Machine Learning for Data Science - Competition 2

Antoine Klopocki, Jaydev Kshirsagar, Travis Westura, Vishisht Tiwari
Kaggle Team Name: Antravishjay

1 Introduction

In this competition we are given a robot that is moving around in $\mathbb{R}^2$. The robot makes 10,000 runs, and for 1,000 timesteps we observe the angle θ that the robot's position makes with the x-axis. We are not given the robot's precise (x, y) location.

2 Determining the Observer's Position

The competition specification describes the robot moving in the first quadrant, and the angles given in the observations file contain the angle made between the robot's location $(x_{r,t}, y_{r,t})$ and the x-axis. However, the coordinates given in the label file contain coordinates with negative values. We'll denote to these values as $(x'_{r,t}, y'_{r,t})$. Since these values are negative, the observer of the locations is positioned in the first quadrant as well, and we call it's location (a, b). First we need to determine the coordinates of this point so that we can match up the data in the observation and label files.

Consider a labeled point $(x'_{r,t}, y'_{r,t})$. The angle θ that this point forms with the x-axis is given by

$$\tan\theta = \frac{b + y'_{r,t}}{a + x'_{r,t}}.$$

We know the value of θ from the observations file. Since there are two unknowns a and b, we pick two labeled points and solve the system of equations

$$\tan\theta_1 = \frac{b + y'_{r_1,t_1}}{a + x'_{r_1,t_1}}, \quad \tan\theta_2 = \frac{b + y'_{r_2,t_2}}{a + x'_{r_2,t_2}}.$$

We use the points from run 1 time steps 205 and 216. Solving the system of equations yields the position of the location observer as $(1.5, 1.5)$. Using this position, we consider the observation angles as being measured at $(0, 0)$.

Further, we can use the 600,000 labeled points to gain intuition about the robot's movement. We compute the average of radius of these points to $(1.5, 1.5)$ to see that this average is approximately 1. Plotting the labeled points in figure 1, we see that the robot is moving roughly in a circle of radius 1 centered at $(1.5, 1.5)$, with kinks occurring at the top, bottom, left, and right of the circle.

Every run starts the robot at $(1.5, 1.5)$. But we are not given any labels for the first 200 runs, and we see that the robot stabilizes into its standard path by the time we start obtaining labels for its location.

3 Hidden Markov Models (HMM's)

We represent this problem as a Hidden Markov Model. A Hidden Markov Model is a graphical model with two types of variables: latent variables S_t and observed variables X_t. The latent variables represent states that are not visible to the observer. Each observed variable X_t depends only on the corresponding state variable S_t. And each state variable S_t depends only on the previous state variable S_{t-1}. We depict latent variables as shaded vertices.

In our problem the observed values are the angles that the robot's position makes with the x-axis. The robot is moving around in the plane $\mathbb{R}^2$. This space is continuous, so our observations are angles in the continuous range $\left(0, \frac{\pi}{2}\right)$. We discretize this space in order to use a Hidden Markov Model. We divide the first quadrant into K sectors, $\left[0, \frac{1}{K}\frac{\pi}{2}\right), \left[\frac{1}{K}\frac{\pi}{2}, \frac{2}{K}\frac{\pi}{2}\right), \ldots, \left[\frac{K-1}{K}\frac{\pi}{2}, \frac{\pi}{2}\right)$, where K is a parameter that we choose. The observation, rather than being a value in $\left(0, \frac{\pi}{2}\right)$, is instead given by an integer $1, 2, \ldots, K$ representing the segment of the interval in which the angle lies. As a further refinement of this technique, we find the minimum and maximum angles that occur in the observations file, $\theta_{\min}$ and $\theta_{\max}$, and divide the shorter interval $[\theta_{\min}, \theta_{\max}]$.

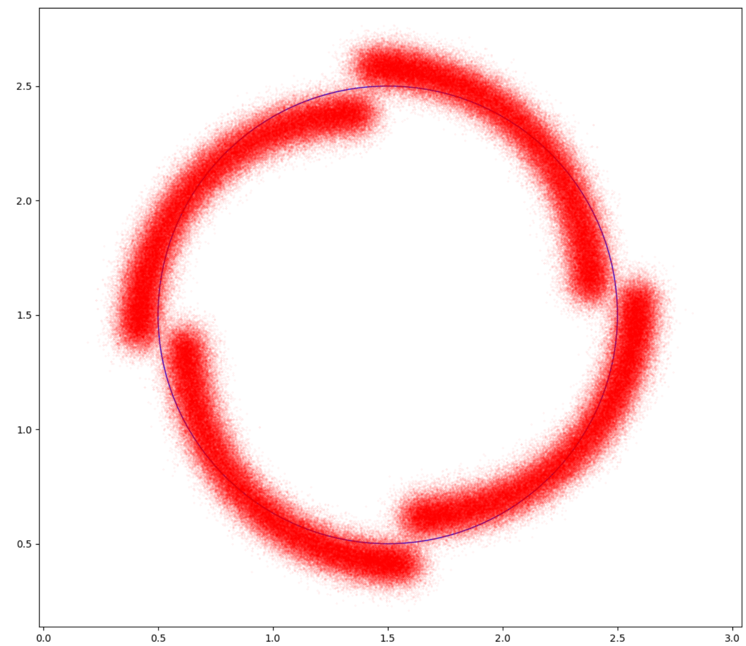


Figure 1: Locations of the robot given as labels.

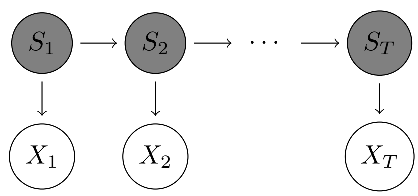


Figure 2: Hidden Markov Model

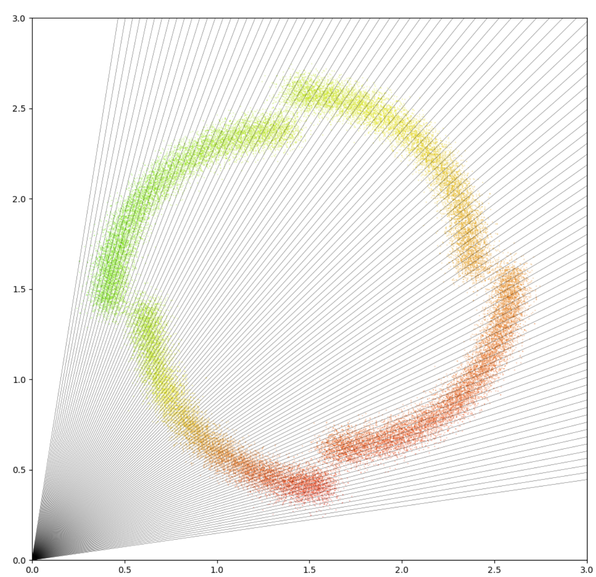


Figure 3: Labeled points colored based on the segment in which they lie with $K = 100$.

Given an observation and the corresponding state, we need to map the state to the robot's position. We outline a procedure for doing this in .

Given the observations, we need to estimate the transition matrix A and emission matrix B of the Hidden Markov Model. The transition matrix is defined by

$$a_{i,j} = \mathbf{P}(S_t = j \mid S_{t-1} = i),$$

that is, each entry $a_{i,j}$ gives the probability of being in state i given that the previous state is state j. With N states, A is an $N \times N$-matrix. The emission matrix is defined by

$$b_{i,k} = \mathbf{P}(X_t = k \mid S_t = i),$$

that is, each entry $b_{i,k}$ give the probability of the observation being k given state i. With N states and K possible observations, B is an $N \times K$-matrix. In section 4 we describe our process for estimating these matrices using the Baum Welch algorithm.

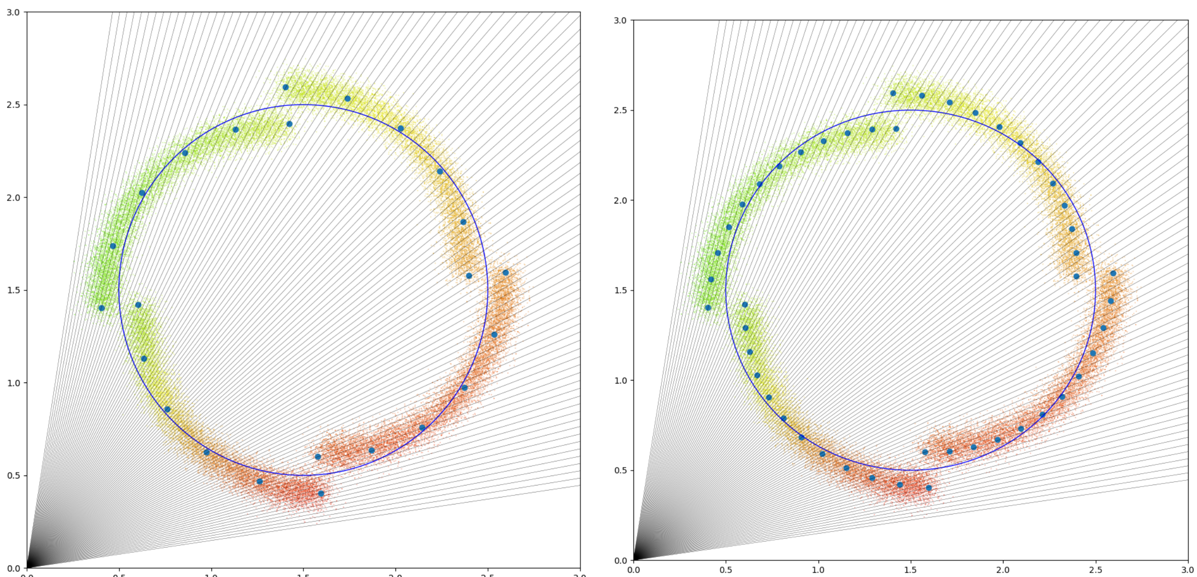
4 Baum Welch

The Baum Welch algorithm is an Expectation Maximization (EM) algorithm for learning the parameters of Hidden Markov Models. We use a forward-backwards algorithm for the expectation step and then update the HMM parameters in the maximization step. We thereby find the maximum likelihood estimate of the parameters of the model. The algorithm takes as parameters a triple (A, B, π), where A and B are the transition and emission matrices and π is the initial state distribution, that is, the probability of the first state being state i is given by $\pi_i = \mathbf{P}(S_1 = i)$.

$$\pi_i := \gamma_i(1)$$

$$a_{i,j} := \frac{\sum_{t=1}^{T-1} \xi_{i,j}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

$$b_{i,k} := \frac{\sum_{t=1}^{T} \mathbf{1}_{X_t=k} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$



(a) N = 24.

(b) N = 48.

Figure 4: Initial state positions for different numbers of states.

5 Viterbi