Hidden Markov Model

Programming Assignment 7

Patrick Humphries (7097-1087-72, pvhumhr@usc.edu)

INF552 Machine Learning for Data Science (32458)

Viterbi School of Engineering

University of Southern California

Spring 2020

Abstract We live in uncertain world. In this exercise, the next step of a robot is to be predicted based on its previous position, state, emissions (distances) and transition template (obstacles). However, its calculated position is uncertain. These factors used in conjunction with a recursive process allowed for the estimating the probable path of the robot. Another part of the exercise was to become familiar with an HMM library. *Pomegranate* was used to find a region of free cells. Such a region could be used as a bias to direct the path of the robot.

Contents

Locations

In the recursive process, the previous location is input and next location is output. The collection of the next locations is the objective of this exercise: "You should output the coordinates of the most likely trajectory of the robot for 11 time-steps."

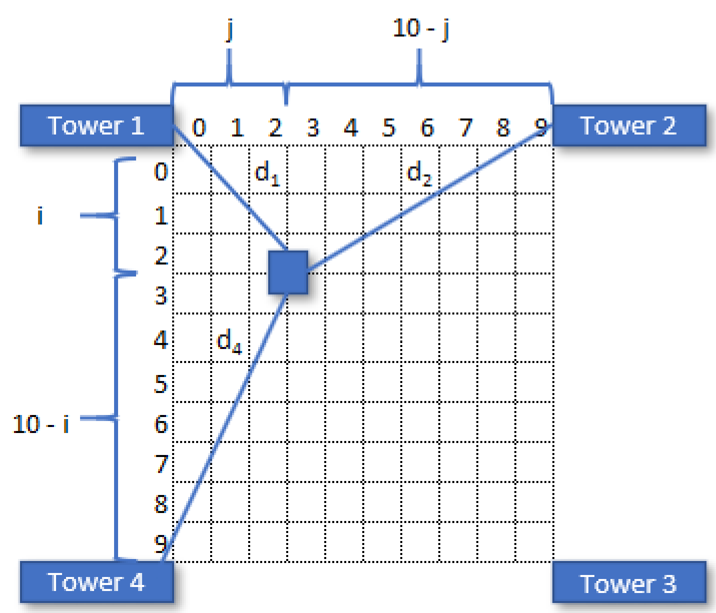
Emissions (Calculate Location)

The robot can only report the noisy distance to each of the four transmission towers.

Using the distances from the towers, an estimate of its location can be made.

In this exercise, the variables i and j are calculated from the distances d_1, d_2, and d_4.

The key is the equation for Tower 1. It can be used for replacing either i^2 or j^2 in either of the other two equations



Tower 1:

$$d_1^2 = i^2 + j^2 \Longrightarrow j^2 = d_1^2 - i^2 \Longrightarrow i^2 = d_1^2 - j^2$$

Tower 4:

$$d_4^2 = (10 - i)^2 + j^2$$

Apply Tower 1 to Tower 4 for j^2:

$$d_4^2 = (10 - i)^2 + d_1^2 - i^2$$

$$d_4^2 - d_1^2 = 100 - 20i + i^2 - i^2$$

$$d_4^2 - d_1^2 - 100 = -20i$$

$$-d_4^2 + d_1^2 + 100 = 20i$$

$$(100 + d_1^2 - d_4^2)/_{20} = i$$

Tower 2:

$$d_2^2 = i^2 + (10 - j)^2$$

Apply Tower 1 to Tower 2 for i^2:

$$d_2^2 = d_1^2 - j^2 + (10 - j)^2$$

$$d_2^2 - d_1^2 = -j^2 + 100 - 20j + j^2$$

$$d_2^2 - d_1^2 - 100 = -20j$$

$$100 + d_1^2 - d_2^2 = 20j$$

$$(100 + d_1^2 - d_2^2)/_{20} = j$$

Apply i and j to Tower 3 to verify:

$$d_3^2 = (10 - i)^2 + (10 - j)^2$$

State

By combining the estimated location of the robot and the topology of the grid, the state of the robot's location can be estimated. The *state* is defined as the cells directly surrounding the robot.

Transition Template

Since a free cell has a value of 1 and 0 for the value of an obstructed cell, the validity of a transition can be easily determined by the product of the two cell values. This is shown in the Conditional Probability Table (CPT) .

		Previous Cell	
		0	1
Current Cell	0	0	0
	1	0	1

The transitions (products) in all eight directions can be summed. To get the probability, they were divided by 100. The resulting Transition Template is a vector of eight members, each representing a direction. The likely transitions are North and South. This makes sense because of the unobstructed columns. The unobstructed rows yields the next highest probability.

```
Transition Template
N:      0.72
NE:     0.56
E:      0.68
SE:     0.56
S:      0.72
SW:     0.56
W:      0.68
NW:     0.56
```

However, since the locations needed to be contiguous, the calculated transition template was not useful. What served as the transition template where the locations of the obstacles.

Viterbi Algorithm

The Viterbi Algorithm is a recursive process. The previous location combined with the calculated location yields the current location. This is tempered by the state (surrounding cells) and the transition template (obstacles).

$$previous\ location * emissions\ (calculated\ location) * state \\ * transition\ template\ (obstacles) = next\ location$$

Data

Before starting any analytical process, the data needs to be examined. There maybe visual clues about structure.

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	1	1	1
2	1	1	0	1	1	1	0	1	1	1
3	1	1	0	1	1	1	0	1	1	1
4	1	1	0	1	1	1	0	1	1	1
5	1	1	0	1	1	1	0	1	1	1
6	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

Observations (Steps)

```
Steps (Observations)
Step: 1         Tower 1: 6.3    Tower 2: 5.9    Tower 3: 5.5    Tower 4: 6.7
Step: 2         Tower 1: 5.7    Tower 2: 7.2    Tower 3: 4.4    Tower 4: 6.8
Step: 3         Tower 1: 7.6    Tower 2: 9.4    Tower 3: 4.3    Tower 4: 5.4
Step: 4         Tower 1: 9.5    Tower 2: 10.0   Tower 3: 3.7    Tower 4: 6.6
Step: 5         Tower 1: 6.0    Tower 2: 10.7   Tower 3: 2.8    Tower 4: 5.8
Step: 6         Tower 1: 9.3    Tower 2: 10.2   Tower 3: 2.6    Tower 4: 5.4
Step: 7         Tower 1: 8.0    Tower 2: 13.1   Tower 3: 1.9    Tower 4: 9.4
Step: 8         Tower 1: 6.4    Tower 2: 8.2    Tower 3: 3.9    Tower 4: 8.8
Step: 9         Tower 1: 5.0    Tower 2: 10.3   Tower 3: 3.6    Tower 4: 7.2
Step: 10        Tower 1: 3.8    Tower 2: 9.8    Tower 3: 4.4    Tower 4: 8.8
Step: 11        Tower 1: 3.3    Tower 2: 7.6    Tower 3: 4.3    Tower 4: 8.5
```

There actually 12 observations. The first one is implied and it affected the location of the first location. The assignment specified that "initial position of the robot has a uniform prior over all free cells." It is understood that the state of the first location will be all free cells.

Execution

There are three guiding principals in the execution:

1. The calculated location merely indicates the direction of the next location.
2. The calculated location is adjusted to become contiguous with the previous location.
3. Avoid crossing borders and encountering obstacles.

Observation 1: Row 4, Column 4

The calculated location was (4,5). However, the requirements specified: "Assume that the initial position of the robot has uniform prior over all free cells. Since the calculated location was up against an obstacle, it was moved West.

```
Step 1
Getting robot location based on emissions (distances).
Calculated i: 4 j: 5
Next to obstruction, moved West.
Plot Cells
```

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	1	1	1
2	1	1	0	1	1	1	0	1	1	1
3	1	1	0	1	1	1	0	1	1	1
4	1	1	0	1	step 1	1	0	1	1	1
5	1	1	0	1	1	1	0	1	1	1
6	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

Observation 2: Row 4, Column3

Just happened that the calculated to be in the same location as the prior location. It was moved to the West. This was a randomly selected direction.

```
Step 2
Getting robot location based on emissions (distances).
Calculated i: 4 j: 4
Calculated location on top of previous location.
Move West.
Plot Cells
```

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	1	1	1
2	1	1	0	1	1	1	0	1	1	1
3	1	1	0	1	1	1	0	1	1	1
4	1	1	0	step 2	step 1	1	0	1	1	1
5	1	1	0	1	1	1	0	1	1	1
6	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

Observation 3: Row 5, Column 3

The robot is moving South. The location required a single adjustment.

```
Step 3
Getting robot location based on emissions (distances).
Calculated i: 6 j: 3
South of previous.
Moved North.
Plot Cells
```

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	1	1	1
2	1	1	0	1	1	1	0	1	1	1
3	1	1	0	1	1	1	0	1	1	1
4	1	1	0	step 2	step 1	1	0	1	1	1
5	1	1	0	step 3	1	1	0	1	1	1
6	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

Observation 4: Row 6, Column 3

The robot is still moving South.

```
Step 4
Getting robot location based on emissions (distances).
Calculated i: 7 j: 4
South of previous.
Moved North.
East of previous.
Moved West.
Plot Cells
```

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	1	1	1
2	1	1	0	1	1	1	0	1	1	1
3	1	1	0	1	1	1	0	1	1	1
4	1	1	0	step 2	step 1	1	0	1	1	1
5	1	1	0	step 3	1	1	0	1	1	1
6	1	1	1	step 4	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

Observation 5: Row 6, Column 2

The robot has turned the corner, heading West.

```
Step 5
Getting robot location based on emissions (distances).
Calculated i: 5 j: 1
North of previous.
Moved South.
West of previous.
Moved East.
Plot Cells
```

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	1	1	1
2	1	1	0	1	1	1	0	1	1	1
3	1	1	0	1	1	1	0	1	1	1
4	1	1	0	step 2	step 1	1	0	1	1	1
5	1	1	0	step 3	1	1	0	1	1	1
6	1	1	step 5	step 4	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

Observation 6: Row 6, Column 3

The robot has backtracked! There is no reason of it not being allowed to do so.

```
Step 6
Getting robot location based on emissions (distances).
Calculated i: 7 j: 4
South of previous.
Moved North.
East of previous.
Moved West.
Plot Cells
```

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	1	1	1
2	1	1	0	1	1	1	0	1	1	1
3	1	1	0	1	1	1	0	1	1	1
4	1	1	0	step 2	step 1	1	0	1	1	1
5	1	1	0	step 3	1	1	0	1	1	1
6	1	1	step 5	step 6	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

Observation 7: Row 6, Column 2

The robot is making a second attempt to round the corner. The calculated location of (3,0) is a definite indication that it wants to turn the corner.

```
Step 7
Getting robot location based on emissions (distances).
Calculated i: 3 j: 0
North of previous.
Moved South.
West of previous.
Moved East.
North of previous.
Moved South.
West of previous.
Blocked, moved South.
West of previous.
Moved East.
Plot Cells
```

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	1	1	1
2	1	1	0	1	1	1	0	1	1	1
3	1	1	0	1	1	1	0	1	1	1
4	1	1	0	step 2	step 1	1	0	1	1	1
5	1	1	0	step 3	1	1	0	1	1	1
6	1	1	step 7	step 6	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

Observation 8: Row 6, Column 1

The robot has made the corner. There was a manual intervention. Low-end location calculations where used instead. This is valid because they were within the requirements. However, it was not automated.

```
Step 8
Getting robot location based on emissions (distances).
Calculated i: 1 j: 2
North of previous.
Blocked, moved West.
North of previous.
Moved South.
West of previous.
Blocked, moved South.
North of previous.
Moved South.
West of previous.
Blocked, moved South.
North of previous.
Moved South.
Plot Cells
```

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	1	1	1
2	1	1	0	1	1	1	0	1	1	1
3	1	1	0	1	1	1	0	1	1	1
4	1	1	0	step 2	step 1	1	0	1	1	1
5	1	1	0	step 3	1	1	0	1	1	1
6	1	step 8	step 7	step 6	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

Observation 9: Row 5, Column 1

Heading home. Again, a calculated location of (3,0) is a clear indicator of the probable direction.

```
Step 9
Getting robot location based on emissions (distances).
Calculated i: 3 j: 0
North of previous.
Moved South.
West of previous.
Moved East.
North of previous.
Moved South.
Plot Cells
```

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	1	1	1
2	1	1	0	1	1	1	0	1	1	1
3	1	1	0	1	1	1	0	1	1	1
4	1	1	0	step 2	step 1	1	0	1	1	1
5	1	step 9	0	step 3	1	1	0	1	1	1
6	1	step 8	step 7	step 6	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

Observation 10: Row 4, Column 1

Still heading North.

```
Step 10
Getting robot location based on emissions (distances).
Calculated i: 1 j: 0
North of previous.
Moved South.
West of previous.
Moved East.
North of previous.
Moved South.
North of previous.
Moved South.
Plot Cells
```

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	1	1	1
2	1	1	0	1	1	1	0	1	1	1
3	1	1	0	1	1	1	0	1	1	1
4	1	step 10	0	step 2	step 1	1	0	1	1	1
5	1	step 9	0	step 3	1	1	0	1	1	1
6	1	step 8	step 7	step 6	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

Observation 11: Row 3, Column 1

It is suspected that the robot would have turned the corner without any backtracking. However, the assumption was made that using a range of possible calculated locations would introduce additional complexity and uncertainty. So merely using the calculated location with the steps given produced these results.

```
Step 11
Getting robot location based on emissions (distances).
Calculated i: 1 j: 2
North of previous.
Blocked, moved West.
North of previous.
Moved South.
North of previous.
Moved South.
Plot Cells
```

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	1	1	1
2	1	1	0	1	1	1	0	1	1	1
3	1	step 11	0	1	1	1	0	1	1	1
4	1	step 10	0	step 2	step 1	1	0	1	1	1
5	1	step 9	0	step 3	1	1	0	1	1	1
6	1	step 8	step 7	step 6	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

Conclusion

It is likely there are sophisticated applications that consider the connecting locations by matching minimum and maximum location estimates. However, this implementation keeps it simple. Use the calculated location as a direction, then adjust that location until it is contiguous with the previous location.

```
Robot Locations
step: 0 i: 4 j: 4
step: 1 i: 4 j: 3
step: 2 i: 5 j: 3
step: 3 i: 6 j: 3
step: 4 i: 6 j: 2
step: 5 i: 6 j: 3
step: 6 i: 6 j: 2
step: 7 i: 6 j: 1
step: 8 i: 5 j: 1
step: 9 i: 4 j: 1
step: 10 i: 3 j: 1
```

Part 2: Software Familiarization

Pomegranate

Pomegranate is an extensive library of classification programs, one being for Hidden Markov Models. In this exercise, the region of high probability for free cells is determined.

The first step is to describe two distributions. The first would be for the known distribution of obstacles and free cells. The other would be for the desired distribution. In this exercise, the desired distribution is 90 percent free cells and 10 percent obstructed cells.

Next, states are defined. These are determined by the distributions. The states are named. *Chaos* is from the current distribution. *Freedom* from the desired distribution.

The model is then created and the states are applied.

Transitions are then added:

- There is a 50 percent probability of transition from the starting position to Chaos.
- There is a 50 percent probability of transition from the starting position to Freedom.
- There is a 90 percent probability of transition from Chaos to Chaos.
- There is a 10 percent probability of transition from Chaos to Freedom.
- There is a 10 percent probability of transition from Freedom to Chaos.
- There is a 90 percent probability of transition from Freedom to Freedom.

Results

As expected there is a large region of 90 percent free cells in the Southern part of the grid. This visually intuitive. If the Viterbi Algorithm were to have this region as a *bias* as one of the factors, it could

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	1	1	1
2	1	1	0	1	1	1	0	1	1	1
3	1	1	0	1	1	1	0	1	1	1
4	1	1	0	1	1	1	0	1	1	1
5	1	1	0	1	1	1	0	1	1	1
6	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

have used as a tie-breaker when the transition template had the same probability for more than one direction. The bias would nudge the robot to the center of Freedom, the region of free cells.

There are two Freedom regions, both indicated by green cells.

Part 3: Applications

Machine Translation A field of computational linguistics that studies software to translate text or speech from on language to another.

Speech Recognition Applications where commands or data is entered using natural language speech. Siri, to you work for the government? The applications range from selecting television programming to hands-free control of automotive devices.

Handwriting Recognition This is used when entering a signature on a portable point of sale (POS) device or for scanning handwritten documents such as a will where the handwritten version is legally binding. Also used in fraud detection when the signature does not appear legitimate .

Computational Finance Computational finance deals with the calculation of risk necessary for assigning a value to interest.

Gene Prediction The *Pomegranate* example used in this example was adaptation of a tutorial. The tutorial was finding islands of gene clustering. In this exercise, it was used for free cell clustering.