EX5 Mobile Robotics

Abdulrahman Hameshli

December 2024

05.01: Explanation of the Forward-Backward Algorithm Code

Purpose of the Code

The code implements the **Forward-Backward Algorithm** for a Hidden Markov Model (HMM) to compute the following:

- **Forward Probabilities (α_k):** The probability of being in each state s at time k, given the observations up to time k.
- **Backward Probabilities (β_k):** The probability of observing the remaining sequence from time $k+1$ onwards, given the current state at time k.
- **Smoothed Probabilities (γ_k):** Combined forward and backward probabilities, providing the most accurate estimate of the hidden states at each time step.
- **True State Comparison:** The true state sequence generated during simulation is plotted alongside the smoothed probabilities to compare the model's estimates with the ground truth.

Why We Do It

- **Infer Hidden States:** To estimate hidden states in a probabilistic model when observations are noisy or incomplete.
- **Smooth Estimates:** The smoothed probabilities (γ_k) incorporate both past and future observations, improving accuracy compared to forward probabilities alone.
- **Validate Model Performance:** Plotting the true state sequence alongside the smoothed probabilities helps assess how well the model captures the underlying dynamics.

Necessary Steps

1. **Forward Pass:** Compute α_k recursively from the initial priors, using the transition matrix (A) and observation matrix (E).
2. **Backward Pass:** Compute β_k recursively starting from the end of the sequence, using A and E.
3. **Smoothing:** Combine α_k and β_k to compute γ_k:

$$\gamma_k(s) = \frac{\alpha_k(s) \cdot \beta_k(s)}{\sum_{s'} \alpha_k(s') \cdot \beta_k(s')}.$$

4. **Plot Results:** Plot γ_k for each state over time and overlay the true state sequence for comparison.

Inference Comparison for 05.01

- **Forward Probabilities (α_k):** These probabilities estimate the hidden state based only on observations up to the current time step k. They are effective for real-time filtering but do not account for future observations.

- **Backward Probabilities (β_k):** These probabilities estimate the hidden state based only on future observations. They are useful for post-hoc smoothing but lack information about the past.
- **Smoothed Probabilities (γ_k):** By combining forward and backward probabilities, γ_k provides the most accurate estimate of the hidden states by leveraging all observations (past and future).
- **Comparison Summary:**
  - α_k is ideal for filtering in real-time applications.
  - β_k is useful for backward inference in batch processing.
  - γ_k is the gold standard for inference when the entire observation sequence is available.

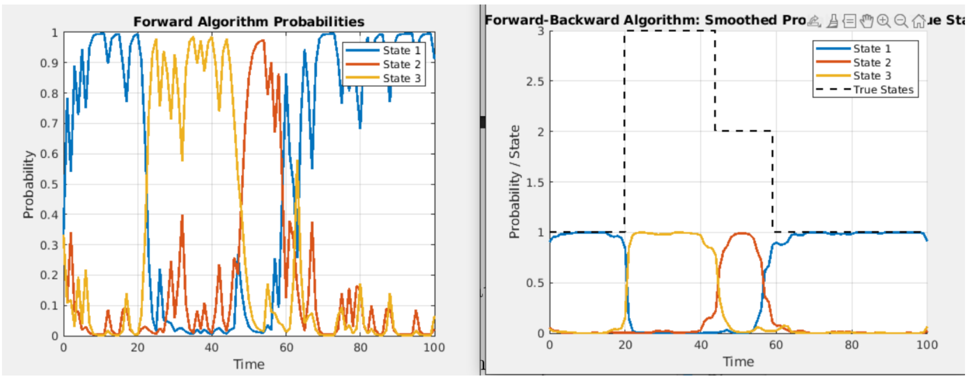

Figure 1: Comparison of forward, backward, and smoothed probabilities with true states.

Random seeds gives us the same sequence of random numbers generated by Matlab.

05.02: Explanation of the Confusion Matrix

What is a Confusion Matrix?

The confusion matrix is a table used to evaluate the performance of a classification model. In this exercise, it compares the **true state sequence** to the **predicted state sequence (path)** recovered using the Viterbi algorithm.

Structure of the Confusion Matrix

- Rows correspond to **true states**.
- Columns correspond to **predicted states**.
- Each entry (i, j) represents the number of times the true state i was predicted as state j.

Normalized Confusion Matrix

To make the confusion matrix interpretable, each row is normalized to sum to 1. This converts counts into probabilities, allowing us to interpret the likelihood of a true state being predicted as another state.

Interpretation

- **Diagonal Elements:** High values indicate accurate predictions (true states match predicted states).
- **Off-Diagonal Elements:** High values indicate misclassifications, where the algorithm confused one state with another.
- **Example:** A strong diagonal with minimal off-diagonal values indicates high path reconstruction accuracy.

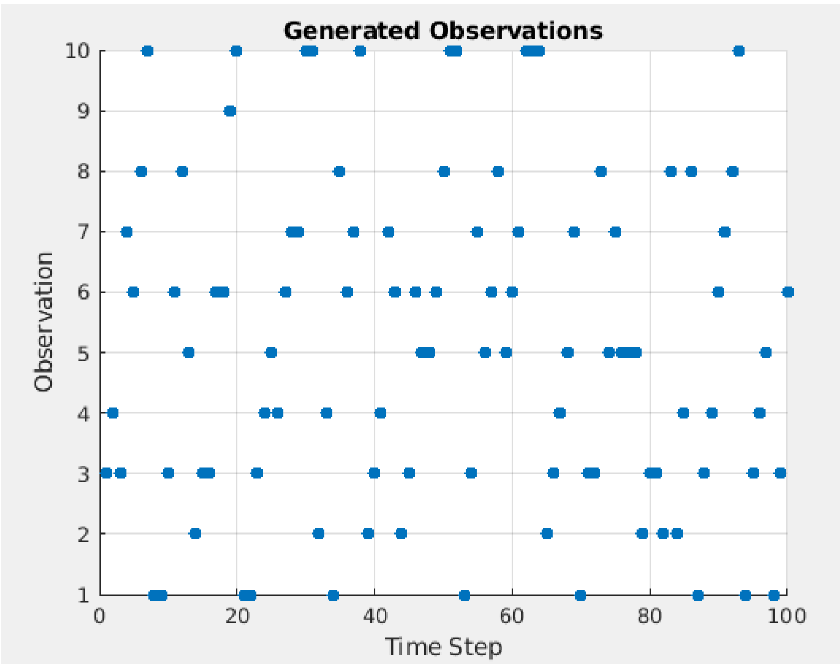

Figure 2: Confusion Matrix for Path Reconstruction.

This seems to be about 80 percent accurate, which is not good for a moving robot among people.

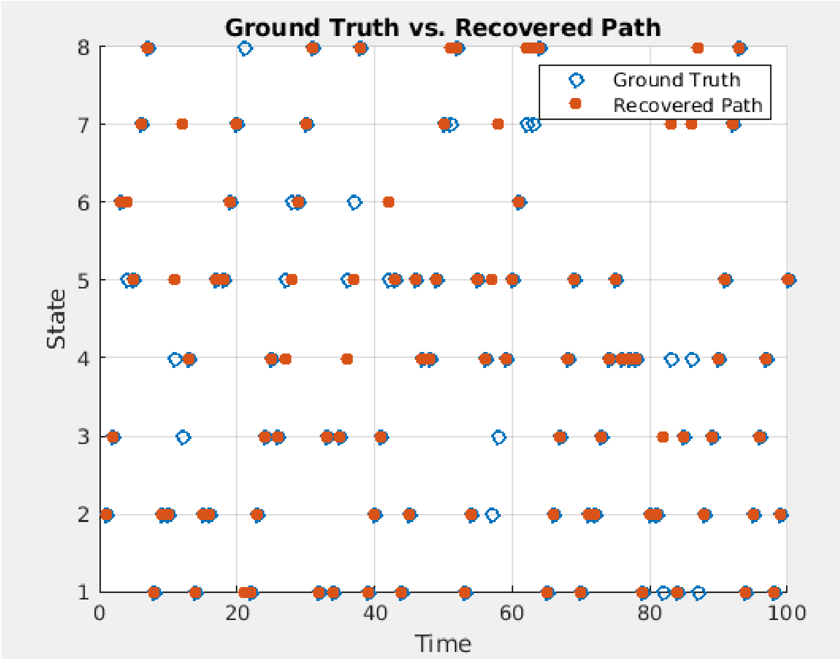

Figure 3: Confusion Matrix for Path Reconstruction.

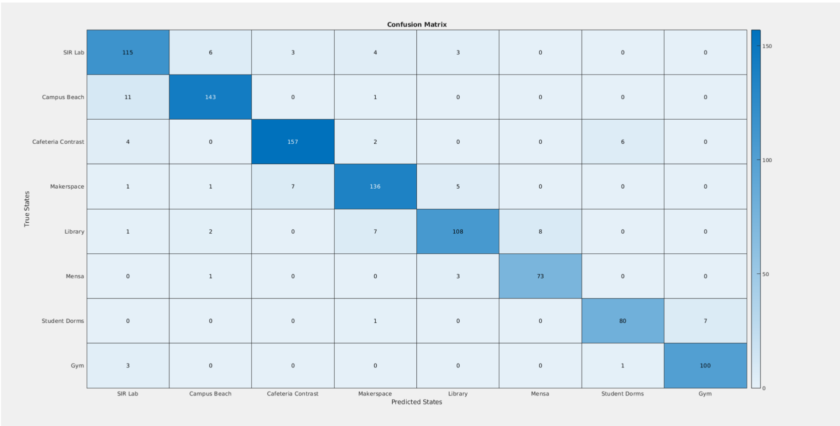

Figure 4: Confusion Matrix for Path Reconstruction.

Exercise 05.03: Modified Transition and Emission Matrices

Scenario a: A "Cheap Delivery Robot with Bad Sensing"

Characteristics: The robot has poor sensing capabilities:

- It struggles to identify landmarks accurately.
- The emission probabilities (E) become more uniform across landmarks.

Modified Emission Matrix (E):

$$E = \begin{bmatrix} 0.3 & 0.3 & 0.05 & 0.05 & 0 & 0.2 & 0.05 & 0.05 & 0 & 0 \\ 0.1 & 0.05 & 0.6 & 0.05 & 0 & 0.1 & 0.05 & 0.05 & 0 & 0 \\ 0.05 & 0.1 & 0.05 & 0.5 & 0 & 0.05 & 0.05 & 0.2 & 0.05 & 0 \\ 0.05 & 0.05 & 0.05 & 0.05 & 0.5 & 0.15 & 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.3 & 0.3 & 0.1 & 0.05 & 0 \\ 0.05 & 0.05 & 0.1 & 0.05 & 0 & 0.05 & 0.6 & 0.05 & 0.05 & 0 \\ 0.05 & 0.05 & 0.05 & 0.05 & 0 & 0.05 & 0.05 & 0.5 & 0.15 & 0.05 \\ 0.1 & 0.05 & 0.05 & 0.05 & 0 & 0.05 & 0.05 & 0.05 & 0.1 & 0.5 \end{bmatrix}$$

Impact:

- Accuracy decreases because the robot is more likely to confuse landmarks.

Scenario b: A "Delivery Robot with a Broken Leg"

Characteristics: The robot has limited mobility:

- Transition probabilities (A) favor staying in the same state or moving to nearby states.

Modified Transition Matrix (A):

$$A = \begin{bmatrix} 0.7 & 0.2 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0.5 & 0.2 & 0.1 & 0 & 0 & 0 & 0 \\ 0.1 & 0.2 & 0.5 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0.2 & 0.6 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0.7 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0.6 & 0.2 & 0.1 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0.7 & 0.1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0.9 \end{bmatrix}$$

Impact:

- The reconstructed path is more likely to reconstructed as good as not a broken leg because we can have good observation and recognize where we are.