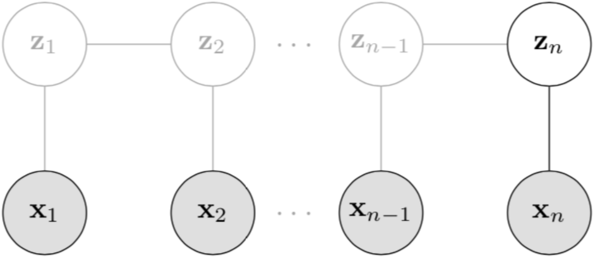
Lab3

ECE 368

Hidden Markov Models

- A Hidden Markov Model (HMM) is graphical model given by a Markov sequence of latent variables (or hidden states), i.e., $z_1, \ldots, z_n$, and a sequence of observations, i.e., $x_1, \ldots, x_n$, one for each hidden state.



- To characterize an HMM, we need: (i) transition probabilities $p(z_{k+1}|z_k)$, (ii) emission probabilities $p(x_k|z_k)$, and (iii) initial distribution $p(z_1)$. These are usually assumed to be known.
- **In this lab, we want to do inference on the hidden states given the observations.**

Inference Algorithm

How to make inferences?

1. **Forward Backward Algorithm:** The goal is to compute the MAP estimate of each state given the observations. For HMM, we can show that:

   $$p(z_k|x_{1:n}) \propto \alpha(z_k)\beta(z_k),$$

   where the forward messages $\alpha(z_k)$ and backward messages $\beta(z_k)$ follow from recursive formulas.

2. In some practical situations, the posterior estimate of individual states does not yield predictions that are consistent with the expected the behavior (e.g., in text recognition).
   Instead, one may be interested in computing the most likely sequence of states given the observations. This can be done efficiently using the **Viterbi Algorithm**, which aims at computing:

   $$\max_{z_{1:n}} \quad p(z_{1:n}|x_{1:n})$$

Problem Setup

- A Mars rover is wandering in a region which is modeled as a grid of width 12 and height 8. The exact location of the rover is unknown, but we have some noisy observations from a sensor.

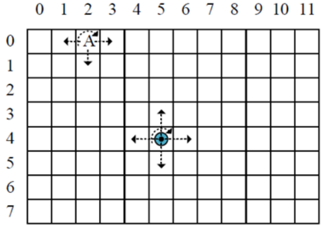


Figure 1: A wandering rover (blue circle) in a grid of width 12 and height 8.

Problem Setup

- We model the rover's **hidden state** z_i at time i as a super variable that includes both the rover's location $(x_i; y_i)$ and its most recent action a_i, where the action space is $\{\text{STAY}, \text{LEFT}, \text{RIGHT}, \text{UP}, \text{DOWN}\}$.

$$z_i = ((x_i; y_i); a_i)$$

- At time i, the observation is given by a pair of noisy measurements $(\hat{x}_i, \hat{y}_i)$. The actions are not observed.

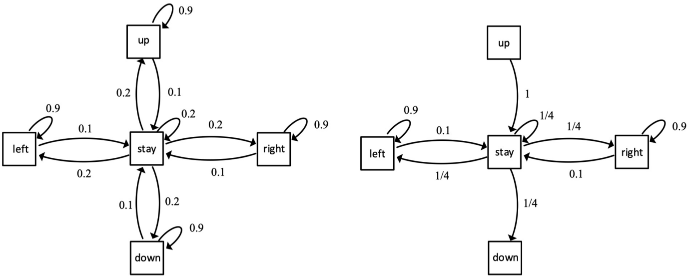
(a) Hidden state of the rover, blue circle indicates the current location (x_i, y_i), and the arrow indicates the most recent action a_i.

(b) Observation $(\hat{x}_i, \hat{y}_i)$, whose value is taken from one of the five possible locations (green circles) with equal probability, given the true location shown in the left figure.

Figure 3: Hidden state and observation.

Problem Setup

- The rover never escapes the grid and its behaviour is described by the following:
  1. If the rover is not on the boundary
     - ★ If the previous action is was a movement, it repeats the previous action with probability 0.9, or stays at the same location with probability 0.1.
     - ★ If the previous action is STAY, it chooses its next action from a uniform distribution.
  2. If the rover is not on the boundary, it will adjust its behavior so it remains consistent with the nonboundary case above.



(a) Transition diagram if the rover is not on the boundary

(b) Transition diagram if the rover is at A

Figure 2: Transition diagrams of rover's behavior

Questions

1.
    1. **Write down the formulas** of the forward-backward algorithm to compute the posterior distribution $p(z_i|(\hat{x}_0, \hat{y}_0), \ldots, (\hat{x}_{N-1}, \hat{y}_{N-1}))$. The formulas include the initialization of the forward and backward messages, the recursion relations of the messages, and the computation of the marginal distribution based on the messages.
    2. **Write a code** to implement the forward-backward algorithm. Use the data in **text.txt** to determine the posterior distributions.
2. Suppose some observations are lost during transmission from Mars to Earth. Modify your existing code to handle missing observations. Use the data in **text_missing.txt** to determine the posterior distributions.
   A missing observation is uninformative: $p\left((\hat{x}_i, \hat{y}_i) \mid z_i\right) = 1$, if $(\hat{x}_i, \hat{y}_i) =$ missing.

Questions

3.
   1. **Write down the formulas** of the **Viterbi algorithm** to compute the most likely trajectory. The formulas include the initialization the messages, the recursion relations of the messages.
   2. **Write a code** to implement the Viterbi algorithm. Your code should handle missing observations. Use the data in **text_missing.txt** to determine the most likely trajectory.

Questions

4. Compute the error probability for the forward-backward algorithm and the Viterbi algorithm.
Inference:

$$\check{\mathbf{z}}_i = \underset{\mathbf{z}_i}{\operatorname{argmax}} p\left(\mathbf{z}_i \mid (\hat{x}_0, \hat{y}_0), \ldots, (\hat{x}_{99}, \hat{y}_{99})\right), \quad i = 0, \ldots, 99$$

$$\{\tilde{\mathbf{z}}_0, \ldots, \tilde{\mathbf{z}}_{99}\} = \underset{\mathbf{z}_0 \ldots \mathbf{z}_{99}}{\operatorname{argmax}} p\left(\mathbf{z}_0, \ldots, \mathbf{z}_{99} \mid (\hat{x}_0, \hat{y}_0), \ldots, (\hat{x}_{99}, \hat{y}_{99})\right)$$

Error probability:

$$\tilde{P}_e = 1 - \frac{\sum_{i=0}^{99} \mathbb{I}\left(\tilde{\mathbf{z}}_i = \dot{\mathbf{z}}_i\right)}{100}$$

$$\check{P}_e = 1 - \frac{\sum_{i=0}^{99} \mathbb{I}\left(\check{\mathbf{z}}_i = \dot{\mathbf{z}}_i\right)}{100}$$

5. Check the posterior estimates of states given by the forward-backward algorithm. Check whether such a sequence of states is consistent with the expected behavior. For axample:

$$
\begin{aligned}
&\vdots \\
\check{\mathbf{z}}_i &= ((1,1),\ \text{stay }) \\
\check{\mathbf{z}}_{i+1} &= ((1,1),\ \text{left }) \\
&\vdots
\end{aligned}
$$

Visualization Tool

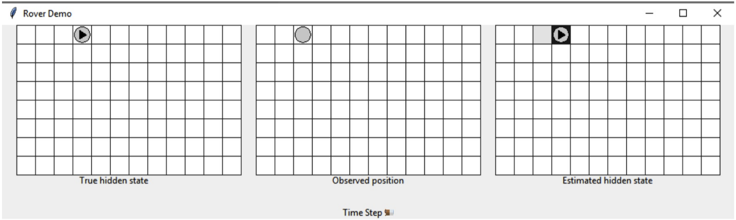


Figure 1: Left: true state. Middle: observation. Right: estimate