ECE368: Probabilistic Reasoning
Lab 3: Hidden Markov Model

Suppose that a Mars rover is wandering in a region which is modeled as a grid of width 12 and height 8, as shown in Fig 1. We do not know the exact location of the rover over time. Instead, we only get some noisy observations about the rover from a sensor. In this lab, we use hidden Markov model to track the rover's movement over time.

The rover's position at time $i = 0, 1, 2, \ldots$ is modeled as a random vector $(x_i, y_i) \in \{0, 1, \ldots, 11\} \times \{0, 1, \ldots, 7\}$. For example, $(x_2, y_2) = (5, 4)$ means that at time step 2, the rover is in column 5, row 4 illustrated as a blue circle in Fig 1. The movement of the rover is quite predictable. At each time step, it makes one of the five actions: it stays put, goes left, goes up, goes right, or goes down.

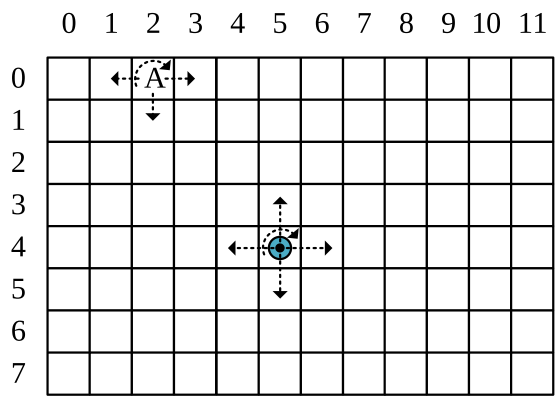
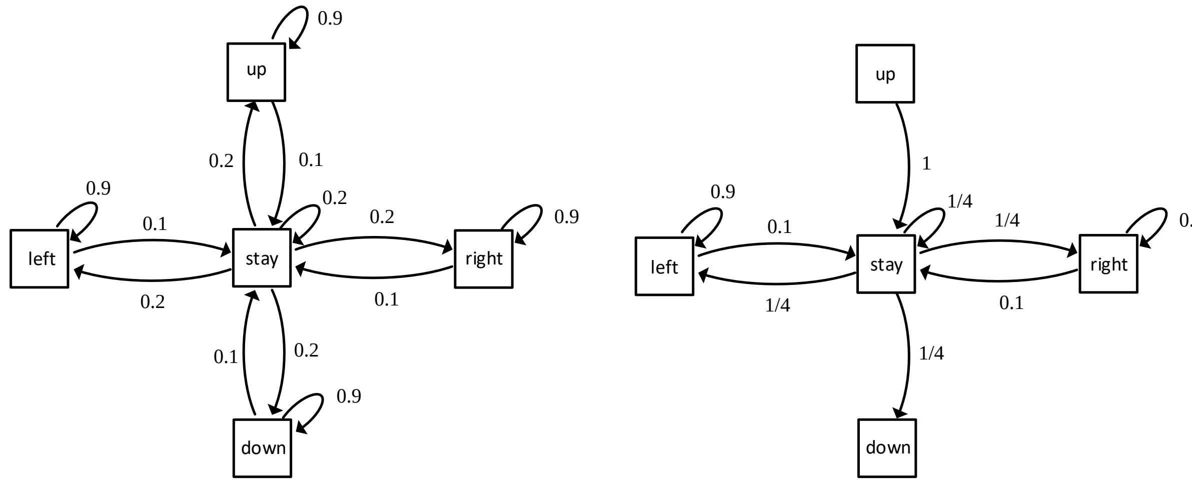


Figure 1: A wandering rover (blue circle) in a grid of width 12 and height 8.

The action of the rover at any time i depends on its previous action as well as its current location. Given that the rover's current location is not at the boundary of the grid, its action is simple: if the rover's previous action was a movement (left, up, right, or down), the rover moves in the same direction with probability 0.9 and stays put with probability 0.1; if the rover's previous action was to stay, it stays again with probability 0.2, and moves with each direction (left, up, right, or down) with probability 0.2. The rover's action can be shown by a transition diagram in Fig. 2a.

In the case that the rover is on the boundary of the grid, the rover's behavior should adjust such that it will not go outside the region, and meanwhile the behavior should also be consistent with the non-boundary case. For example, when the rover is in location A in Fig. 1, the rover cannot go higher. Therefore, there are only four possible actions: it stays, goes left, goes right, or goes down. Based on its previous action, those probabilities may need to be re-normalized such that they sum to 1. Specifically, if the rover's previous action was to go left, the rover moves in the same direction with probability 0.9 and stays put with probability 0.1; if the rover's previous action was to go up, it stays at A with probability 1 (due to re-normalization); if the rover's previous action was to stay, it stays again with probability 0.25, and moves with each direction (left, right, or down) with probability 0.25 (due to re-normalization). The resulting transition diagram is depicted in Fig. 2b.

Since the rover's behavior at any time i depends on its previous action as well as its current location, we model the rover's hidden state $\mathbf{z}_i$ 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, i.e., $\mathbf{z}_i = ((x_i, y_i), a_i)$, where a_i is a random variable that takes the value from



(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

{stay, left, up, right, down}. Pay attention that although we use subscript i in a_i, it actually represents the action of the rover at time $i-1$ (most recent action).

(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.

We do not directly observe the rover's hidden state $\mathbf{z}_i = ((x_i, y_i), a_i)$ as shown in Fig. 3a. Instead, we have access to a noisy sensor that puts a uniform distribution on the valid grid positions within one grid cell of the rover's current true position, as shown in Fig. 3b. Note that when the rover is on the boundary, the possible locations of the observation should adjust such that the observation is in the region. We do not directly observe the rover's action from the sensor, either. In words, at time i the observation is represented by random variable $(\hat{x}_i, \hat{y}_i) \in \{0, 1, \dots, 11\} \times \{0, 1, \dots, 7\}$, and $(\hat{x}_i, \hat{y}_i)$ is uniformly distributed over the possible locations determined by $\mathbf{z}_i$.

Lastly, we assume that the rover's initial position (x_0, y_0) is equally likely to be any of the grid locations, and its initial action a_0 is stay.

Download hmm.zip under Files/Lab3 on Quercus and unzip the file. File rover.py contains functions for generating the initial distribution, the transition probabilities given a current hidden state, and the observation probabilities given a current hidden state. Thus you do not need to re-write these. File test.txt contains the data for Question 1. File test_missing.txt contains the data for Questions 2, 3, 4, and 5. In both test.txt and test_missing.txt, the first three columns correspond to the hidden states, and the last two columns correspond to the observations.

To help you understand what the code does, we provide a visualization tool in inference.py, which can be turned on by setting enable_graphics to True. Note that whether you use the visualization will not affect the grading. When you run inference.py, three panes will pop up. The left pane shows the true state of the rover, including location and the most recent action, represented by an arrow. The middle pane shows the observed position of the rover, with red signifying the missing data. The right pane shows the estimated state of the rover as well as the marginal distribution by grey level. After you implement your inference algorithms, the right pane will automatically show the results.

Please follow the questions below and complete the file inference.py. This is the only file you need to modify.

Questions

1. (a) Write down the formulas of the forward-backward algorithm to compute the marginal distribution $p(\mathbf{z}_i|(\hat{x}_0, \hat{y}_0), \ldots, (\hat{x}_{N-1}, \hat{y}_{N-1}))$ for $i = 0, 1, \ldots, N-1$. The formulas include the initializations of the forward and backward messages, the recursion relations of the messages, and the computation of the marginal distribution based on the messages.

   (b) Complete function forward_backward in file inference.py to implement the forward-backward algorithm. Now use the data $(\hat{x}_0, \hat{y}_0), \ldots, (\hat{x}_{99}, \hat{y}_{99})$ in test.txt to determine the marginal distribution of $\mathbf{z}_i$ at time $i = 99$, i.e., $p(\mathbf{z}_{99}|(\hat{x}_0, \hat{y}_0), \ldots, (\hat{x}_{99}, \hat{y}_{99}))$. Only include states with non-zero probability in your answer.

   Sanity check: The marginal distribution of the state at time $i = 1$ is

$$p(\mathbf{z}_1|(\hat{x}_0, \hat{y}_0), \ldots, (\hat{x}_{99}, \hat{y}_{99})) = \begin{cases} 0.5 & \text{if } \mathbf{z}_1 = ((6,5), \textsf{down}), \\ 0.5 & \text{if } \mathbf{z}_1 = ((6,5), \textsf{right}). \end{cases} \quad (1)$$

2. Some of the observations were lost when they were transmitted from Mars to earth. Modify function forward_backward so that it can handle missing observations. In a list of observations, a missing observation is represented by None in inference.py. Now use the data in test_missing.txt to determine the marginal distribution at time $i = 30$, i.e., $p(\mathbf{z}_{30}|(\hat{x}_0, \hat{y}_0), \ldots, (\hat{x}_{99}, \hat{y}_{99}))$ with missing observations. Only include states with non-zero probability in your answer.

   Sanity check: The mode of this marginal distribution $p(\mathbf{z}_{30}|(\hat{x}_0, \hat{y}_0), \ldots, (\hat{x}_{99}, \hat{y}_{99}))$ should be ((6,7),right) with probability 0.91304.

3. (a) Instead of computing the marginal distributions, we now seek the most likely sequence of the states via the Viterbi algorithm. Write down the formulas of the Viterbi algorithm using $\mathbf{z}_i$ and $(\hat{x}_i, \hat{y}_i), i = 0, \ldots, N-1$. The formulas include the initialization of the messages and the recursion of the messages in the Viterbi algorithm.

   (b) Complete the function Viterbi in file inference.py to implement the Viterbi algorithm. Your implementation should be able to handle missing observations. Use the data in test_missing.txt to determine the last 10 hidden states of the most likely sequence (i.e., $i = 90, 91, 92, \ldots, 99$) based on the MAP estimate obtained from the algorithm.

   Sanity check: For the MAP sequence, the last 3 states, i.e., the states at $i = 97, 98, 99$ are: ((8,7),left), ((7,7),left), ((6,7),left).

4. Let $\tilde{\mathbf{z}}_i, i = 0, 1, \ldots, 99$ be the estimate obtained from Question 3 by Viterbi algorithm, which corresponds to the most probable sequence given the observations:

$$\{\tilde{\mathbf{z}}_0, \ldots, \tilde{\mathbf{z}}_{99}\} = \arg\max_{\mathbf{z}_0, \ldots, \mathbf{z}_{99}} p(\mathbf{z}_0, \ldots, \mathbf{z}_{99}|(\hat{x}_0, \hat{y}_0), \ldots, (\hat{x}_{99}, \hat{y}_{99})). \quad (2)$$

Let $\check{\mathbf{z}}_i, i = 0, 1, \ldots, 99$ be the set of the states obtained from Question 2 by forward-backward algorithm, which are individually the most probable at each time step, corresponding to the maximization of the marginal distribution in Question 2, i.e.,

$$\check{\mathbf{z}}_i = \arg\max_{\mathbf{z}_i} p(\mathbf{z}_i | (\hat{x}_0, \hat{y}_0), \ldots, (\hat{x}_{99}, \hat{y}_{99})), \quad i = 0, \ldots, 99. \tag{3}$$

To compare $\tilde{\mathbf{z}}_i$ and $\check{\mathbf{z}}_i$, we let $\dot{\mathbf{z}}_i, i = 0, 1, \ldots, 99$ be the true hidden states, and define the error probabilities of $\tilde{\mathbf{z}}_i$ and $\check{\mathbf{z}}_i$, respectively, as

$$\tilde{P}_e = 1 - \frac{\sum_{i=0}^{99} \mathbb{I}(\tilde{\mathbf{z}}_i = \dot{\mathbf{z}}_i)}{100}, \tag{4}$$

$$\check{P}_e = 1 - \frac{\sum_{i=0}^{99} \mathbb{I}(\check{\mathbf{z}}_i = \dot{\mathbf{z}}_i)}{100}, \tag{5}$$

where $\mathbb{I}(\cdot)$ is the indicator function, i.e., $\mathbb{I}(X) = 1$, if X is true; otherwise $\mathbb{I}(X) = 0$. Please compute and compare $\tilde{P}_e$ and $\check{P}_e$ for the data in test_missing.txt.

5. Although the states $\check{\mathbf{z}}_i, i = 0, 1, \ldots, 99$, obtained from (3), are individually the most probable states, the sequence $\check{\mathbf{z}}_0, \check{\mathbf{z}}_1, \ldots, \check{\mathbf{z}}_{99}$ may not representant a valid sequence. By a valid sequence, we mean that the rover can behave physically as the sequence $\check{\mathbf{z}}_0, \check{\mathbf{z}}_1, \ldots, \check{\mathbf{z}}_{99}$ as described by the Markov transition diagram of Fig. 2. For example,

$$\vdots \tag{6}$$

$$\check{\mathbf{z}}_i = ((1,1),\text{stay}) \tag{7}$$

$$\check{\mathbf{z}}_{i+1} = ((1,1),\text{left}) \tag{8}$$

$$\vdots \tag{9}$$

is not a valid sequence because the transition from state ((1,1),stay) at time step i to state ((1,1),left) at time step $i+1$ is impossible according to the transition model. Please check $\check{\mathbf{z}}_0, \check{\mathbf{z}}_1, \ldots, \check{\mathbf{z}}_{99}$ in Question 4 to see whether or not it is a valid sequence. If not, please find a small segment $\check{\mathbf{z}}_i, \check{\mathbf{z}}_{i+1}$ that violates the transition model for some time step i.

We thank Prof. Greg Wornell of MIT in creating this lab.