Mobile Robotics Winter term 2024/2025

Lecturer: Prof. Kai Arras
Lab instructors: Till Hielscher, Dennis Rotondi, Fabio Scaparro

Socially Intelligent Robotics Lab
Institute for Artificial Intelligence
Faculty of Computer Science
University of Stuttgart

Submission and voting deadline: see ILIAS

Submission Instructions: For the solution of theoretical tasks, use a header with your name and Matrikelnummer on each sheet and combine all files (pictures, scans, LaTeX-ed solutions) into a single `.pdf` document. For code solutions, if not stated otherwise, your code should execute correctly when called from a single `.m` or `.mlx` script (external functions are ok as long as they are called from the script). Each file you submit must include a header with your name and Matrikelnummer. Please add comments to make your code readable and to indicate to which task and subtask it refers to. For submission, all files should be included in a single `.zip` archive named as: `Ex05_YourLastname_Matrikelnummer.zip`. Remember to vote on the tasks that you solved and be ready to present them.

Exercise 05: Temporal Inference: HMM

Exercise 05.1: Forward-Backward Algorithm

In this exercise you will implement the forward-backward algorithm, and compare the inference results for filtering and smoothing. The algorithm has a compact matrix implementation which makes it particularly simple to implement in Matlab/Octave.

Let us remember our HMM notation: hidden variables $\mathbf{x}$ have $s \in \{1, ..., S\}$ states, the transition and observation models are described by matrices A and E of dimension $S \times S$ and $S \times O$, respectively, where O is the number of observation symbols (here we will assume $S = O = 3$).

The algorithm uses the probability $p(\mathbf{z}_k \mid \mathbf{x}_k = s)$ which specifies how likely it is that state s causes $\mathbf{z}_k$ to appear. Following our definition of matrix E this corresponds to a column of E. For mathematical convenience we place these values into a $S \times S$ diagonal matrix D. If, for instance, $p(\mathbf{z}_k \mid \mathbf{x}_k = 1) = (0.9 \;\; 0.2)^T$ (from the umbrella example), then

$$D_k = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}$$

(**Hint:** to do this use `diag`). Now, using column vectors to represent the forward and backward probabilities α_k and β_k, all the computations become simple matrix-vector operations. The forward step becomes

$$\alpha_{k+1} = \eta \cdot D_k \cdot A^T \cdot \alpha_k \tag{1}$$

and the backward step becomes

$$\beta_k = A \cdot D_k \cdot \beta_{k+1} \tag{2}$$

The initial conditions are $\alpha_0 = p(\mathbf{x}_0)$, where $p(\mathbf{x}_0)$ is a $S \times 1$ prior distribution, and $\beta_K = \mathbf{1}$, a $S \times 1$ vector of 1's, where K is the length of the sequence. HMM parameters and additional explanations are given in the file `RunHMM.m`. Proceed as follows:

a) Implement the forward algorithm and plot the probabilities for each state s over time. Normalize the α's for each step in order them to be proper probability distributions.

b) Implement the forward-backward algorithm and plot the probabilities for each state s over time.

c) Compare the results visually, try different random seeds and rerun the algorithms. Familiarize yourself with the algorithms' behavior.

Use `nstates` $\times$ `nobservations+1` matrices for α, β and the smoothed probabilities (the `+1` is for the prior α_0). Iterate in both algorithms over the index range `2` to `nobservations+1` corresponding to time steps $k = 1, ..., K$.

Exercise 05.2: Localization (and path recovery) in a topological map

A delivery robot is tasked with navigating the University of Stuttgart Vaihingen campus to deliver food to hardworking students. The robot **begins** its journey at the SIR Lab and must efficiently identify its location based on various landmarks scattered throughout the campus. The following locations (states) are close to each other and are the possible locations where the robot could be during its journey:

1. SIR Lab
2. Campus Beach
3. Cafeteria Contrast
4. Makerspace
5. Library
6. Mensa
7. Student Dorms
8. Gym

The robot can observe the following landmarks, each associated with specific locations where they are most likely to be found:

1. Go2 Robot - Likely located at SIR Lab
2. Pepper Robot - Likely located at SIR Lab
3. Beach Chairs - Likely located at Campus Beach
4. Espresso Machine - Likely located at Cafeteria Contrast
5. 3D Printers - Likely located at Makerspace
6. Study Tables - Likely located at Library
7. Bookshelves - Likely located at Library
8. Buffet Station - Likely located at Mensa
9. Student Bikes - Likely located at Student Dorms
10. Workout Equipment - Likely located at Gym

The robot will generate a sequence of noisy observations based on its encounters with the landmarks as it navigates through the environment. Each observation corresponds to a landmark it detects. Assuming we model the following problem using an HMM and the transition and observation models described by matrices A and E in the file `RunHMM.m`, the task is to determine the most likely path (sequence of states) that explains the observations made by the robot.

Plot the ground truth state sequence, the noisy landmark observations and the recovered most likely path. As a measure of accuracy, plot the confusion matrix of your estimated states of the path using the Matlab function `heatmap`.

Exercise 05.3: Modify the transition and emission matrices

Considering the previous exercise, *Localization (and path recovery) in a topological map*, how would you modify the transition matrix A and the emission matrix E presented in the code for the following scenarios:

a) A "cheap delivery robot with bad sensing."

b) A "delivery robot with a broken leg."

Write down the full expressions for the matrices and comment on the most important values in these matrices. How does the accuracy of the reconstructed path and the confusion matrix change as a function of your modifications?