To implement the Hidden Markov Model and find the coordinates of the most likely trajectory of the robot for 11 time-steps.

PROJECT DESCRIPTION

- Programming language used: Python
- Data Structure: Lists, Arrays
- File Name : HMM.py
- Inputs : hmm-data.txt
- Output : the coordinates of the most likely trajectory of the robot for 11 time stamps

IMPLEMENTATION

- <u>Modules created:</u>
  - Read_File() - reads the input file line by line, and creates a grid and noisy distance matrix based on the inputs given in the text file.

  - Distance(t1, t2, Rx, Ry) - Given t1,t2 as the coordinates of the tower location on the grid, and Rx, Ry as the coordinates of the current position of the robot in the grid, this function returns the absolute Euclidean distance between the point and the tower.

  - HMM(grid, nd) – takes the grid and noisy distances as input, and computes the prior matrix, transition matrix X, four Evidence Matrices E1, E2, E3, E4 for storing the probabilities of the distance between the respective tower and the point, considering the noisy interval of [0.7d, 1.3d]. Based on the computed distances in E, their probabilities, and the given noisy distances -nd, robot's location is estimated. The outputs from all the four towers are combined for a single timestamp and a rough list of all probable points where the robot can be, in the 11 timestamps is returned from the module.

  - FindLoc(E, er) – Based on the Evidence Matrix E for a tower, and the row header er where the distances are same as the given noisy distance, this function returns the list of probable locations where the distance given in the noisy matrix have some non-zero probabilities.

  - Evidence(Z, D1, t1, t2, n) – Computes and returns the Evidence matrices for given t1,t2 as the tower location, n = number of free cells, Z = matrix of free cell locations, D1 =matrix of distances from 0.0 to $9\sqrt{2}$ *1.3.

  - findNeighbours(location,X) – Based on transition matrix X, if there are

traversable neighbors for a location, list of neighbors is returned.

- Path(p,X,location_list,i,Z) – Recursive function call, that checks in the list of most probable points returned from HMM(), all the possible robot trajectories. It was observed there were 3 trajectories that were longest. The trajectory that covers the path for 11 timestamps and has the minimum x,y starting coordinates is reported as the final output.

- Coord(fin_path,Z)- prints the final output in coordinate format.

- Termination Condition:

  When the trajectory covering all 11 timestamps is found and also begins with minimum x and y coordinates, no more neighbors are evaluated to find the route.

- Result Interpretation:

  HMM: The result represents the coordinates of the trajectory it follows, when it moves through the grid for each of the 11 timestamps. Though there were other possible trajectories too starting at (5,4) and ending at (3,0), but the one chosen for the final output has the smaller x and y coordinates.