Assignment 1
COL864: AI for Robot Intelligence

Navreet Kaur
Indian Institute of Technology,Delhi
2015TT10917
tt1150917@iitd.ac.in

I. Probabilistic Graphical Model

A. State Space

The lake is considered to be a 2D grid of size 5x5, which forms the state space, consisting of a total of 25 states.

The following figure shows the state space.

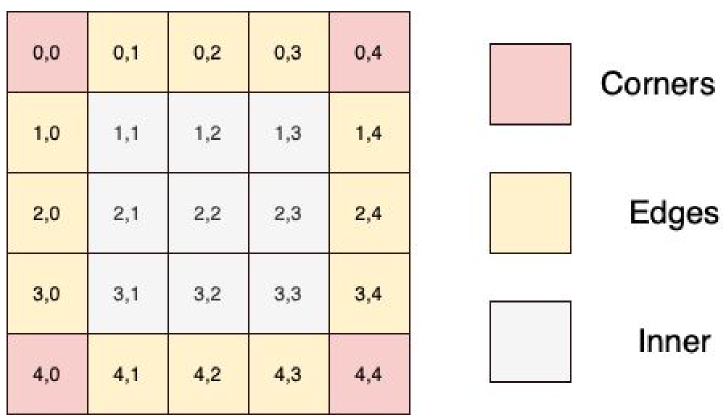

Figure 1

B. Action Space

From any state, agent can take four, equally probable actions: UP, DOWN, LEFT, RIGHT

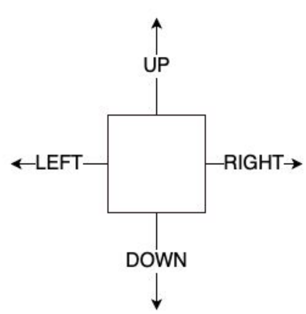

Figure 2

C. Observation Space

At each time step, we get two independent sound observations: ROTOR (R) and BUMP (B). Thus, the observation space consists of 4 different observations - (R,B), (R,~B), (~R,B), (~R,~B).

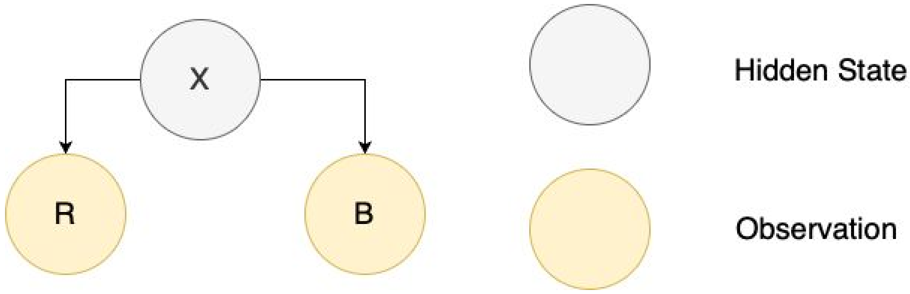

Figure 3

D. Observation Model

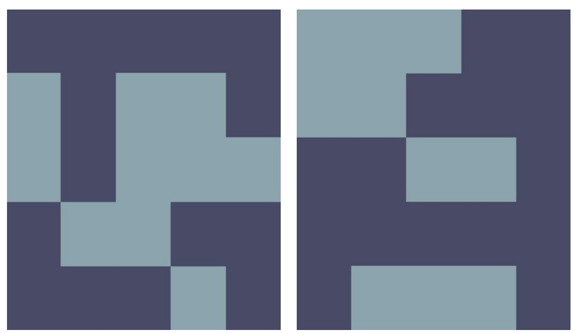

Figure 5(a) *Figure 5(b)*

Figure 5(a) and (b) shows the observation model with probabilities $P(R_t | X_t)$ and $P(B_t | X_t)$ respectively. The dark squares denote 0.1 and light squares denote 0.9 probability.

E. Transition Model

The transition model denotes the probability of transitioning to a state given the current state i.e. $P(X_{t+1} | X_t)$ .There are three types of states: *corner, edge* and *inner* (Figure 1).

Since all actions are equally likely, given any *inner* state, the four adjacent states are equally likely, i.e. 0.25. For the *edge* states, the probabilities of transitioning to the three adjacent states are 0.25 each and with the same probability, the agent stays in the same state. For the *corner* states, the probabilities of transitioning to two adjacent states is 0.25 each and with 0.5 probability it stays in the same state.

Figure 4 shows the transition model. States have been color coded to show their emission probabilities.

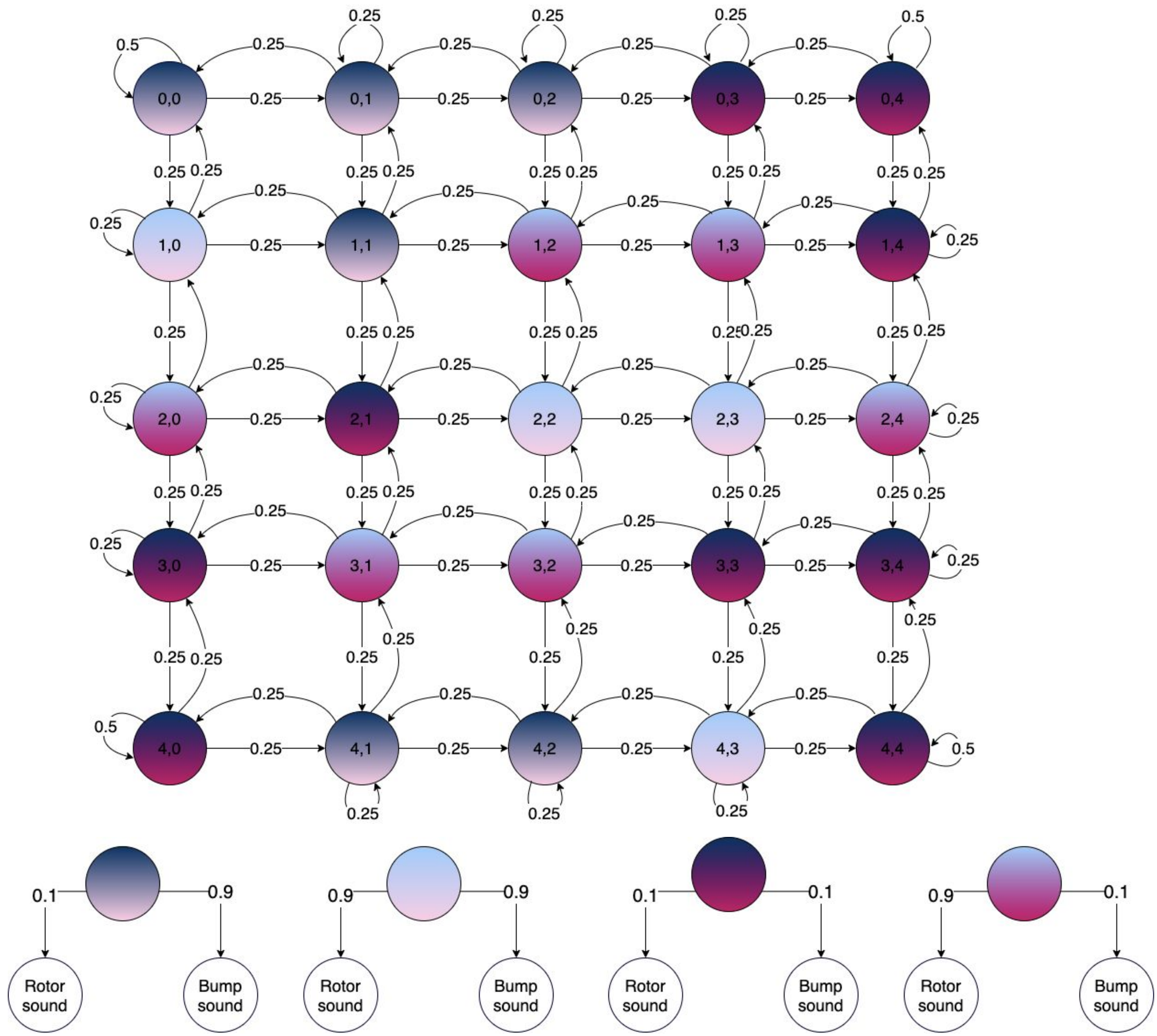

Figure 4: Transition Model

F. Conditional Independence Assumptions

Figure 4 represents the graphical model for this problem.

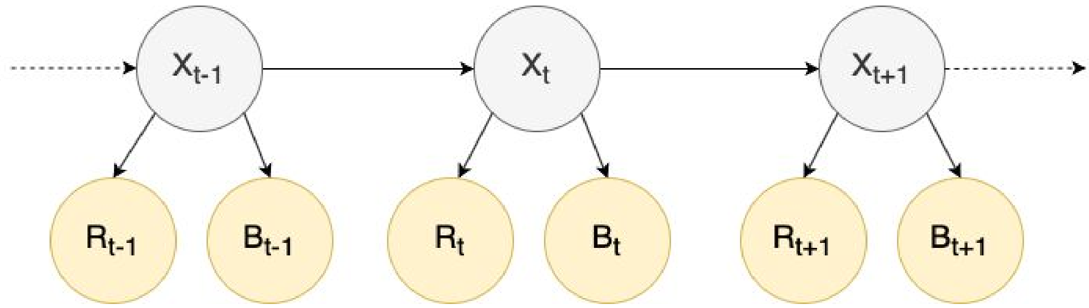

Figure 6

The conditional independence assumptions can be derived from Figure 6as follows:

- $X_{t+1} \perp X_i \,|\, X_t \;;\; \forall\; i < t$
- $R_t \perp X_i \,|\, X_t \;;\; \forall\; i \neq t$
- $B_t \perp X_i \,|\, X_t \;;\; \forall\; i \neq t$
- $R_t \perp R_i \,|\, X_t \;;\; \forall\; i \neq t$
- $R_t \perp B_i \,|\, X_t \;;\; \forall\; i$
- $B_t \perp B_i \,|\, X_t \;;\; \forall\; i \neq t$
- $B_t \perp R_i \,|\, X_t \;;\; \forall\; i$

X_t is the hidden state at time t;
R_t is the rotor observation at time t;
B_t is the bump observation at time t;

The **joint distribution** of the model is given by:

$$P(X_{0:t},\, R_{1:t},\, B_{1:t}) \;=\; P(X_0) \prod_{i=1:t} P(X_i |\, X_{i-1}).\, P(R_i |\, X_i).\, P(B_i |\, X_i)$$

The **Inference tasks** for the following problem are:

- Filtering Task: $P(X_t \,|\, R_{1:t},\, B_{1:t})$
- Smoothing Task: $P(X_k \,|\, R_{1:t},\, B_{1:t})\;;\; 0 \leq k < t$
- Prediction Task: $P(X_{t+k} |\, R_{1:t},\, B_{1:t})\;;\; k > 0$
- Most likely Path: $argmax_{x_{1:t}}\, P(X_{1:t} |\, R_{1:t},\, B_{1:t})$

Simulation of observations for 10 time steps:

Time		State	Action	Observation
[[0]]	:	(0 , 1)	Action.DOWN	~R, B
[[1]]	:	(1 , 1)	Action.DOWN	~R, B
[[2]]	:	(2 , 1)	Action.DOWN	~R, ~B
[[3]]	:	(3 , 1)	Action.RIGHT	R, ~B
[[4]]	:	(3 , 2)	Action.RIGHT	~R, ~B
[[5]]	:	(3 , 3)	Action.DOWN	~R, ~B
[[6]]	:	(4 , 3)	Action.DOWN	R, B
[[7]]	:	(4 , 3)	Action.DOWN	R, B
[[8]]	:	(4 , 3)	Action.RIGHT	R, B

II. Filtering Task

The forward algorithm is used to compute the probability distribution over states at time *t* given the observations from time *0* to *t*, i.e. $P(X_t \mid R_{1:t}, B_{1:t})$

$$P(X_t = j \mid R_{1:t}, B_{1:t}) \propto P(X_t = j , R_{1:t}, B_{1:t})$$

$$\alpha_j(t) = P(X_t = j , R_{1:t}, B_{1:t})$$

$$\alpha_j(t) = \begin{cases} \pi_j b_{jk} & \text{when } t = 1 \\ b_{jk} \sum_{i=1}^{M} \alpha_i(t-1) a_{ij} & \text{when } t \text{ greater than } 1 \end{cases}$$

Figure 7 shows the log-likelihood distribution over states at each time step *t* given evidence till that time step, calculated using the forward algorithm.

Figure 8 shows a comparison between the predicted state and the ground state. We can infer that the maximum log likelihood is a good criteria to predict the state at each time step. The estimation improves with increasing number of time-steps as more and more observations are collected.

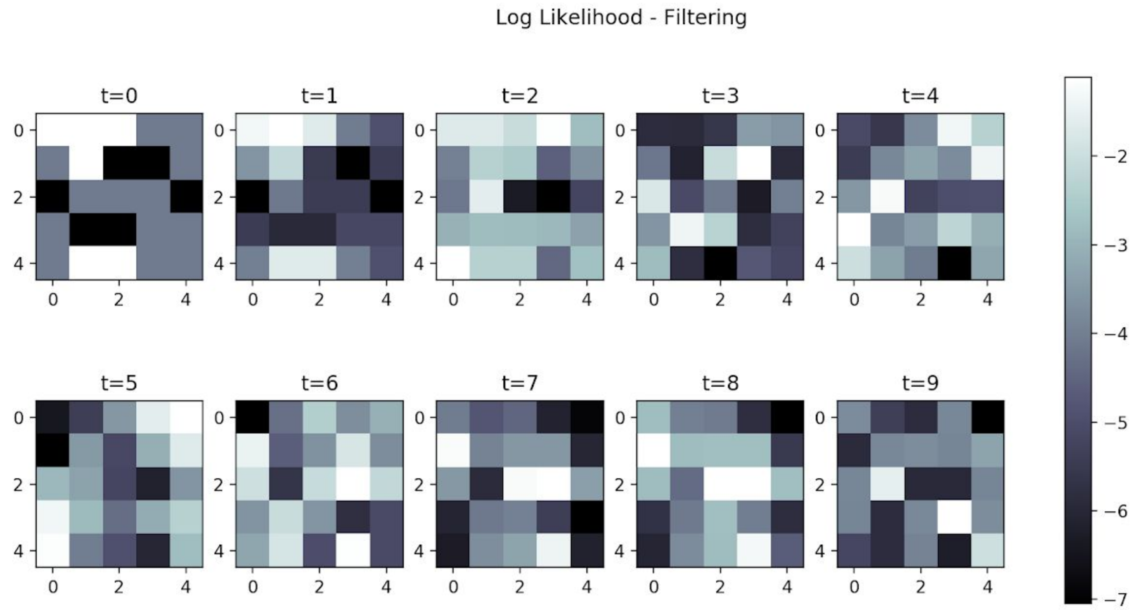

Figure 7: Log-likelihood over states (Filtering)

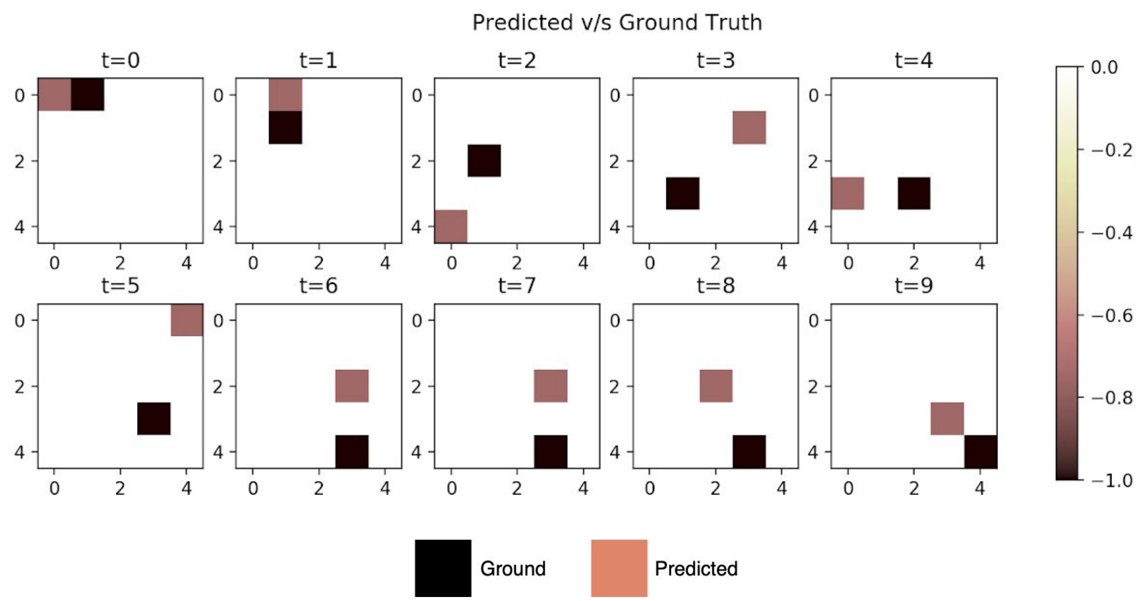

Figure 8: Predicted state v/s Ground state (Filtering)

Now we consider the case of single modality i.e. only bump observations are made.

For Single Modality (Bump)

(1) Filtering

It is noticed that predictions are better when we take more observations as compared to this case, when we consider only bump observations. This makes sense, since more evidence should improve our knowledge of the environment.

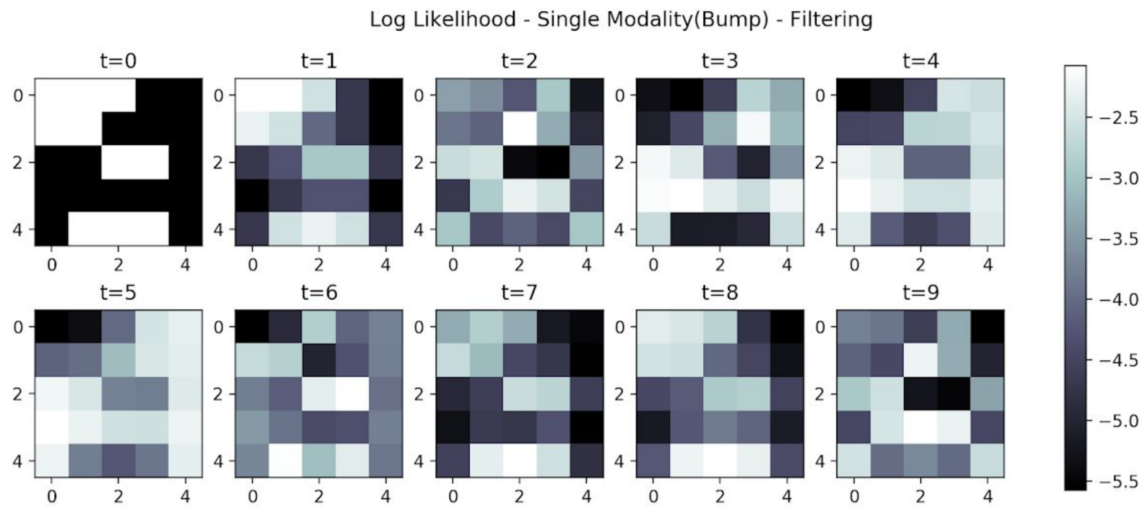

Figure 9: Log-likelihood over states (Filtering) - Only Bump Observations

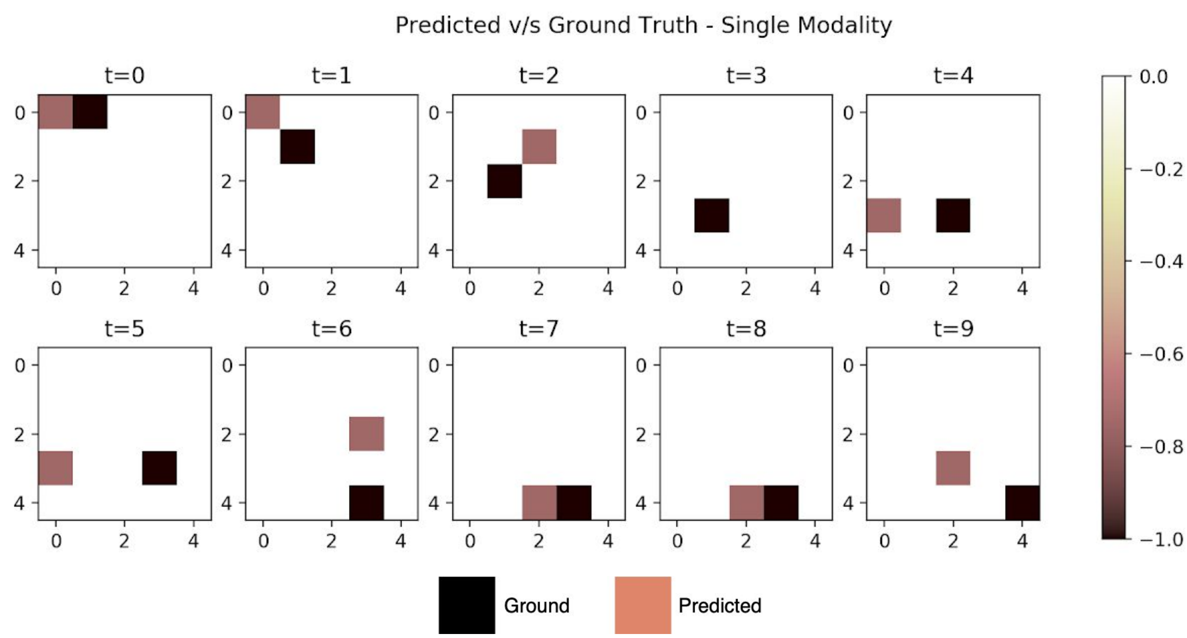

Figure 10: Predicted state v/s Ground state (Filtering) - Only Bump Observations

(2) Prediction

Here, we compute the probability of a future state given the evidence till some previous time step, i.e. $P(X_{t+k} \mid R_{1:t},\ B_{1:t});\ k > 0$

Figure 11 shows the probability distribution over states at the next time-step (in this case, t=10), given all the evidence till time step t=9.

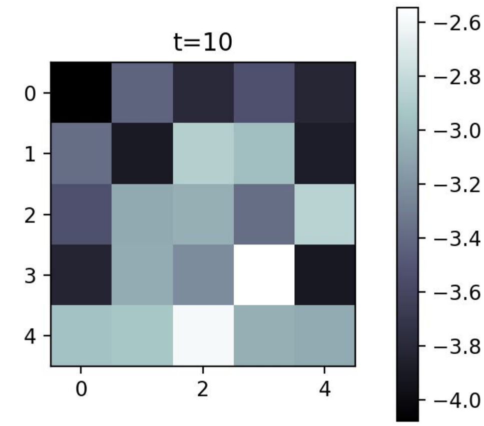

Figure 11: Predictive Distribution over next state given all the evidence

Figure 12 shows the likelihood over states over 5 future time-steps. It can be seen that the probability distribution becomes more spread out at later time steps (like *t+5*) rather than being concentrated around some states, as compared to *t+1.* This makes sense as we are trying to predict future states with limited evidence, and our confidence in the predictions decreases due to increased uncertainty in previously estimated states.

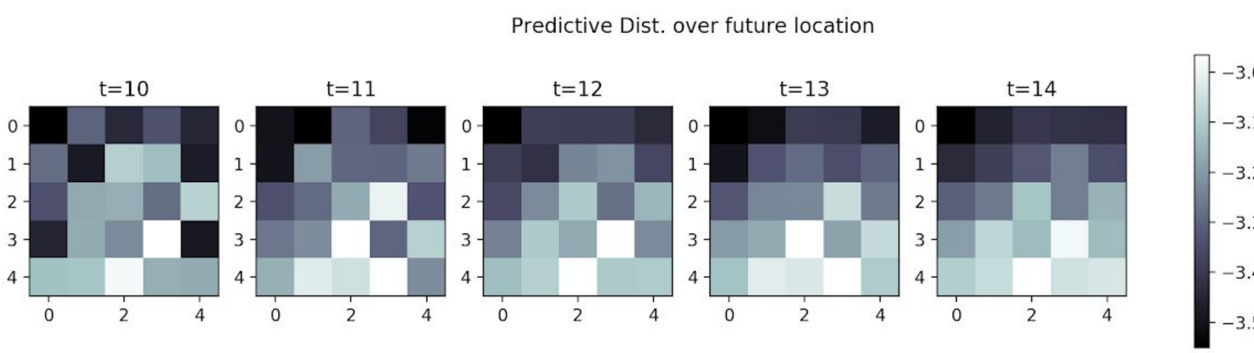

Figure 12: Predictive Distribution over next 5 states

III. Smoothing Task

Smoothing task is of computing the posterior distribution given all the evidence i.e.

$P(X_k \mid R_{1:t}, B_{1:t})\ ;\ 0 \le k < t$

The forward backward algorithm to calculate this distribution.

$$P(X_k = j \mid R_{1:t}, B_{1:t})\ \alpha\ P(X_k = j \mid R_{1:k}, B_{1:k}).\ P(R_{k+1:t}, B_{k+1:t} \mid X_k, R_{1:k}, B_{1:k})$$

$$\beta_j(t) = P(R_{k+1:t}, B_{k+1:t} \mid X_k, R_{1:k}, B_{1:k})$$

$$\beta_i(t) = \begin{cases} 1 & \text{when } t = T \\ \sum_{j=0}^{M} a_{ij} b_{jkv(t+1)} \beta_j(t+1) & \text{when } t \text{ less than } T \end{cases}$$

Figure 13 shows the comparison between ground states and predicted states using forward-backward algorithm. As compared to the predictions from the filtering task, these predictions are much better as we are also using the future evidence for making the prediction - more data helps!

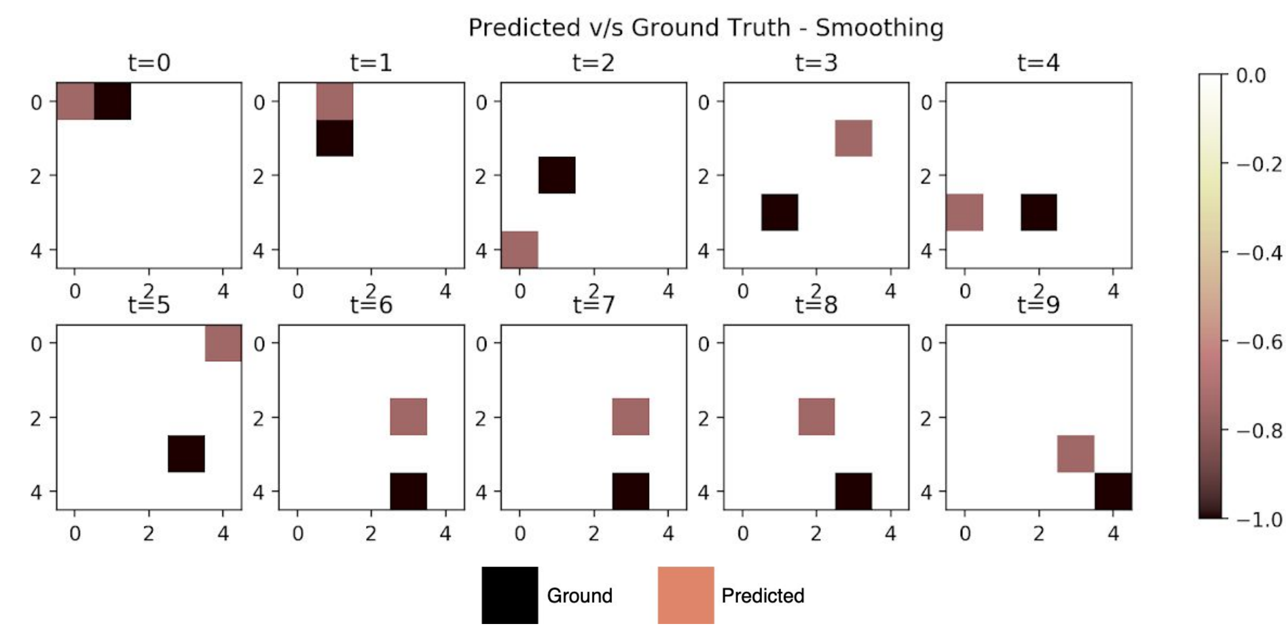

Figure 13: Predicted state v/s Ground state (Smoothing)

IV. Most Likely Path

The Viterbi Algorithm was used to estimate the most likely path taken by the agent, given all the evidence i.e. $argmax_{X_{1:t}} P(X_{1:t} \mid R_{1:t}, B_{1:t})$

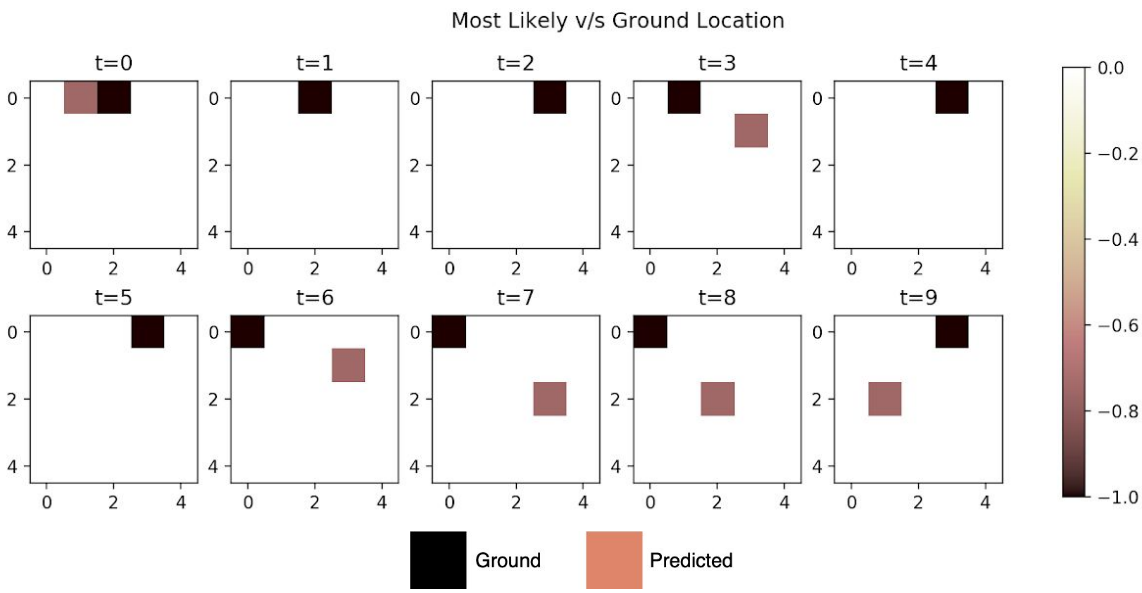

Figure 14: Predicted state v/s Ground state (Most Likely Path)

IV. Effect of State Space Size

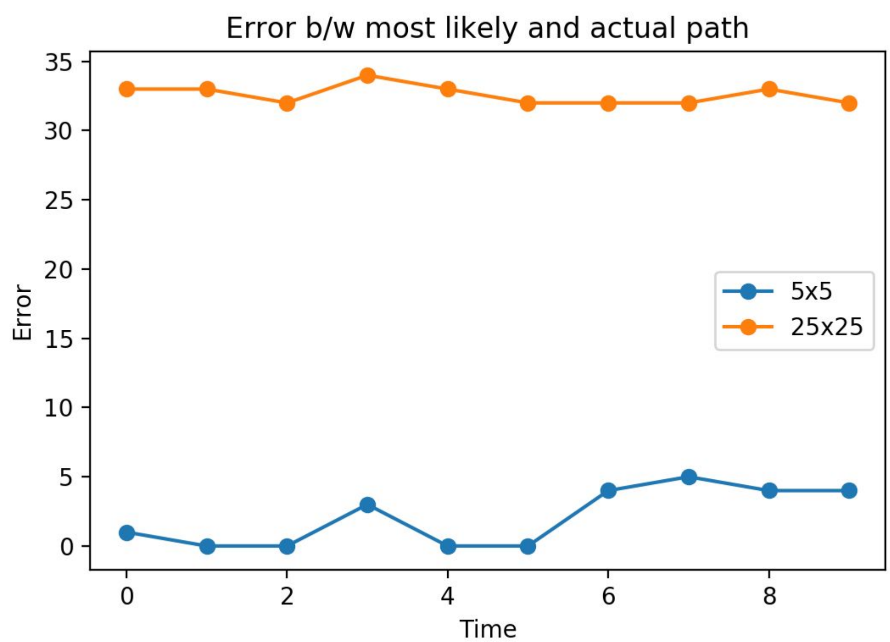

Figure 15: Error(Manhattan Distance) between most likely and actual path

Increase in Error with increase in State Space Size

The Manhattan distance is used to compute the error between Actual and Predicted state for grid sizes 5 and 25. It can be seen that the average error per time step is much higher for more number of states and the total error increases by ~15%

Sum of error for 5x5 grid = 21
Sum of error for 25x25 grid = 326
Relative increase in error = 15.5%

Time complexity with increase in State Space Size

The theoretical complexity of the Viterbi Implementation using Dynamic Programming is linear in number of time steps and square in number of states. Hence, theoretically speaking, we should obtain a relative computation complexity of $\frac{|25|^2}{|5|^2} = 25$