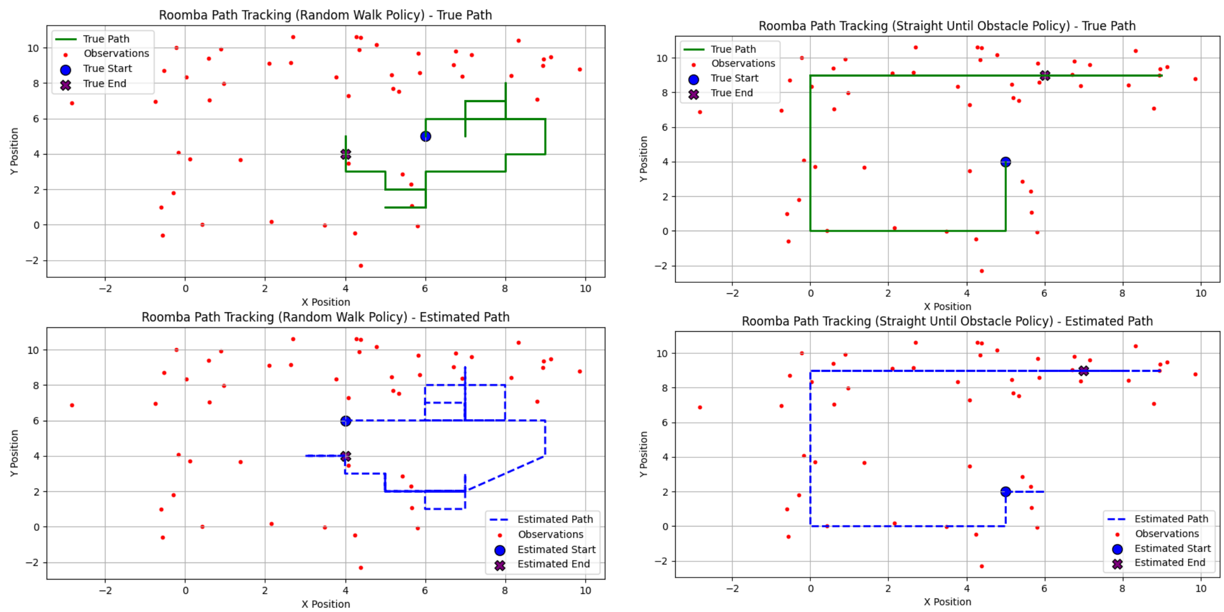
Seed = 125

Processing policy: random_walk

Tracking accuracy for random walk policy: 24.00%

Processing policy: straight_until_obstacle

Tracking accuracy for straight until obstacle policy: 84.00%



References:

- https://www.probabilitycourse.com/chapter1/1_3_2_probability.php
- https://pieriantraining.com/viterbi-algorithm-implementation-in-python-a-practical-guide/
- https://youtu.be/fX5bYmnHqqE?si=XumOOlyLHKAaSYMp
- https://www.youtube.com/watch?v=IqXdjdOgXPM