Programming Assignment 7: Hidden Markov Models

Srihari Akash Chinam
USC ID - 2953497706
Net ID - chinam

Sayan Nanda
USC ID - 2681592859
Net ID - snanda

Implementation

- Language used - Python version 2
- Libraries used - Numpy (arrays and matrices), math

Hidden Markov Models + Viterbi Algorithm

The given data was loaded and separated as tower locations, free spaces in the grid world and noisy distances to each tower for every instance. After that the transmission matrix was generated which contained the probability of going from one state to another. A matrix containing prior probabilities was initialized uniformly. The observation space was generated by calculating the Euclidian distance from every point and generating them in the range of [0.7d, 1.3d]. The probabilities for each observation was not generated initially, but was computed lazily. The output achieved for the 11 time steps is shown below in Table 1.

Table 1 Output obtained by Viterbi algorithm for given inputs

Time Step	Coordinates
1	(5,3)
2	(6,3)
3	(6,4)
4	(7,4)
5	(7,3)
6	(7,2)
7	(7,1)
8	(6,1)
9	(5,1)
10	(4,1)
11	(3,1)

To generate an emission matrix, that is, a matrix containing the probabilities of emitting an observation from a given state, it would take up a lot of time and space. With the given range of error as [0.7d,1.3d] with a one decimal interval, the number of unique observations generated on the whole by every point in the grid is in the range of 200 million. To avoid this, the emission probabilities were calculated lazily. Library implementations of the Viterbi algorithm on HMMs were not used for this assignment as the input observation space was a matrix of length 200 million and the emission matrix was a matrix of dimensions (100,200 million).

Software Familiarization

Scikit learn provides an implementation of Hidden Markov Models (HMMS). For the library implementation, some parameters are expected to be provided. These include a start probability vector, a transition matrix and the emission probability of the observable. Typical problems expect the emission probability to have a multinomial or gaussian or other distribution. However the given problem which we were given to implement required a custom emission matrix. This was due to the presence of obstacles.

The scikit learn implementation can solve three kinds of problems. It can estimate the optimal sequence of hidden states, calculate the likelihood of data or estimate model parameter. This first is similar to the problem we have at hand which we have solved using the Viterbi algorithm. This is however customizable with the use of a parameter in the library implementation. This is done by changing the ‘algorithm’ parameter. Currently the algorithms supported are Viterbi and a maximum a-posteriori estimation (map) algorithm.

Applications

Hidden Markov Models have are used in computational biology. They are used for statistical modeling, database searching and alignment of protein families and protein domains. HMMs are found to be effective in distinguishing between family members and non-family members. This is done by examining the sequences of globin, kinase and EF-hand HMMs. HMMs are found to have a advantage as compared to the existing techniques in terms of lower rates of false negatives and false positives [1].

A combination of HMMs, vector quantization and linear predictive coding analysis are used in speaker independent, isolated word recognition. The HMMs need to be trained for the vocabulary being recognized. Classification consists of computing the probability of generating the test word with each word model and choosing the word model that gives the highest probability. Research done on a 100 talker data set was found to have a 96.5% accuracy [2].

HMMs also find many applications in bioinformatics such as prediction of protein coding regions in genome sequences, modeling families of related DNA or protein sequences, prediction of secondary structure elements in proteins [3]. Other applications include computational finance, speech synthesis, gene prediction, activity recognition, partial discharge and cryptanalysis.

Individual Contributions

Sayan Nanda

- Implementation of models
    - Preprocessing Data - Grid, Tower locations, Noisy data
    - Viterbi Algorithm
- Documentation
    - Software Familiarization
    - Applications

Srihari Akash Chinam

- Implementation of models
    - Initialization of parameters - Observation space, transmission matrix, prior probabilities
    - Viterbi Algorithm
- Documentation
    - Implementation

References

[1] Hidden Markov Models in Computational Biology: Applications to Protein Modeling, http://www.sciencedirect.com/science/article/pii/S0022283684711041

[2] On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition, http://onlinelibrary.wiley.com/doi/10.1002/j.1538-7305.1983.tb03115.x/full

[3] http://users-cs.au.dk/cstorm/courses/PRiB_f12/slides/hmm-bioinf.pdf