INF552 Machine Learning for Data Informatics HW7

1. Contribution:

a. Shih-Yu Lai (40%): (Monday)

Email: shihyula@usc.edu

USC-ID: 7183984563

Implement the HMM.

b. Shiuan-Chin Huang (30%): (Monday)

Email: shiuanch@usc.edu

USC-ID: 9815633745

Write the ReadMe.

c. Danhui Wu (30%): (Tuesday)

Email: danhuiwu@usc.edu

USC-ID: 8038800437

Implement the hmm learn library and write the ReadMe.

2. Steps of the algorithm and data structure

In HMM.py, we use Viterbi Algorithm and divide our implementation into following steps:

Step 1: Load the hmm-data.txt and store the free cells, towers and noisy distances into a list, respectively.

Step 2: Find all the possible states for the 11 time-steps by calculating the Euclidean distances d between free cells and towels and applying the interval constraints [0.7d, 1.3d] and store them into a list.

Step 3: For each state, find all the neighboring cells.

Step 4: For each state, calculate the transition probability to each of its corresponding neighboring cells and store the result as dictionary.

Step 5: By Viterbi Algorithm, dynamically calculate the possibility from a state to the next one via all paths

Step 6: Backtracking to find the path with maximum probability.

3. Our Outcome

HMM.py

Path:

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

```
Noisy :
[[6.3, 5.9, 5.5, 6.7], [5.6, 7.2, 4.4, 6.8], [7.6, 9.4, 4.3, 5.4], [9.5, 10.0, 3.7, 6.6], [6.0, 1
0.7, 2.8, 5.8], [9.3, 10.2, 2.6, 5.4], [8.0, 13.1, 1.9, 9.4], [6.4, 8.2, 3.9, 8.8], [5.0, 10.3, 3
.6, 7.2], [3.8, 9.8, 4.4, 8.8], [3.3, 7.6, 4.3, 8.5]]
states :
defaultdict(<class 'list'>, {(3, 4): [0], (3, 5): [0], (4, 3): [0, 1], (4, 4): [0], (4, 5): [0],
(5, 3): [0, 1, 2, 7], (5, 4): [0, 1], (5, 5): [0], (6, 3): [0, 1, 2, 8], (6, 4): [0, 1, 2], (5, 1
): [1, 7, 8, 9], (7, 3): [1, 2, 3, 4, 5], (7, 4): [1, 2, 3], (6, 5): [2], (7, 5): [2], (8, 4): [2
, 3], (8, 5): [2], (9, 4): [2], (7, 2): [3, 4, 5], (8, 3): [3, 4, 5], (9, 3): [3, 5], (7, 1): [4,
 6], (8, 2): [4, 5], (9, 2): [5], (7, 0): [6], (5, 0): [7, 8, 9], (6, 0): [7, 8], (6, 1): [7, 8],
 (4, 0): [8, 9, 10], (4, 1): [8, 9, 10], (3, 0): [9, 10], (3, 1): [9, 10]})
Path :
[(5, 5), (5, 4), (6, 4), (7, 4), (7, 3), (7, 2), (7, 1), (6, 1), (5, 1), (4, 1), (3, 1)]
```

4. Challenge

The formulas and equations are very complicated, it takes us lots of time to understand the meanings of mathematic symbols and relations between each other.

It is also difficult to verify the correctness of results from each step, however, if something goes wrong in one step, it will affect the correctness of the whole process, so it is also challenging to be careful to all the steps.

5. Optimization

a. We build multiple list arrays for free cells, neighboring cells, distances to towers, states, transition, possible paths…to make it easier and more efficient to understand the structure of our code and maintain our code.

b. we memorize the previous status and keep tracking of the maximum probability of current state, which allows us to avoid repeatedly doing the same calculation and thus speed up our program.

6. Software Familiarization

We use hmmlearn library to implement the Hidden Markov Models.

Implementation Steps :

a. Build the model

   model = hmm.MultinomialHMM(n_components=n_states)

   n_components: Number of states in each timestamp.

b. Set the start probability, the transfer matrix and emission probability model.startprob_= start_probability model.transmat_= transition_probability

model.emissionprob_= emission_probability

c. Predict

Model.predict(observation_sequence)

Comparison and Suggestion of Improvements

- ✓ Our model can only train discrete data. But GaussianHMM and GMMHMM in the hmmlearn library can be used for continuous data.
- ✓ Our model only resolved 'decoding problem' (Given a known model and observation sequence , what is the optimal hidden state sequence) using Viterbi algorithm, but the library all typical HMM problems in the following.

Improve our code so that it could train the continuous observation and solve other 2 typical HMM problems(1. Given a known model what is the likelihood of sequence of observation happening; 2. Given observation sequence and number of hidden states, what is the optimal model which maximizes the probability of the observation sequence)

7. Applications

Hidden Markov models (HMMs) have many applications such as speech recognition, mental task, classification, biological analysis, and anomaly detection.