CSE643: Artificial Intelligence
Assignment 3: Uncertainty, Bayesian Nets, HMM and Kalman Filtering

Himanshu Raj (2022216)

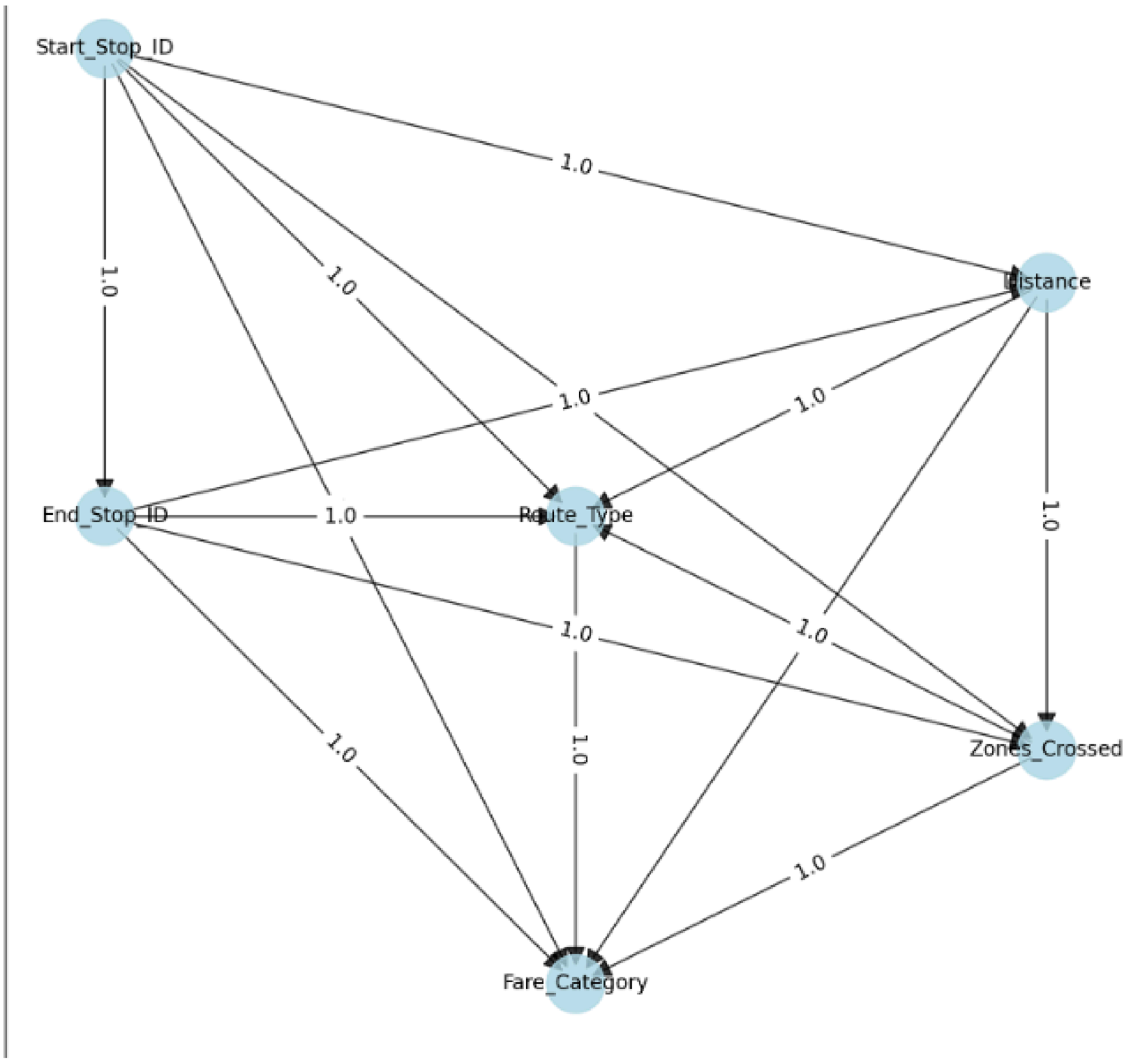
November 24, 2024

Computational

Bayesian network for fare classification

1. Base Model

The Bayesian network for the base model is shown below:



Time taken and memory usage to initialize and train the base model:

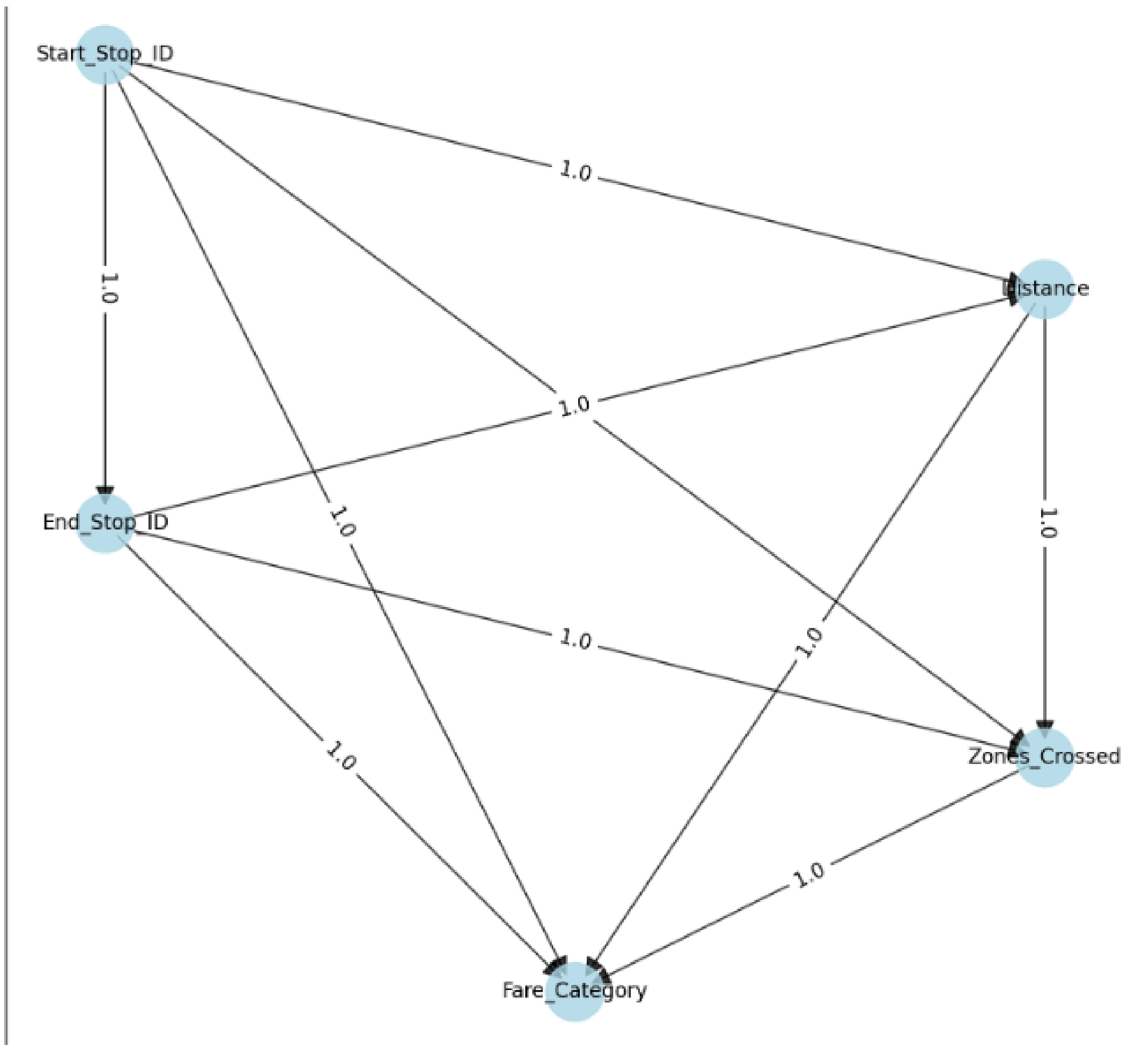
```
Metrics for base model
Time taken: 3224.912490129471 seconds
Peak memory usage: 13986057.478515625 KB
```

Base model's accuracy on validation set:

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

2. Pruned Model

The Bayesian network for the pruned model is shown below:



Performed independence test on DAG from the base model and pruned nodes and edges using the ‘independence_test’ function in the bnlearn package, which computes edge strength with chi-square test.

The new Bayesian network had the node ‘Route_Type’ pruned and all incoming and outgoing edges of this node because it had a constant value of 3 in the entire dataset and didn’t seem to affect the prediction of ‘Fare_Category’. This makes the Bayesian network simpler, from 15 directed edges and 6 nodes to 10 directed edges and 5 nodes which also simplifies Conditional Probability Tables.

Due to this network simplification, it improved the model’s efficiency (time taken to fit the data) by almost half, from 53 minutes to 28 minutes. The accuracy on the validation set remains the same as before, i.e. 100%. Memory usage differs by 1634 MB.

Time taken and memory usage to initialize and train the pruned model:

```
Metrics for pruned model
Time taken: 1712.2595636844635 seconds
Peak memory usage: 12311929.4453125 KB
```

Pruned model’s accuracy on validation set:

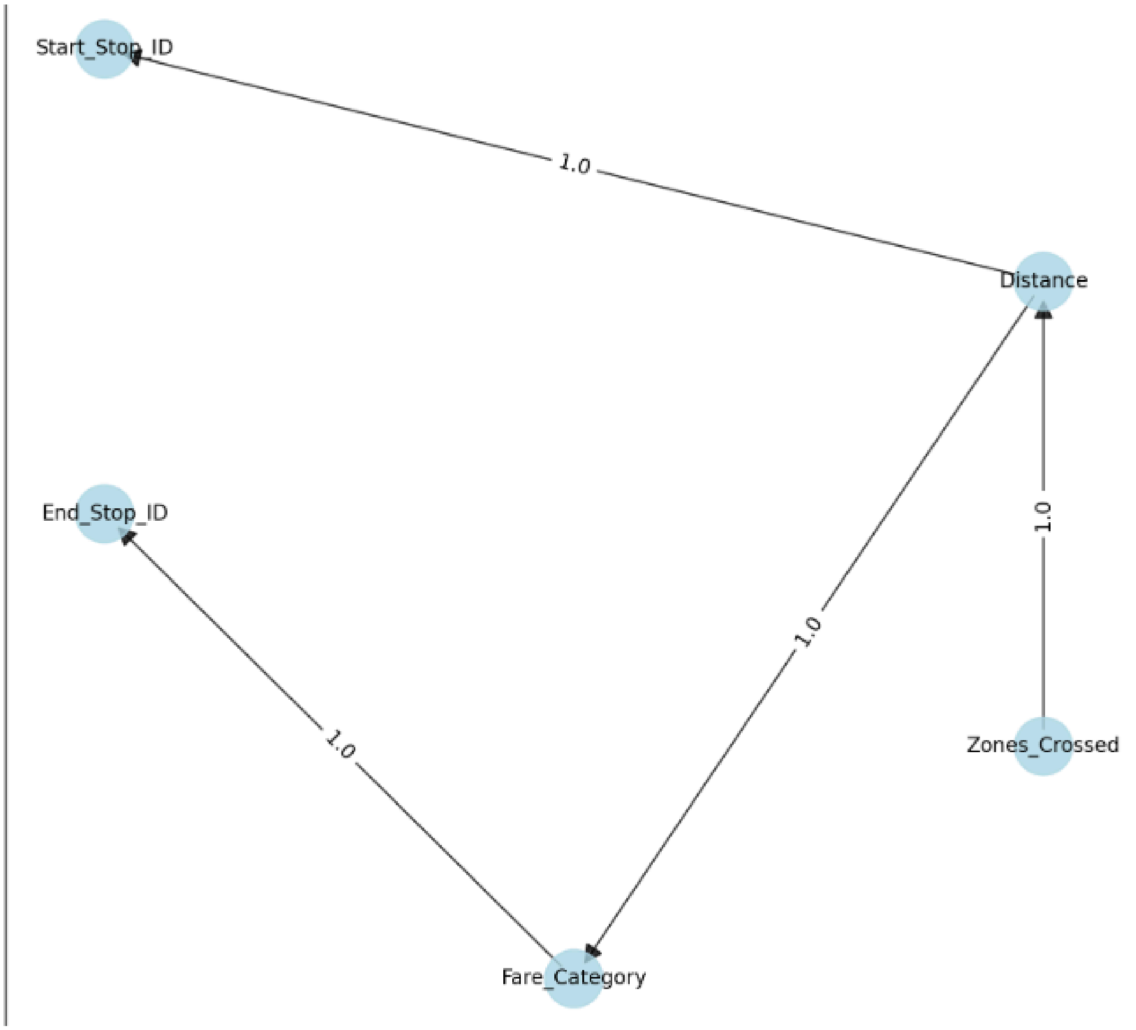
```
Total Test Cases: 350
Total Correct Predictions: 350 out of 350
Model accuracy on filtered test cases: 100.00%
```

3. Optimized Model

Applied Hill Climb Search method in the ‘structure_learning’ function in the bnlearn package to optimize and refine the Bayesian network structure. The obtained network is way simpler than the base model, from 15 directed edges and 6 nodes to 4 directed edges and 5 nodes which also simplifies Conditional Probability Tables.

Due to this network simplification, it improved the model’s efficiency (time taken to fit the data) by a factor of 400, from 53 minutes to 8.5 seconds. The accuracy on the validation set remains the same as before, i.e. 100%. Memory usage differs by 13.33 GB.

The Bayesian network for the optimized model is shown below:



Time taken and memory usage to initialize and train the optimized model:

```
Metrics for optimized model
Time taken: 8.411476373672485 seconds
Peak memory usage: 7781.361328125 KB
```

Optimized model’s accuracy on validation set:

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

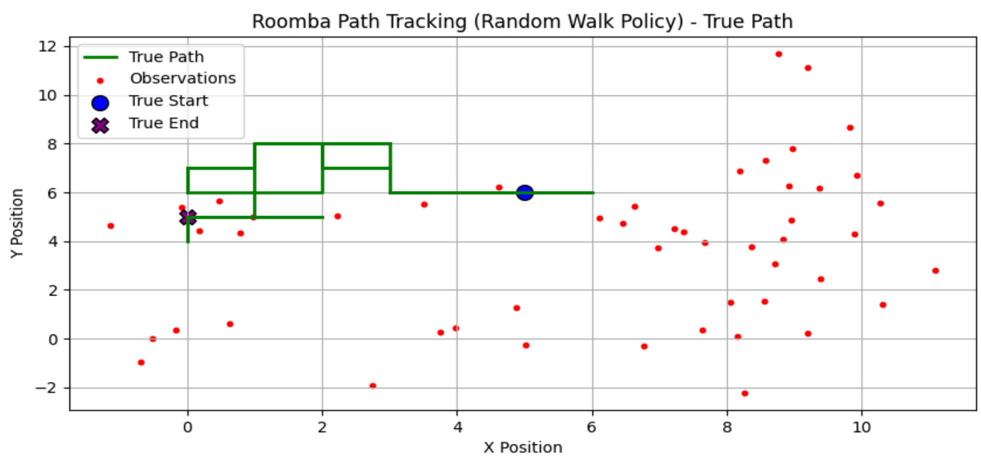
Tracking a Roomba Using the Viterbi Algorithm (HMM)

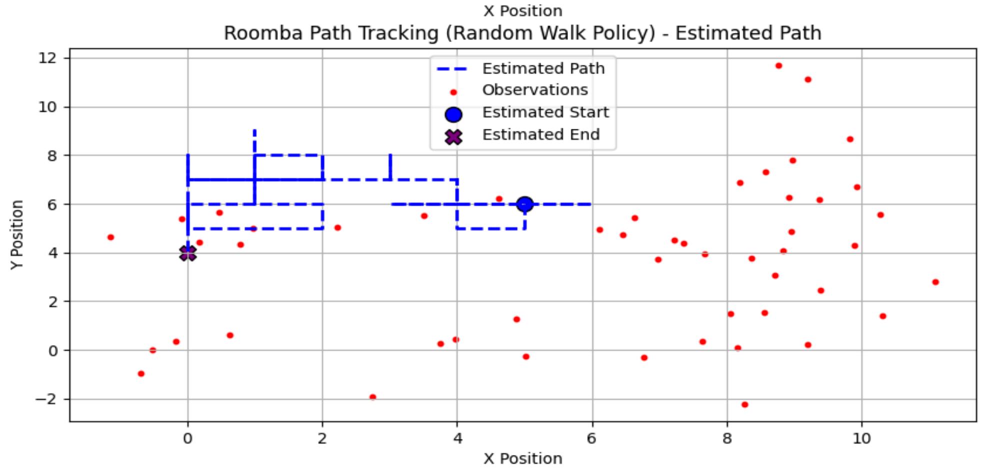
Seed values = [111, 69, 42]

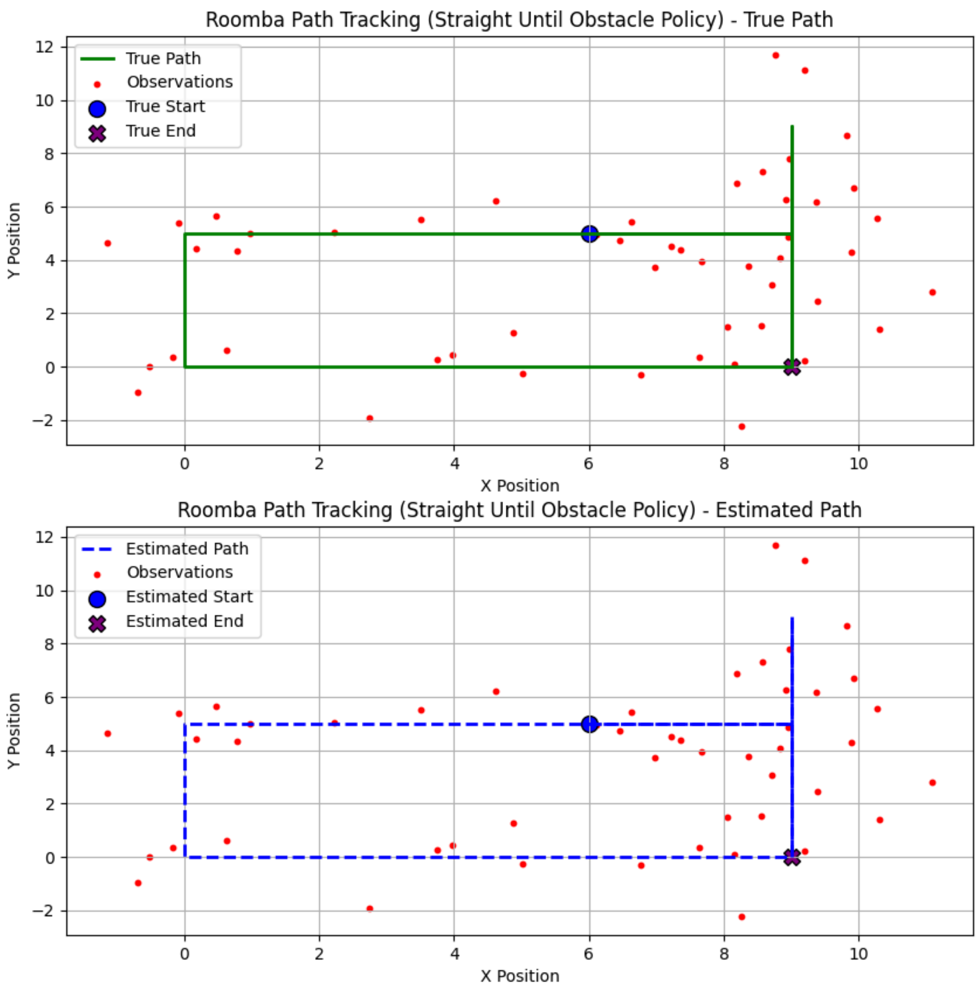
Seed value = 111

```
Processing policy: random_walk
Tracking accuracy for random walk policy: 42.00%

Processing policy: straight_until_obstacle
Tracking accuracy for straight until obstacle policy: 100.00%
```

straight_until_obstacle is more accurate because it is majorly deterministic. It walks in a straight line until it encounters any obstacle, which is a deterministic step, and non-determinism only kicks in when it encounters any obstacle. random_walk is entirely non-deterministic and hence has low accuracy.



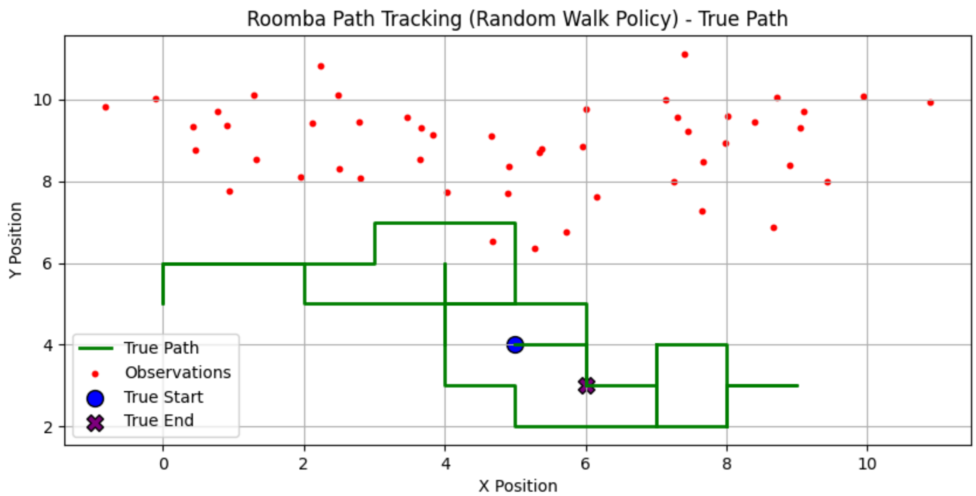




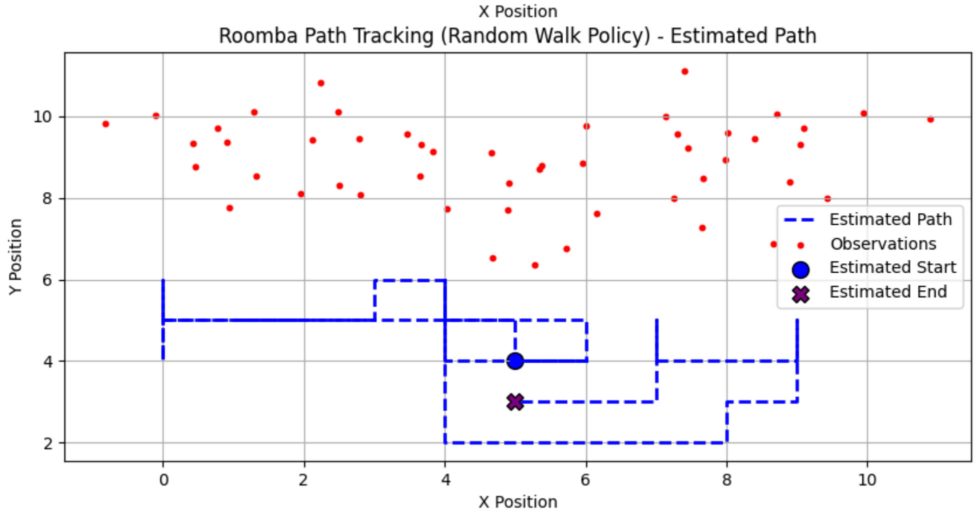
Seed value = 69

```
Processing policy: random_walk
Tracking accuracy for random walk policy: 44.00%

Processing policy: straight_until_obstacle
Tracking accuracy for straight until obstacle policy: 82.00%
```

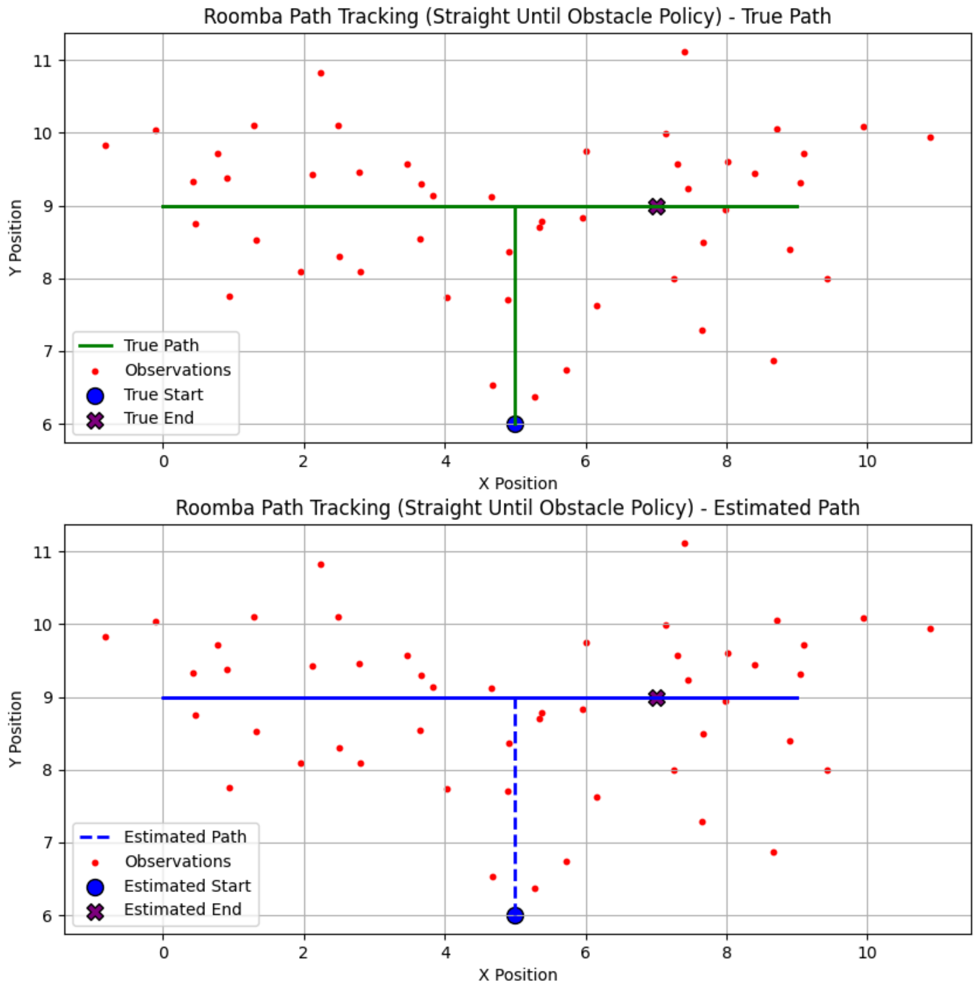
straight_until_obstacle is more accurate because it is majorly deterministic. It walks in a straight line until it encounters any obstacle, which is a deterministic step, and non-determinism only kicks in when it encounters any obstacle. random_walk is entirely non-deterministic and hence has low accuracy.



Roomba Path Tracking (Random Walk Policy) - True Path
True Path
Observations
True Start
True End
X Position
Y Position



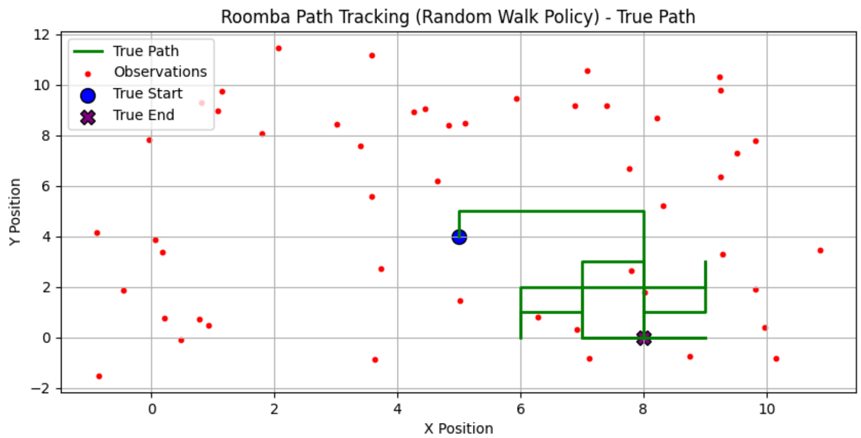
Roomba Path Tracking (Random Walk Policy) - Estimated Path
Estimated Path
Observations
Estimated Start
Estimated End
X Position
Y Position

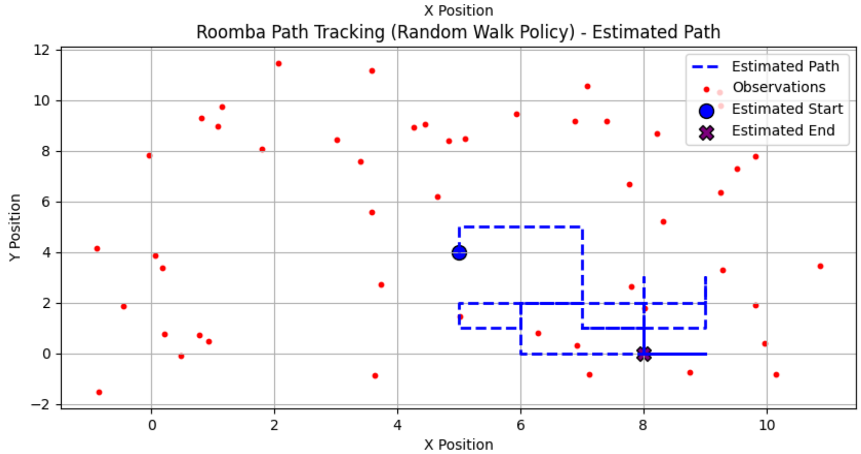


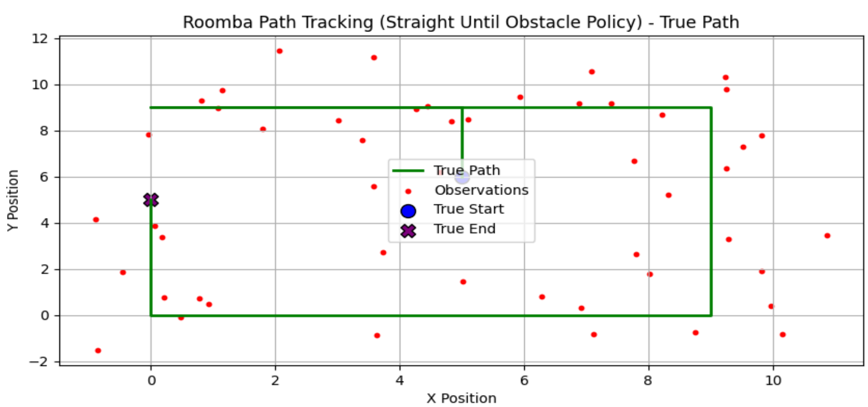
Seed value = 42

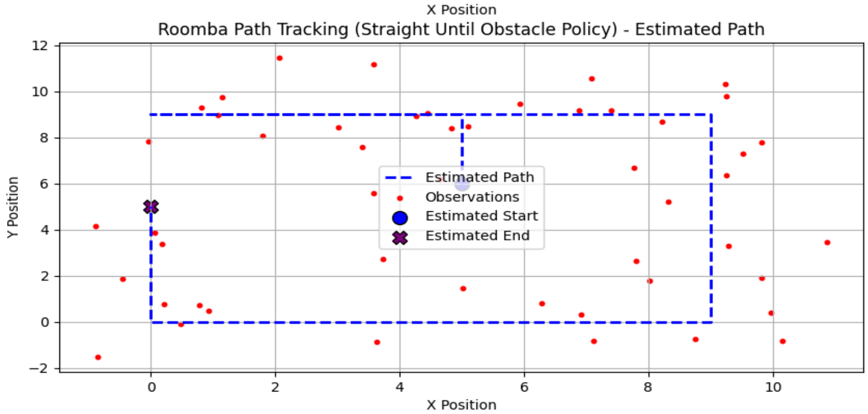
```
Processing policy: random_walk
Tracking accuracy for random walk policy: 64.00%

Processing policy: straight_until_obstacle
Tracking accuracy for straight until obstacle policy: 100.00%
```

straight_until_obstacle is more accurate because it is majorly deterministic. It walks in a straight line until it encounters any obstacle, which is a deterministic step, and non-determinism only kicks in when it encounters any obstacle. random_walk is entirely non-deterministic and hence has low accuracy.


Roomba Path Tracking (Random Walk Policy) - True Path
True Path
Observations
True Start
True End
Y Position
X Position
12
10
8
6
4
2
0
−2
0
2
4
6
8
10


Roomba Path Tracking (Random Walk Policy) - Estimated Path
Estimated Path
Observations
Estimated Start
Estimated End
Y Position
X Position
12
10
8
6
4
2
0
−2
0
2
4
6
8
10


Roomba Path Tracking (Straight Until Obstacle Policy) - True Path
True Path
Observations
True Start
True End
Y Position
X Position
12
10
8
6
4
2
0
−2
0
2
4
6
8
10


Roomba Path Tracking (Straight Until Obstacle Policy) - Estimated Path
Estimated Path
Observations
Estimated Start
Estimated End
Y Position
X Position
12
10
8
6
4
2
0
−2
0
2
4
6
8
10

Himanshu Raj
2022216

Date: / /
Page No.

AI Assignment -3
Theory

Q1

a. Direct sampling - directly samples from the given distribution.

- strength - easier if the probability distribution is known.
- weakness - difficult for complex or constrained distributions.
- This method works well for the given dataset as probabilities are already provided.

Rejection sampling

- It generates samples and rejects them if they don't fit the criteria for target distribution.
- In our dataset, for low probability events it will reject many samples.
- strength: can handle complex or constrained distributions
- weakness: inefficient if acceptance rate is low because it will waste/reject many samples.

Gibbs Sampling

- samples are iteratively generated from the conditional distribution of each variable.
- It is not required for our dataset because it doesn't have complex joint distributions.
- strength: can handle complex and high-dimensional probability distributions.
- weakness: slow for bigger or ~~larger~~ complex datasets.

Date : / /
Page No.

b $P(\text{leisure} \mid \text{train}) = 0.4$

sample size $= 100$

people who prefer train $= 30$

~~P(T)~~ $P(\text{train}) = 0.3$

$P(\text{leisure} \wedge \text{train}) = P(\text{leisure} \mid \text{train}) \times P(\text{train})$

$= 0.4 \times 0.3$

$= \underline{0.120}$

c $P(\text{Air}) = 0.8$

$P(\text{Business} \mid \text{Air}) = 0.2$

$P(\text{Business} \wedge \text{Air}) = P(\text{Air}) \times P(\text{Business} \mid \text{Air})$

$= 0.8 \times 0.2$

$= \underline{0.160}$

d Increasing sampling size would improve accuracy because larger samples would mitigate the effect of outliers. and estimates would converge to true values (if they exist) ~~Increas~~ Increasing sampling size would also improve precision as larger sample size gives regular samples leading to more consistent results.

In our dataset, on increasing sampling size, the estimates of small probability events will improve. and it will minimize the irregularities among samples

Date : / /
Page No.

Q2.

a) Random Variables - B : reads book
J : accesses academic journals
C : participates in book clubs

$P(J \vee B) = 0.910$

$P(J \mid B) = 0.400$ $\quad P(\bar{J} \mid B) = 0.600$

$P(C \mid B) = 0.320$

$P(J \wedge \bar{B}) = 0.227$

$P(\bar{B} \wedge \bar{J}) = 0.090$

$P(J \mid \bar{B}) = 0.716$

$P(C \wedge J) = 0.088$

$P(C \vee J) = 0.631$

$P(J \mid C) = 0.400$

~~P(A) = 0.500~~ $P(J) = 0.500$

$P(C \mid \bar{B}, J) = 0.0044 = P(C \mid \bar{B}, J)$

b) All the probabilities are greater than equal to 0 in dataset.
Non-negativity ← (Axiom 1)

Axiom 2 - $P(S) = 1$ $\quad$ S is sample space
in our case, one can be J, and another on next page.
$P(J \vee B) + P(\bar{J} \wedge \bar{B}) = 0.91 + 0.09 = \underline{1}$

Axiom3 - Disjoint events are mutually exclusive.
we don't have any disjoint events.

They satisfy axiom-1 and 2 clearly and not sure about 3 as there are no disjoint events.

Summing all joint probabilities
$= 0.9993$
≈ 1

$P(S) = 1 \rightarrow$ axiom 2

c) Joint probability distribution Table

B	J	C	Probability
T	T	T	0.087
T	T	F	0.186
T	F	T	0.131
T	F	F	0.2789
F	T	T	0.0009
F	T	F	0.226
F	F	T	0.0004
F	F	F	0.089

d) Conditional independence b/w RVs.

RVs X & Y are independent iff $P(X \wedge Y) = P(X)P(Y)$

- B and J

$P(B \wedge J) = 0.273$

$P(B)\,P(J) = 0.683 \times 0.499 = 0.34$

$P(B \wedge J) \neq P(B)\,P(J)$, so they are dependent.

- J and C

$P(J \wedge C) = 0.0879$

$P(J)\,P(C) = 0.499 \times 0.219 = 0.109$

$P(J \wedge C) \neq P(J)\,P(C)$, so they are dependent.

- B and C

$P(B \wedge C) = 0.218$

$P(B)\,P(C) = 0.683 \times 0.219 = 0.150$

$P(B \wedge C) \neq P(B)\,P(C)$, so they are dependent.

Q3 - a) A - adversarial perturbations
B - backdoor attacks
M - misclassification alarm

A & B are considered independent. Both A and B can cause M.

Bayesian network (DAG) -

$$\begin{tikzcd} A \arrow[r] & M \\ B \arrow[ur] & \end{tikzcd}$$

b) Prior Probabilities - initial prob of an event happening
$P(A)$, $P(B)$ and $P(M)$

Likelihood - prob. of misclassification given A or B happens
~~$P(A|M)$ and $P(B|M)$~~ $P(M|A)$ and $P(M|B)$

Posterior - prob. of A or B given misclassification occurs.
~~$P(M|A)$ and $P(M|B)$~~ $P(A|M)$ and $P(B|M)$

c) $P(M) = P(M|A)\,P(A) + P(M|B)\,P(B)$

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

given $P(B)$ increases, $P(M)$ will also increase.
and as $P(M)$ increases, $P(A|M)$ decreases given other probabilities remain same.

So, if backdoor triggers are increased, misclassification alarm prob also increases.
And the ~~reason~~ cause for misclassification being adversial perturbation is reduced or less likely, i.e. $P(A|M)$ decreases.