1 RANSAC with Ellipse I (18)

In this exercise, you are tasked to detect an ellipse based on given dataset using RANSAC.

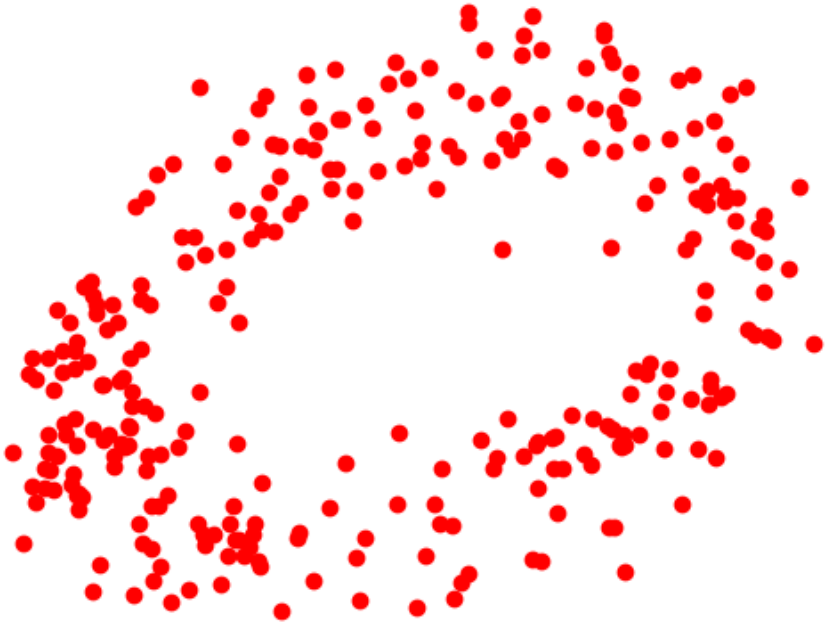

Figure 1: A Plot of Randomly Selected Dataset

The goal is to estimate the parameters $\mathbf{Q}$ and $\mathbf{b}$ by fitting the sensor data to an ellipse. An ellipse can be described by the quadratic equation:

$$(\mathbf{p}-\mathbf{b})^\top \mathbf{Q}(\mathbf{p}-\mathbf{b}) = 1$$

or

$$Ax^2 + 2Bxy + Cy^2 - 2Dx - 2Ey + 1 = 0$$

where:

- $\mathbf{Q} = \mathbf{Q}^\top \in \mathbb{R}^{2\times 2}$ is a unique symmetric positive definite matrix of coefficients. [If $\mathbf{Q}$ is not positive definite, you may have a hyperbola instead.]

Note that the coefficients (A,B,C,D,and E) are also unique. The conversion from the polynomial coeffcients to the quadratic form can be expressed as follows:

$$\mathbf{Q} = \alpha \begin{bmatrix} A & B \\ B & C \end{bmatrix}$$

$$\mathbf{Qb} = \alpha \begin{bmatrix} D \\ E \end{bmatrix}$$

$$\alpha = \frac{1}{\begin{bmatrix} D & E \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix}^{-1} \begin{bmatrix} D \\ E \end{bmatrix} - 1}$$

a.) Determine the minimum number of points to uniquely determined ellipse's parameters as well as derive a method to construct an ellipse with the parameters (A,B,C,D, and E) based on those points. $\mathbf{p}_i = (x_i, y_i)$ denote the i^{th} data point. You must explain your rationale to receive credits. [10 pts]

Answer:

The general form of an ellipse is:

$$Ax^2 + 2Bxy + Cy^2 - 2Dx - 2Ey + 1 = 0$$

From this equation, it is evident that with five unknowns, five points are needed to uniquely determine an ellipse's parameters. Each point is written as:

$$p_i = (x_i, y_i)$$

Substituting $x = x_i$ and $y = y_i$ for $i = 1, 2, 3, 4, 5$ into the ellipse equation. This results in a system of five linear equations:

$$\begin{cases} Ax_1^2 + 2Bx_1y_1 + Cy_1^2 - 2Dx_1 - 2Ey_1 + 1 = 0 \\ Ax_2^2 + 2Bx_2y_2 + Cy_2^2 - 2Dx_2 - 2Ey_2 + 1 = 0 \\ Ax_3^2 + 2Bx_3y_3 + Cy_3^2 - 2Dx_3 - 2Ey_3 + 1 = 0 \\ Ax_4^2 + 2Bx_4y_4 + Cy_4^2 - 2Dx_4 - 2Ey_4 + 1 = 0 \\ Ax_5^2 + 2Bx_5y_5 + Cy_5^2 - 2Dx_5 - 2Ey_5 + 1 = 0 \end{cases}$$

Rewriting the above system of linear equations:

$$\begin{bmatrix} x_1^2 & 2x_1y_1 & y_1^2 & -2x_1 & -2y_1 \\ x_2^2 & 2x_2y_2 & y_2^2 & -2x_2 & -2y_2 \\ x_3^2 & 2x_3y_3 & y_3^2 & -2x_3 & -2y_3 \\ x_4^2 & 2x_4y_4 & y_4^2 & -2x_4 & -2y_4 \\ x_5^2 & 2x_5y_5 & y_5^2 & -2x_5 & -2y_5 \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

Solving:

$$\begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = \begin{bmatrix} x_1^2 & 2x_1y_1 & y_1^2 & -2x_1 & -2y_1 \\ x_2^2 & 2x_2y_2 & y_2^2 & -2x_2 & -2y_2 \\ x_3^2 & 2x_3y_3 & y_3^2 & -2x_3 & -2y_3 \\ x_4^2 & 2x_4y_4 & y_4^2 & -2x_4 & -2y_4 \\ x_5^2 & 2x_5y_5 & y_5^2 & -2x_5 & -2y_5 \end{bmatrix}^{-1} \cdot \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

b.) Fit an ellipse with the following datasets and compute for **Q**, **b**, A, B, C, D and E.

$$(2.92, -6.01), \quad (3.40, -7.20), \quad (4.99, -7.84), \quad (5.48, -7.04), \quad (4.20, -5.91)$$

You don't have code for this part, but you still have to do it with Python in the next part. [8 pts.]

Answer: Plugging in the points given above into the matrix, we can obtain:

$$\begin{bmatrix} (2.92)^2 & 2(2.92)(-6.01) & (-6.01)^2 & -2(2.92) & -2(-6.01) \\ (3.40)^2 & 2(3.40)(-7.20) & (-7.20)^2 & -2(3.40) & -2(-7.20) \\ (4.99)^2 & 2(4.99)(-7.84) & (-7.84)^2 & -2(4.99) & -2(-7.84) \\ (5.48)^2 & 2(5.48)(-7.04) & (-7.04)^2 & -2(5.48) & -2(-7.04) \\ (4.20)^2 & 2(4.20)(-5.91) & (-5.91)^2 & -2(4.20) & -2(-5.91) \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} 8.5264 & -35.0984 & 36.1201 & -5.84 & 12.02 \\ 11.56 & -48.96 & 51.84 & -6.80 & 14.40 \\ 24.9001 & -78.2432 & 61.4656 & -9.98 & 15.68 \\ 30.0304 & -77.1584 & 49.5616 & -10.96 & 14.08 \\ 17.64 & -49.644 & 34.9281 & -8.40 & 11.82 \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

Solving the system of equations yields:

$$\begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = \begin{bmatrix} 0.01806 \\ 0.01207 \\ 0.03015 \\ -0.00625 \\ -0.15441 \end{bmatrix}$$

Finding **Q** and **b**

$$\alpha = \frac{1}{\begin{bmatrix} D & E \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix}^{-1} \begin{bmatrix} D \\ E \end{bmatrix} - 1}$$

$$\alpha = \frac{1}{\begin{bmatrix}(-0.00625) & (-0.15441)\end{bmatrix}\begin{bmatrix}(0.01806) & (0.01207)\\(0.01207) & (0.03015)\end{bmatrix}^{-1}\begin{bmatrix}(-0.00625)\\(-0.15441)\end{bmatrix} - 1} = 41.67811288$$

$$\mathbf{Q} = \alpha\begin{bmatrix}A & B\\B & C\end{bmatrix} = 41.67811288\begin{bmatrix}0.01806 & 0.01207\\0.01207 & 0.03015\end{bmatrix} = \begin{bmatrix}0.75282437 & 0.50302811\\0.50302811 & 1.25673065\end{bmatrix}$$

$$\mathbf{b} = \alpha\mathbf{Q}^{-1}\begin{bmatrix}D\\E\end{bmatrix} = 41.67811288\begin{bmatrix}0.75282437 & 0.50302811\\0.50302811 & 1.25673065\end{bmatrix}^{-1}\begin{bmatrix}-0.00625\\-0.15441\end{bmatrix} = \begin{bmatrix}4.19868909\\-6.80150371\end{bmatrix}$$

2 RANSAC with Ellipse II (30)

In this exercise, you are tasked to implement RANSAC algorithm for determining the shape of an ellipse in Python. The implementation must match your explanation in the report in order to receive full credits.

To test whether point $\mathbf{p}_i$ is an inlier, let the following inequality be the condition for inliers.

$$\left\| \sqrt{(\mathbf{p}_i - \mathbf{b})^\top \mathbf{Q} (\mathbf{p}_i - \mathbf{b})} - 1 \right\| \leq \varepsilon$$

where ε is the tunable threshold.

a.) Based on the given template in Python, implement *fit_ellipse_subset* that compute $\mathbf{Q}$ and $\mathbf{b}$ from a given set of points. (The number points is determined from 1.a). [8 pts]
Answer:
Plotting the ellipse from Question 1 yields:

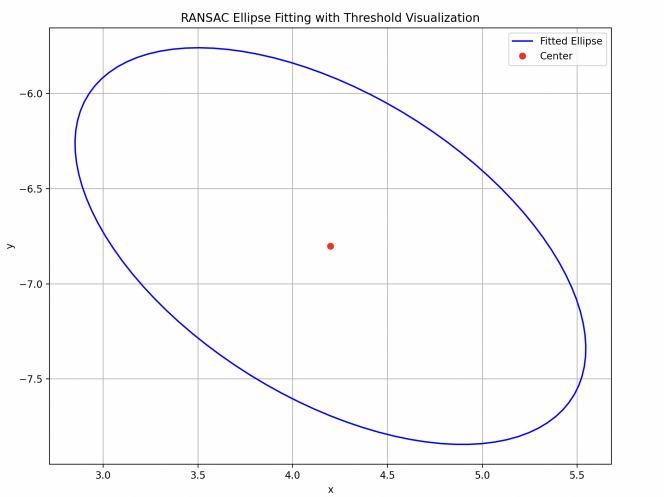

Figure 2: Question 1 Ellipse

b.) Based on the given template in Python, implement *ransac_ellipse* that determined the best $\mathbf{Q}$, $\mathbf{b}$, and the corresponding inliers based on the given dataset, number of iterations, and threshold. You must explain your implementation step-by-step [12 pts] with the rationale in your report.
Answer:
Step 1:
Define three variables: 1. Best set of inliers 2. Best Q matrix 3. Best b vector These will be updated according based on the error (distnace to inliers) of the fitted ellipse

Step 2:
Create a `for` loop that runs the RANSAC algorithm for the number of iterations specified by `num_iterations` and given a certain threshold.

Within the loop:

- Reset the `inliers` array to be empty.
- Select 5 random points from the input data.
- Use these 5 points to fit an ellipse, yielding $\mathbf{Q}$ and $\mathbf{b}$.
- Determine the set of inliers corresponding to these 5 points based on the obtained $\mathbf{Q}$ and $\mathbf{b}$ matrices (detailed in further steps)

Step 3:
Create an internal `for` loop to iterate through each point in the dataset. For each point, calculate its distance to the generated ellipse using the equation provided. If the calculated distance is less than the threshold, add the point to the `inliers` arary.

Step 4:
Make sure the ellipse is not too eccentric and the matrix **Q** is positive definite. If both of these conditions are not met, the **Q**, **b**, and inlier set are not used, even if the number of inliers found in the iteration was the highest so far.

Step 5:
If both of the conditions in **Step 4** are met, then compare the length of the current `inliers` array with that of the longest inliers array found so far. If the length of the current inliers array is longer, update `best_inliers`, `best_Q`, and `best_b` accordingly. This concludes the algorithm.

Step 6:
Format the variables to be returned, then return them.

c.) Based on the given template in Python, use different number of dataset, number of iterations, and threshold value. Comment on the effect of these parameters. [10 pts]

Answer: To evaluate the effects of each parameter, 3 different values of a single parameter were used, while leaving the other two parameters at their default values (dataset size = 500 points, number of iterations = 1000, threshold = 0.2). **Assumption**: The underlying nature of the data was randomly sample with noise from an ellipse.

Varying Dataset Size: Experimenting with dataset sizes of 500, 1000, and 2000

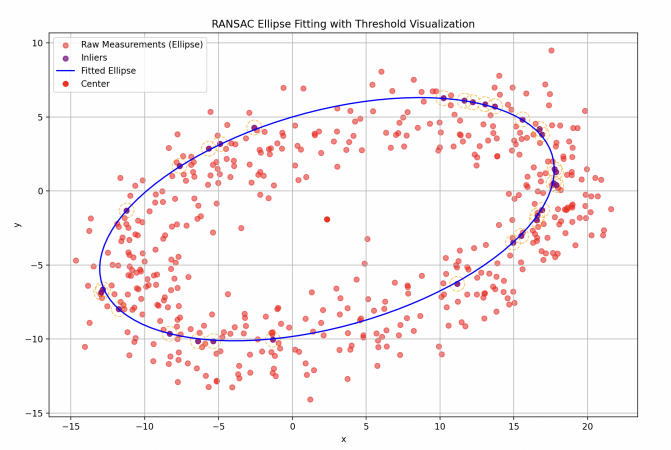

Figure 3: N = 500, Iterations = 1000, Threshold = 0.2

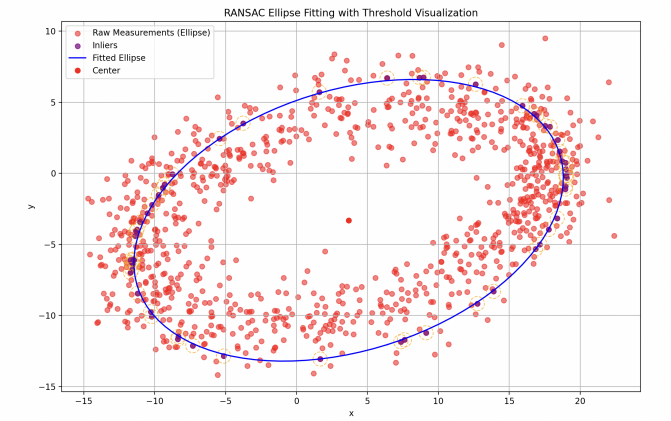

Figure 4: N = 1000, Iterations = 1000, Threshold = 0.2

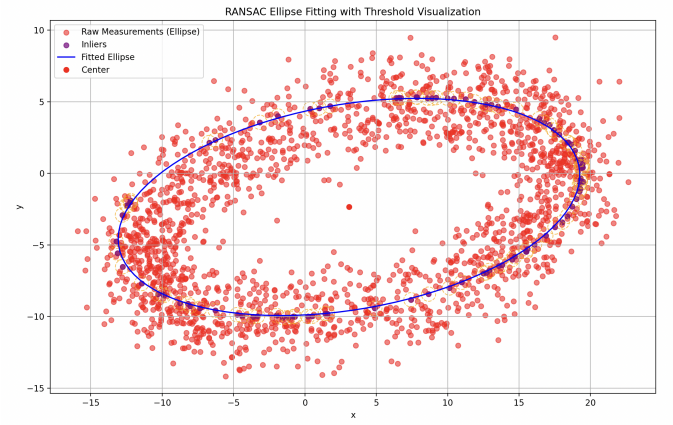

Figure 5: N = 2000, Iterations = 1000, Threshold = 0.2

It is evident that the higher the number of iterations, the more inliers there are, meaning that the parameters of the fitted ellipse more closely resemble the true parameters.

Varying Number of Iterations: Experimenting with iterations of 1000, 2000, 4000:

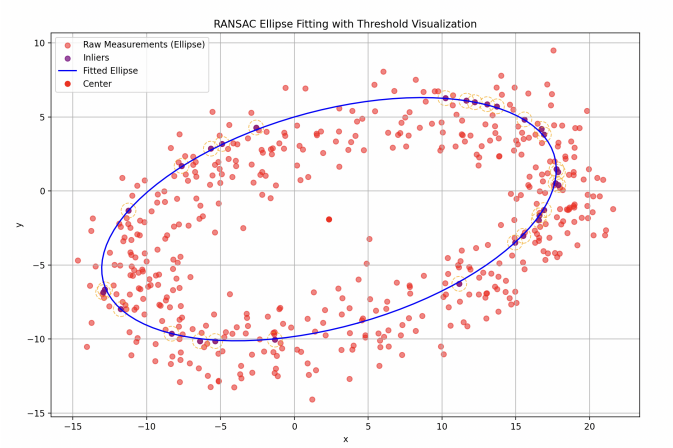

Figure 6: N = 500, Iterations = 1000, Threshold = 0.2

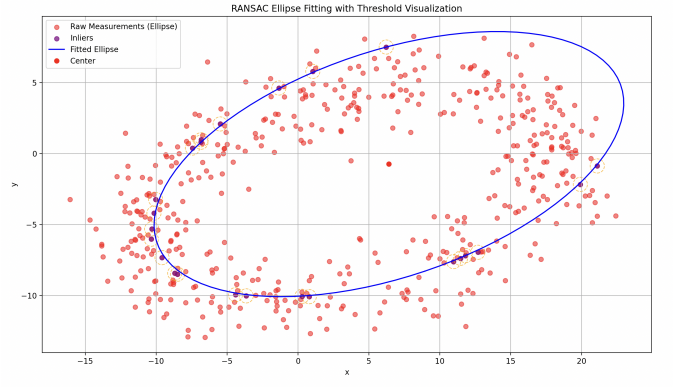

Figure 7: N = 500, Iterations = 100, Threshold = 0.2

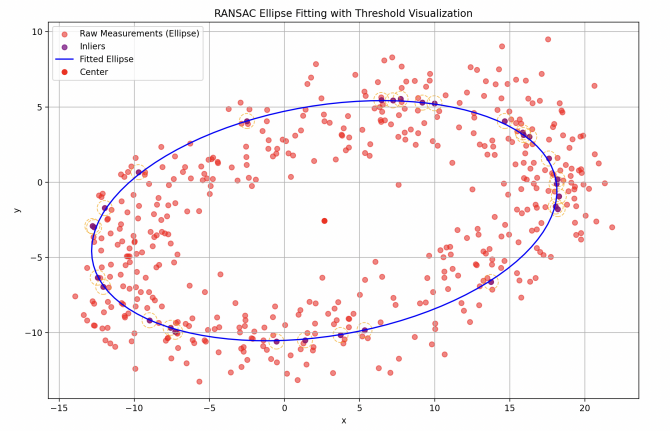

Figure 8: N = 500, Iterations = 3000, Threshold = 0.2

The results of modifying the number of iterations was somewhat expected, as the fitted ellipse had significant improvements with increasing iterations from 100 to 1000, but insignificant improvements with increasing iterations from 1000 to 3000.

Varying Threshold: Experimenting with thresholds of 0.2, 0.4, and 0.8:

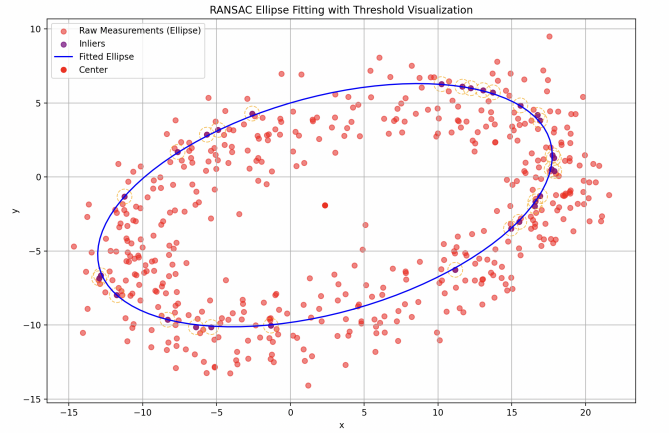

Figure 9: N = 500, Iterations = 1000, Threshold = 0.2

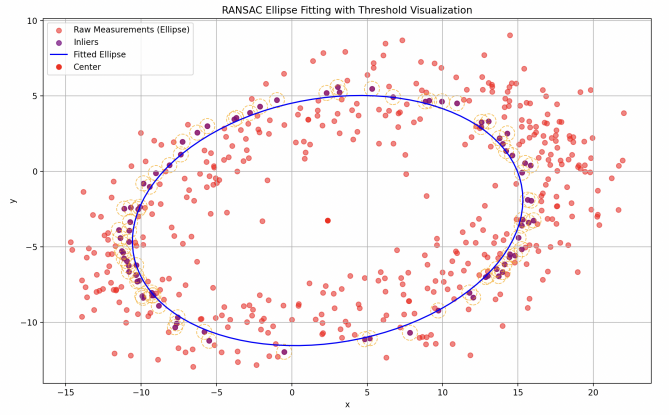

Figure 10: N = 500, Iterations = 1000, Threshold = 0.4

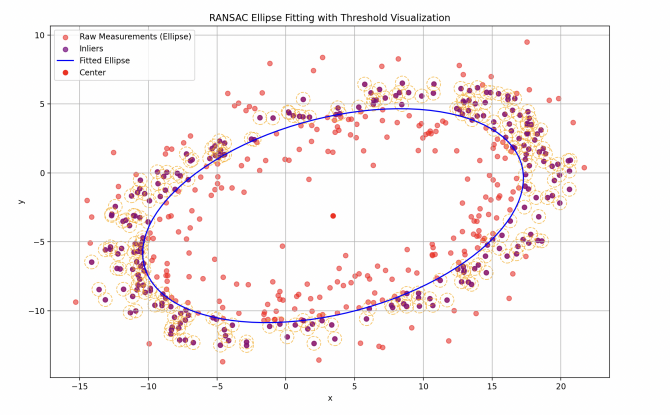

Figure 11: N = 500, Iterations = 2000, Threshold = 0.8

The results of modifying the threshold was as expected, as increasing the threshold is expected to increase the number of inliers which satisfy the condition.

3 Bayesian Filter for Tracking Moo-Deng's Behavior I (32)

Figure 12: An abstract version of Moo-Deng's enclosure

Khao Kheow Open Zoo in Thailand has gained popularity thanks to its star resident, Moo-Deng, a playful yet feisty Pygmy hippo. To better understand Moo-Deng's daily routine, the zoo has hired a wildlife researcher to track her movements between three key zones: Feeding Area ($A = 1$), Resting Area ($A = 2$),Playground Area ($A = 3$). Let $\mathbf{x}$ be a vector of probability of Moo-Deng being in the feeding area, the resting area, and the playground area respectively, which can be formally expressed as follows:

$$\mathbf{x} = \begin{bmatrix} p(A=1) \\ p(A=2) \\ p(A=3) \end{bmatrix}$$

It can be assumed that Moo-Deng only stays in one of these 3 places. Therefore:

$$\sum_{i=1}^{3} p(A=i) = \mathbf{1}^\top \mathbf{x} = 1$$

After careful observation, the researcher has identified Moo-Deng's behavior patterns:

- When Moo-Deng is in the Feeding Area ($A = 1$), she has a 60% chance of staying there, and a 40% chance of moving to the Resting Area ($A = 2$).
- If Moo-Deng is in the Resting Area ($A = 2$), she has a 40% chance of staying there, a 20% chance of returning to the Feeding Area ($A = 1$), and a 40% chance of moving to the Playground Area ($A = 3$).
- If Moo-Deng is in the Playground Area ($A = 3$), she has a 70% chance of staying there, a 20% chance of moving to the Feeding Area ($A = 1$), and a 10% chance of moving to the Resting Area ($A = 2$).

The behavior of Moo-Deng can be modelled as a transition matrix $\mathbf{A} \in \mathbb{R}^{3\times3}$ such that the following is true:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k$$

where $\mathbf{x}_k$ denotes the probability vector at the k^{th} observation.

Furthermore, the researcher uses a digital sensor system to track Moo-Deng's location, but the sensor is not perfectly accurate:

- When Moo-Deng is in the Feeding Area, the sensor reports Feeding Area with an 80% accuracy, but it incorrectly reports Resting Area 10% of the time and Playground Area 10% of the time.
- When Moo-Deng is in the Resting Area, the sensor reports Resting Area 70% of the time, but it incorrectly reports Feeding Area 20% of the time and Playground Area 10% of the time.
- When Moo-Deng is in the Playground Area, the sensor reports Playground Area with an 85% accuracy, but it incorrectly reports Resting Area 10% of the time and Feeding Area 5% of the time.

At the k^{th} observation, the reading from the sensor system can be encoded to a vector of digital values $\mathbf{y}_k$ so that:

$$\mathbf{y}_k = \begin{cases} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top & \text{if the measurement displays } \texttt{Feeding Area} \\ \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^\top & \text{if the measurement displays } \texttt{Resting Area} \\ \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top & \text{if the measurement displays } \texttt{Playground Area} \end{cases}$$

Then, the probaiblity of sensor displaying each area can be modelled as follows:

$$\begin{bmatrix} p(\mathbf{y}_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top) \\ p(\mathbf{y}_k = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^\top) \\ p(\mathbf{y}_k = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top) \end{bmatrix} = \mathbf{C}\mathbf{x}_k$$

where $\mathbf{C} \in \mathbb{R}^{3\times 3}$ is yet to be determined.

a) Assume that Moo-Deng starts in the Feeding Area ($A_0 = 1$). What is the probability that her subsequent movements will follow the sequence (2,3,3,1) in this exact order? Explain your calculation. [4 pts]

Answer:
To find the probability that Moo-Deng follows the sequence $(2, 3, 3, 1)$, starting in the Feeding Area, we multiply the conditional probabilities of each step as follows:

- Step 1: The conditional probability of moving to $A = 2$ given the current state is $A = 1$ is **40**%.
- Step 2: The conditional probability of moving to $A = 3$ given the current state is $A = 2$ is **40**%.
- Step 3: The probability of staying in $A = 3$ is **70**%.
- Step 4: The conditional probability of moving to $A = 1$ given the current state is $A = 3$ is **20**%.

The probability that Moo-Deng follows the sequence $(2, 3, 3, 1)$, starting in the Feeding Area is:

$$0.4 \times 0.4 \times 0.7 \times 0.2 = 0.0224 = \mathbf{2.24}\%$$

b) Determine the numerical value of the transition matrix $\mathbf{A}$. You must explain your rationale. [5 pts]

Answer:
The transition matrix $\mathbf{A}$ describes the probability of transitioning from one area to another and can be understood from the total probability equation:

$$p(A_{k+1} = i) = \sum_{j=1}^{3} P(A_{k+1} = i \mid A_k = j) \cdot p(A_k = j)$$

Given this equation for applying state transitions: $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k$, it is evident that the A matrix should be constructed like so:

$$\mathbf{A} = \begin{bmatrix} P(A_{k+1} = 1 \mid A_k = 1) & P(A_{k+1} = 1 \mid A_k = 2) & P(A_{k+1} = 1 \mid A_k = 3) \\ P(A_{k+1} = 2 \mid A_k = 1) & P(A_{k+1} = 2 \mid A_k = 2) & P(A_{k+1} = 2 \mid A_k = 3) \\ P(A_{k+1} = 3 \mid A_k = 1) & P(A_{k+1} = 3 \mid A_k = 2) & P(A_{k+1} = 3 \mid A_k = 3) \end{bmatrix}$$

When Moo-Deng is in the Feeding Area ($A_k = 1$), she has a 60% chance of staying there ($A_{k+1} = 1$), and a 40% chance of moving to the Resting Area ($A_{k+1} = 2$).

- $P(A_{k+1} = 1 \mid A_k = 1) = 0.6$.
- $P(A_{k+1} = 2 \mid A_k = 1) = 0.4$.
- $P(A_{k+1} = 3 \mid A_k = 1) = 0.0$.

If Moo-Deng is in the Resting Area ($A_k = 2$), she has a 40% chance of staying there ($A_{k+1} = 2$), a 20% chance of returning to the Feeding Area ($A_{k+1} = 1$), and a 40% chance of moving to the Playground Area ($A_{k+1} = 3$).

- $P(A_{k+1} = 1 \mid A_k = 2) = 0.2$.
- $P(A_{k+1} = 2 \mid A_k = 2) = 0.4$.
- $P(A_{k+1} = 3 \mid A_k = 2) = 0.4$.

If Moo-Deng is in the Playground Area ($A_k = 3$), she has a 70% chance of staying there ($A_{k+1} = 3$), a 20% chance of moving to the Feeding Area ($A_{k+1} = 1$), and a 10% chance of moving to the Resting Area ($A_{k+1} = 2$).

- $P(A_{k+1} = 1 \mid A_k = 3) = 0.2$.
- $P(A_{k+1} = 2 \mid A_k = 3) = 0.1$.
- $P(A_{k+1} = 3 \mid A_k = 3) = 0.7$.

Thus, we have:

$$\mathbf{A} = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.4 & 0.4 & 0.1 \\ 0.0 & 0.4 & 0.7 \end{bmatrix}$$

c) Analytically compute the stationary probabilities of Moo-Deng's location $\mathbf{x}^*$. These are the probabilities that, after many transitions, Moo-Deng will be found in each of the zones regardless of the initial location. You must show your work. [Hint: ?? = $\mathbf{A}\mathbf{x}^*$] [8 pts]

Answer:
The stationary vector $\mathbf{x}^* = \begin{bmatrix} x_1^* & x_2^* & x_3^* \end{bmatrix}^\top$ represents the long-term probabilities that Moo-Deng will be in each area, and it satisfies:

$$\mathbf{x}_{k+1}^* = \mathbf{A}\mathbf{x}_k^*$$

while being constrained by the following:

$$x_1^* + x_2^* + x_3^* = 1.$$

which makes intuitive sense since it means that at this point in the transition of the state, the next state will not be different than the previous state, suggesting that the probabilities are 'stationary'.

Expanding $\mathbf{x}^* = \mathbf{A}\mathbf{x}^*$ gives a system of linear equations:

$$\begin{aligned} 0.6x_1^* + 0.2x_2^* + 0.2x_3^* &= x_1^* \\ 0.4x_1^* + 0.4x_2^* + 0.1x_3^* &= x_2^* \\ 0.4x_2^* + 0.7x_3^* &= x_3^* \end{aligned}$$

Using the equations above along with the provided constraint, we get the following result for $\mathbf{x}^*$:

$$\mathbf{x}^* = \begin{bmatrix} \frac{1}{3} & \frac{2}{7} & \frac{8}{21} \end{bmatrix}^\top$$

d) Determine the numerical value of the matrix $\mathbf{C}$. You must explain your rationale. [5 pts]
Answer:

The matrix $\mathbf{C}$ models the sensor's measurement probabilities for each of the three areas (Feeding Area, Resting Area, and Playground Area), based on the current state. To determine $\mathbf{C}$, we need to use the sensor accuracy and error rates provided.
Sensor Accuracy and Error Rates:

- When Moo-Deng is in the Feeding Area, the sensor reports Feeding Area with an 80% accuracy, but it incorrectly reports Resting Area 10% of the time and Playground Area 10% of the time.
  - $P(\mathbf{y}_k = [1, 0, 0]^\top \mid A = 1) = 80\%$
  - $P(\mathbf{y}_k = [0, 1, 0]^\top \mid A = 1) = 10\%$
  - $P(\mathbf{y}_k = [0, 0, 1]^\top \mid A = 1) = 10\%$
- When Moo-Deng is in the Resting Area, the sensor reports Resting Area 70% of the time, but it incorrectly reports Feeding Area 20% of the time and Playground Area 10% of the time.
  - $P(\mathbf{y}_k = [1, 0, 0]^\top \mid A = 2) = 20\%$

  - $P(\mathbf{y}_k = [0,1,0]^\top \mid A = 2) = 70\%$
  - $P(\mathbf{y}_k = [0,0,1]^\top \mid A = 2) = 10\%$
- When Moo-Deng is in the Playground Area, the sensor reports Playground Area with an 85% accuracy, but it incorrectly reports Resting Area 10% of the time and Feeding Area 5% of the time.
  - $P(\mathbf{y}_k = [1,0,0]^\top \mid A = 3) = 5\%$
  - $P(\mathbf{y}_k = [0,1,0]^\top \mid A = 3) = 10\%$
  - $P(\mathbf{y}_k = [0,0,1]^\top \mid A = 3) = 85\%$

Matrix C:

Matrix $\mathbf{C}$ can be constructed by leveraging the total probability principle, similar to the construction of the A matrix:

$$\mathbf{C} = \begin{bmatrix} P(\mathbf{y}_k = [1,0,0]^\top \mid A = 1) & P(\mathbf{y}_k = [1,0,0]^\top \mid A = 2) & P(\mathbf{y}_k = [1,0,0]^\top \mid A = 3) \\ P(\mathbf{y}_k = [0,1,0]^\top \mid A = 1) & P(\mathbf{y}_k = [0,1,0]^\top \mid A = 2) & P(\mathbf{y}_k = [0,1,0]^\top \mid A = 3) \\ P(\mathbf{y}_k = [0,0,1]^\top \mid A = 1) & P(\mathbf{y}_k = [0,0,1]^\top \mid A = 2) & P(\mathbf{y}_k = [0,0,1]^\top \mid A = 3) \end{bmatrix}$$

Substituting the values from above:

$$\mathbf{C} = \begin{bmatrix} 0.8 & 0.2 & 0.05 \\ 0.1 & 0.7 & 0.1 \\ 0.1 & 0.1 & 0.85 \end{bmatrix}$$

e) Derive a state estimator based on a Bayesian Filter that estimates the states and probability of Moo-Deng's based on prior belief $\mathbf{x}_{k-1}$ and the current sensor measurement $\mathbf{y}_k$. You may leave your answer in terms of $\mathbf{A}$ and $\mathbf{C}$. You may also substitute their numeric values. [10 pts]
Answer:
Bayesian Filter for State Estimation (Prediction Step):

$$\overline{bel}(x_k) = \sum p(x_k|x_{k-1})\, bel(x_{k-1}) \tag{1}$$

Bayesian Filter for State Estimation (Correction Step):

$$bel(x_k) = \eta p(y_k|x_k)\, \overline{bel}(x_k) \tag{2}$$

In the context of this problem, $bel(\boldsymbol{x}_k)$ will be a 3×1 vector with 3 potential values (the same potential values as y_k: $\begin{bmatrix}1 & 0 & 0\end{bmatrix}^T$, $\begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$, $\begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$), which represents the best guess of Moo-Deng's current state/location.

Conditional probability of the current state given the previous state, $p(x_k|x_{k-1})$, is captured by the $\boldsymbol{A}$ matrix. Thus, the $\boldsymbol{A}$ matrix can be substituted into equation (1) in place of the $\sum p(x_k|x_{k-1})$

$$\overline{bel}(\boldsymbol{x}_k) = \boldsymbol{A} \cdot bel(\boldsymbol{x}_{k-1}) \tag{3}$$

Conditional probability of the current measurements given the current state, $p(y_k|x_k)$, is captured by the $\boldsymbol{C}$ matrix. Thus, the $\boldsymbol{C}$ matrix can be substituted into equation (2) in place of the $p(y_k|x_k)$

$$bel(\boldsymbol{x}_k) = \eta \boldsymbol{C} \cdot \overline{bel}(\boldsymbol{x}_k) \tag{4}$$

Combining equations (3) and (4) and dropping the normalization factor η for clarity:

$$bel(\boldsymbol{x}_k) = \boldsymbol{C}\boldsymbol{A} \cdot bel(\boldsymbol{x}_{k-1}) \tag{5}$$

4 Bayesian Filter for Tracking Moo-Deng's Behavior II (20)

In this exercise, you are tasked to implement a state estimator for Moo-Deng's whereabouts along with a simple behavior simulator in Python . The implementation must match your explanation in the report in order to receive full credits.

a) Based on the given code in Python, complete the function `moodeng_behavior_update` that randomly returns the state of Moo-Deng based on the given current state and the transition matrix **A**. You must explain your implementation in this report. Comment in the report on how can you use statistics to verify your results ? [5 pts.]

Answer: To verify the results of the completed function, the number of iterations can be increased to infinity (in practice, just a large number), after which the frequency of each state can be compared to x^* from Question 3c.

b) Based on the given code in Python, complete the function `sensor_measurement` that return the noisy measurement $\mathbf{y}_k$ based on the matrix **C** and given Moo-Deng's state. Note that the results should be either (1,0,0), (0,1,0), or (0,0,1). [5 pts.]

Answer: See code provided

c) Based on the designed Bayesian filter and the given code in Python, implement a state estimator and the rest of the simulation in called `sim_moodeng`. The estimator should :

- estimate a sequence state of Moo-Deng's location $\hat{A}_k$
- return a sequence belief (probability vector) of Moo-Deng's location $\mathbf{x}_k$

given an initial belief $\mathbf{x}_0$ and a sequence of sensor measurement $\mathbf{y}_k$ [6 pts.].

Answer: See code provided