

INF552 Homework 6

Group members and contribution :

Krishna Akhil Maddali (Code development and report)

Yash Shahapurkar (Code development, research, report)

Myungjin Lee (research and report)

Part 1: Implementation

- Output after running the program

We have two types of output which are one with precision of the floating point is 0.1 and the other one is with full precision of the floating point. The results are somehow different as below.

Output of hmm_robot_pre1.py (precision : 0.1)

The path from first position to last position is

(5, 3)

(6, 3)

(7, 3)

(7, 2)

(7, 1)

(7, 2)

(7, 1)

(6, 1)

(5, 1)

(4, 1)

(3, 1)

Output of hmm_robot.py (full precision)

The path from first position to last position is

(5, 3)

(6, 3)

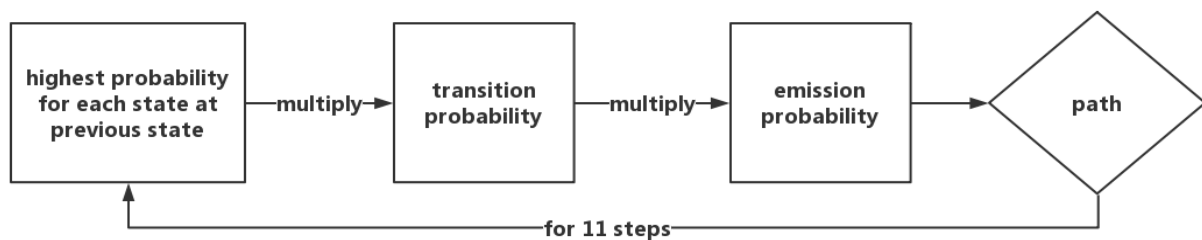
(7, 3)

(8, 3)

(8, 2)

(7, 2)
(7, 1)
(6, 1)
(5, 1)
(4, 1)
(3, 1)

- Data structures



- Code-level optimization

- Conversion 2D label to 1D : In order to convert the coordinate which is in 2 dimension to 1 dimension, we have implemented convert function.
- Vectorization : The code is written in a way that the matrix has been done a linear transformation which converts the matrix into a column vector.

- Challenges

- Emission probability : Defining the emission probability was quite challenging. We first take the observation which is given in the data set. The lower and upper limit of the range were also taken so that observation, lower and upper limit are our parameter to calculate the emission probability.
- Back tracking : Once we finish the forward calculation in viterbi algorithm, we need to track back to find out which state leads to the maximum probability of the next state. The back tracking part was a little bit confusing to figure out the previous state.

Part 2: Software Familiarization

We have used one of hidden markov model library package to compare the result from our code with the one from the

library.(https://pypi.python.org/pypi/hidden_markov/0.3.1)

However, since this type of problem cannot be solved by library package, we have decreased the dimension to 1 dimension problem which has tower 1, coordinate is only x dimension, grid world is first column of it.

These are result from our code (1 dimensional problem) and library (also 1 dimension). The result shows exactly same path how robot would move through the grid.

Output of our implemented code (1d_hmm_robot.py)

```
[9, 8, 9, 8, 7, 8, 7, 6, 5, 4, 3]
```

Output of library code (1d_lib_hmm.py)

```
['9', '8', '9', '8', '7', '8', '7', '6', '5', '4', '3']
```

How to improve our code

- In this code, the performance from our code and the library code is exactly the same based on the 1 dimensional problem, our code has been implemented well.

Part 3: Applications

- Sequence alignment :

A way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences is called sequence alignment. Aligned sequences of nucleotide or amino acid residues are typically represented as rows within a matrix. Sequence alignments are also used for non-biological sequences, such as calculating the edit distance cost between strings in a natural language or in financial data.

- Speech recognition :

Speech recognition is the interdisciplinary subfield of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text by computers. It is also known as "speech to text". It incorporates knowledge and research in the linguistics, computer science, and electrical engineering fields.