AIL 722: Assignment 1
Semester I, 2024-25

Due 25th August 2024 11:55 PM

Instructions:

- This assignment has four implementation questions.
- You should submit all your code (including any pre-processing scripts written by you) and any graphs that you might plot.
- Include **a write-up (pdf) file**, which includes a brief description for each question explaining what you did. Include any observations and/or plots required by the question in this single write-up file.
- Please use Python for implementation using only standard python libraries (e.g. numpy). Do not use any third party libraries/implementations for algorithms.
- Your code should have appropriate documentation for readability.
- You will be graded based on what you have submitted as well as your ability to explain your code.
- Refer to the course website for assignment submission instructions.
- This assignment is supposed to be done individually. You should carry out all the implementation by yourself.
- We plan to run Moss on the submissions. We will also include submissions from previous years since some of the questions may be repeated. Any cheating will result in a zero on the assignment, a penalty of -10 points and possibly much stricter penalties (including a **fail grade** and/or a **DISCO**).
- Starter code is available at this link.

1 Sailing Away

In this assignment, you will implement sampling and likelihood estimation from Hidden Markov Models (HMMs). Consider a grid world of size 15×15; there is a sailboat being tossed about by wind in this grid world. If the sailboat is at position (i, j), we denote its state as $s^{i,j}$. Here, i and j represent the x and y coordinates of the sailboat, respectively. The state set is:

$$S = \{s^{i,j} \mid 1 \leq i \leq 15, 1 \leq j \leq 15\}$$

1.1 Transition Model

We model the sailboat motion under random wind with the Markov assumption, implying that its transition to the next position is conditionally independent of history given current position. The following is the transition probability of the sailboat under the wind:

$$P(s_{t+1} = s^{i',j'} \mid s_t = s^{i,j}) = \begin{cases} 0.4 & \text{if } i' = i+1 \text{ and } j' = j \text{ i.e. going right} \\ 0.3 & \text{if } i' = i \text{ and } j' = j+1 \text{ i.e. going up} \\ 0.1 & \text{if } i' = i \text{ and } j' = j-1 \text{ i.e. going down} \\ 0.1 & \text{if } i' = i-1 \text{ and } j' = j \text{ i.e. going left} \\ 0.1 & \text{if } i' = i \text{ and } j' = j \text{ i.e. staying in the same position} \end{cases} \quad (1)$$

If $s^{i',j'} \notin S$, then the sailboat stays in $s^{i,j}$ with the additional probability. For example if the position of the sailboat is $(1,4)$ then it cannot move to $(0,4)$ as $s^{0,4} \notin S$. Instead it stays at $(1,4)$ with probability, $P(s_{t+1} = s^{1,4} \mid s_t = s^{1,4}) = 0.1 + 0.1 = 0.2$.

The sailboat always starts at the position (1,1).

1.2 Observation Model

Suppose we have 4 different sensors that give us (noisy) information about the sailboat's location. The observations are received from these four sensors placed at the four corners of the grid, as shown in the figure below. Each sensor independently detects the presence or absence of the sailboat. The sensors are noisy, and their chances of detecting the sailboat are high if the sailboat is close to them. Note that each sensor outputs only either 0 or 1.

The readings of the sensors at time step t are denoted as $o_t^1, o_t^2, o_t^3, o_t^4$. Here, o_t^i is 1 if the i^{th} sensor detects the robot at time-step t, otherwise 0.

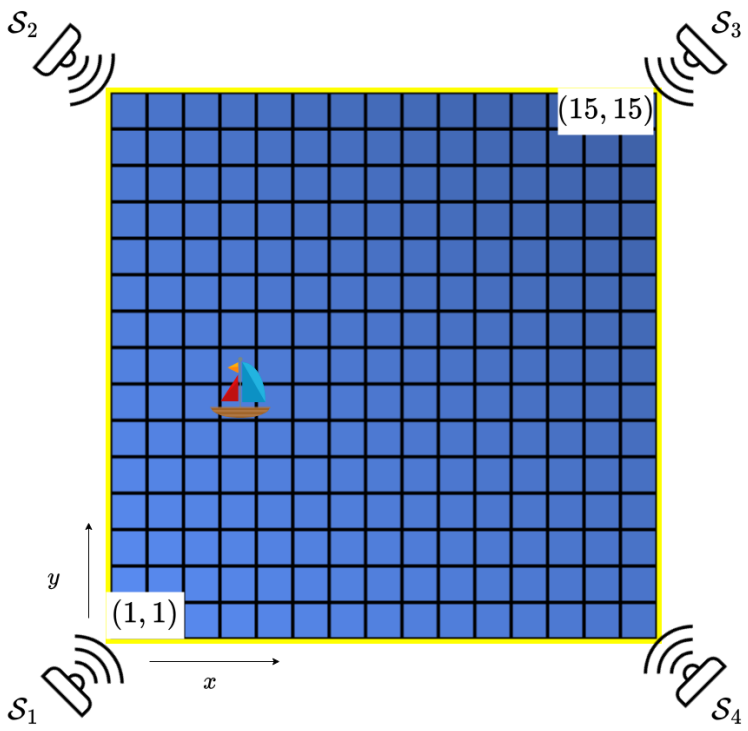

Figure 1: Sensor positions in the grid world

- Sensor at (1,1) (bottom left corner):

$$P(o_t^1 = 1 \mid s_t^{i,j}) = \begin{cases} \frac{18-(i-1)-(j-1)}{18} & \text{if } 1 \le i \le 9 \text{ and } 1 \le j \le 9 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

- Sensor at (1,15) (top left corner):

$$P(o_t^2 = 1 \mid s_t^{i,j}) = \begin{cases} \frac{18-(i-1)+(j-15)}{18} & \text{if } 1 \le i \le 9 \text{ and } 7 \le j \le 15 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

- Sensor at (15,15) (top right corner):

$$P(o_t^3 = 1 \mid s_t^{i,j}) = \begin{cases} \frac{18+(i-15)+(j-15)}{18} & \text{if } 7 \le i \le 15 \text{ and } 7 \le j \le 15 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

- Sensor at (15,1) (bottom right corner):

$$P(o_t^4 = 1 \mid s_t^{i,j}) = \begin{cases} \frac{18+(i-15)-(j-1)}{18} & \text{if } 7 \le i \le 15 \text{ and } 1 \le j \le 9 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The sensor coverage is defined as the set of states where the sensors have a non-zero detection probability. Specifically, this is represented by $\texttt{mask}_i[x, y]$ which is defined as $\texttt{mask}_i[x, y] = \texttt{Ind}(P(o_t^i = 1|s_t^{x,y}) > 0)$. Here `Ind` is an indicator function that equals 1 if the condition inside is true and 0 otherwise.

2 Questions

1. **Sample Trajectories and Observations** **[20 marks]**
   (a) Visualize the sensor detection probabilities on the grid. The plotting code is provided to you (see the last point of the instructions to access the starter code).
   (b) Sample 20 state trajectories of length 30 each in this grid world. Ensure reproducibility by using seed values for the probabilistic execution.
   (c) Plot the sampled trajectories on the grid world, clearly indicating the starting position, end position and paths taken.
   (d) For these 20 trajectories, sample the sensor observations and report them. The sensor observations should be based on the defined sensor model.

2. **Likelihood Estimation** **[20 marks]**
   (a) Calculate and report the likelihood of the 20 sampled observation trajectories obtained from the sensors using the Forward algorithm.
   (b) Provide a brief comment on the obtained likelihood values. What inferences can you draw from these likelihoods regarding the observed state sequences?

3. **Decoding** **[30 marks]**
   (a) Decode the observed sensor readings for the above 20 trajectories using the Viterbi algorithm.
   (b) Report the mean Manhattan distance between the predicted and the true state sequences vs the trajectory time-step.
   (c) Plot the true and decoded state sequences for all the observations, showing the discrepancies between the two where applicable.

4. **Learning Parameters** **[30 marks]**

   Sample $R = 10,000$ sensor observation trajectories of length $T = 20$, assuming a fixed initial distribution ρ. Initialize the emission matrix $\mathcal{B}$ and the state transition matrix $\mathcal{T}$ with uniform observation and transition state probabilities. For all the three parts, use the factored representation of $\mathcal{T}$ based on an underlying structure. Perform 20 iterations of the algorithm, plotting the KL divergence between the true and predicted matrices at each iteration, and report the final KL divergence. Following are the equations for estimating KL divergence of $\hat{\mathcal{T}}$ and $\hat{\mathcal{B}}$:

$$\texttt{AvgKL}(T, \hat{T}) = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}} \mathcal{T}[i, j]\texttt{log}(\mathcal{T}[i, j]) - \mathcal{T}[i, j]\texttt{log}(\hat{\mathcal{T}}[i, j]) \tag{6}$$

$$\texttt{AvgKL}(B, \hat{B}) = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{O}} \mathcal{B}[i, j]\texttt{log}(\mathcal{B}[i, j]) - \mathcal{B}[i, j]\texttt{log}(\hat{\mathcal{B}}[i, j]) \tag{7}$$

   Following are the updated M-step equation for $\mathcal{T}$:
   For every position (x, y) of sailboat define $\mathcal{N}(x, y) = \{(x, y), (x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)\}$ then

$$\hat{P}_{right} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T-1} \sum_{(x,y)=(1,1)}^{14,15} \xi_{r,t}((x, y), (x + 1, y))}{\sum_{r=1}^{R} \sum_{t=1}^{T-1} \sum_{(x,y)=(1,1)}^{14,15} \sum_{(x',y') \in \mathcal{N}(x,y)} \xi_{r,t}((x, y), (x', y'))} \tag{8}$$

   Similarly $\hat{P}_{left}, \hat{P}_{top}, \hat{P}_{bottom}, \hat{P}_{same}$ can also be obtained. $\mathcal{T}$ is constructed using these probabilities. The equivalent function in starter code is `create_T`.

(a) Implement the Baum-Welch algorithm to estimate the state transition matrix $\mathcal{T}$, keeping the emission matrix $\mathcal{B}$ fixed at its true value.

(b) Extend the Baum-Welch algorithm to jointly estimate both the state transition matrix $\mathcal{T}$ and the emission matrix $\mathcal{B}$. The M-step equation for $\mathcal{B}$ is

$$\hat{P}(o_t = o|(x,y)) = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T} \texttt{Ind}(o_t = o)\gamma_{r,t}((x,y))}{\sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{r,t}((x,y))} \tag{9}$$

(c) Assume that the sensors are independent with specific coverage areas. Instead of learning a single emission matrix, estimate separate sensor-specific emission matrices $\mathcal{B}_i$. During each iteration of the Baum-Welch algorithm, update and mask the sensor-specific matrices according to their coverage areas. After learning, visualize the sensor-specific matrices on a grid. Sensor independence implies:

$$\hat{P}(o_t = (o_t^1, o_t^2, o_t^3, o_t^4)|(x,y)) = \hat{P}(o_t^1|(x,y))\hat{P}(o_t^2|(x,y))\hat{P}(o_t^3|(x,y))\hat{P}(o_t^4|(x,y)) \tag{10}$$

The equivalent function in the starter code is `create_B`. The individual components can be computed as:

$$\tilde{P}(o_t^1 = o|(x,y)) = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T} \texttt{Ind}(o_t^1 \in o_t)\gamma_{r,t}((x,y))}{\sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{r,t}((x,y))} \tag{11}$$

Where $\texttt{Ind}(o_t^1 \in o)$ is 1 if the first sensor is active in o_t else 0. Mask $\tilde{P}(o_t^1|(x,y))$ according to $\texttt{mask}_1$ (defined in section 1.2) to get $\hat{P}(o_t^1|(x,y))$

$$\hat{P}(o_t^1|(x,y)) = \texttt{mask}_1[x,y]\tilde{P}(o_t^1|(x,y)) \tag{12}$$