Lab 7 - Viterbi Algorithm

Submitted By:
Shashi Shivaraju
C88650674

Clemson University
November 29, 2018

Abstract

This report explains the process involved in finding the most likely sequence of states for a given Hidden Markov Model(HMM) using Viterbi Algorithm.

1 Introduction

This report considers the problem of predicting the hidden state sequence of a Hidden Markov Model(HMM), that most likely describes a given observation sequence using Viterbi algorithm.A Markov model is a stochastic model used to model randomly changing systems Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (i.e. hidden) states. The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states called the Viterbi Path that results in a sequence of observed events.

The Viterbi algorithm is named after Andrew Viterbi, who proposed it in 1967 as a decoding algorithm for convolutional codes over noisy digital communication links. Viterbi Path and Viterbi Algorithm have become standard terms for the application of dynamic programming algorithms to maximization problems involving probabilities. The algorithm has found universal application in decoding the convolutional codes used in both CDMA and GSM digital cellular, dial-up modems, satellite, deep-space communications and 802.11 wireless LANs.It is now also commonly used in speech recognition, speech synthesis, diarization, keyword spotting, computational linguistics, and bioinformatics.

This report describes implementing Viterbi Algorithm for predicting the sequence of states for a given sequence of observed events from the given Hidden Markov Model(HMM).

2 Methods

The given Hidden Markov Model(HMM) consists of two states, represented as H and L. The prior probabilities are 0.5, 0.5.The state transition probabilities are 0.5, 0.5 for state H and 0.4, 0.6 for state L. Each state observes a discrete value that takes on one of four values A, C, G, T. The emission probabilities of these values are 0.2, 0.3, 0.3, 0.2 for state H and 0.3, 0.2, 0.2, 0.3 for state L.The HMM in consideration is represented in figure 1.

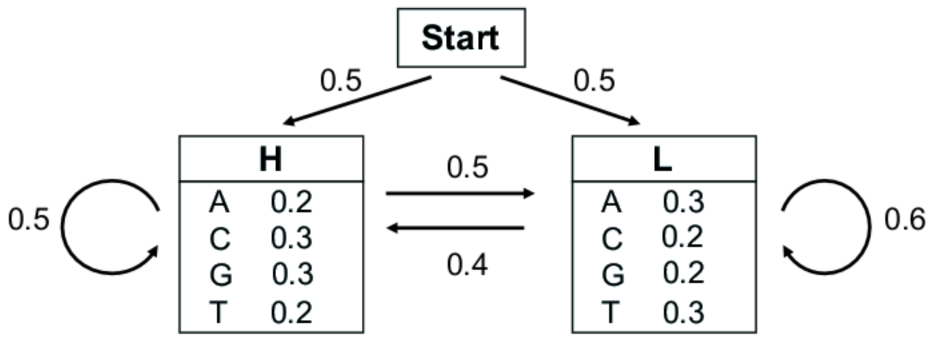

Figure 1: Given Hidden Markov Model

The implementation of Viterbi Algorithm for predicting the sequence of states for the given sequence of observed events from the given HMM is described in below section.

2.2 Implementation of Viterbi Algorithm

The Viterbi algorithm is a dynamical programming algorithm that allows us to compute the most probable path of system state for the given sequence of observations.

Let us consider a Hidden Markov Model(HMM) which consists of k states.The probability of the most probable path ending in state l with observation i is given as:

$$p_l(i, x) = e_l(i) \max_k (p_k(j, x-1).p_{kl}); \tag{1}$$

Here,
x = Position of observation i in sequence of observations.
$e_l(i)$ = Probability of observation i in state l.
$p_k(j, x-1)$ = Probability of the most probable path ending at position $x-1$ in state k with observation j.
p_{kl} = Probability of the transition from state l to state k.

According the equation 1,to determine the maximum probability of the xth observation in the sequence,the knowledge of the maximum probability of the $(x-1)$th observation is sufficient. Thus by using equation 1 recursively (i.e from the first element to the last element of the observation sequence) the most probable path and its probability is determined.

The probability of the most probable path for the the observation sequence is obtained at the last position and this path can be retrieved by back-tracking.

Please refer the Appendix for the Matlab implementation of Viterbi Algorithm.

3 Results

Two sequence of observations O_1 = GGCACTGAA and O_2 = TCAGCGGCT from the Hidden Markov Model shown in Figure 1 are provided. Using equation 1,the most probable state path for each of the sequence of observations is determined.

1. Observation Sequence: GGCACTGAA

Seq.	G	G	C	A	C	T	G	A	A
High	**-2.73**	**-5.47**	**-8.21**	-11.53	-14.00	-17.32	-19.53	-22.86	-25.65
Low	-3.32	-6.05	-8.79	**-10.94**	**-14.00**	**-16.48**	**-19.53**	**-22.01**	**-24.48**
Max	-2.73	-5.47	-8.21	-10.94	-14.00	-16.48	-19.53	-22.01	-24.48
Path	**H**	**H**	**H**	**L**	**L**	**L**	**L**	**L**	**L**

Table 1: Table of Maximum Probability of the observations in the sequence GGCACTGAA.

The most probable state path for the Observation Sequence: GGCACTGAA is HHHLLLLLL and its probability is $2^{-24.48} = 4.2517e - 08$.

2. Observation Sequence: TCAGCGGCT

Seq.	T	C	A	G	C	G	G	C	T
High	-3.32	-5.79	-9.11	**-11.32**	**-14.06**	**-16.80**	**-19.53**	**-22.27**	-25.59
Low	**-2.73**	**-5.79**	**-8.26**	-11.32	-14.38	-17.38	-20.12	-22.86	**-25.01**
Max	-2.73	-5.79	-8.26	-11.32	-14.06	-16.80	-19.53	-22.27	-25.01
Path	**L**	**L**	**L**	**H**	**H**	**H**	**H**	**H**	**L**

Table 2: Table of Maximum Probability of the observations in the sequence TCAGCGGCT.

The most probable state path for the Observation Sequence: TCAGCGGCT is LLLHHHHHL and its probability is $2^{-25.01} = 2.9525e - 08$

4 Conclusion

The report describes in detail the analytical process involved in implementing the Viterbi Algorithm to predict the most probable state path for the given sequence of observations from the provided Hidden Markov Model (HMM).

Viterbi Algorithm utilizes the knowledge of the HMM model parameters and for a particular output sequence,it finds the state sequence that is most likely to have generated that output sequence by finding a maximum over all possible state sequences.

Appendix

Matlab Code for Viterbi Algorithm

```
% FILE NAME       : viterbi_algo.m
%
% DESCRIPTION     : Code to implement Viterbi Algorithm.
%
% PLATFORM                : Matlab
%
% DATE                      NAME
% 27th-Nov-2018      Shashi Shivaraju

clc; %clear all the varaibles
close all;%close all windows
clear; %clear the screeeen

%Given State Sequence Order in HMM = [A C G T] = [1 2 3 4]

%Given State Sequence
%State_Seq = ['G', 'G', 'C', 'A', 'C', 'T', 'G', 'A', 'A' ];
%State_Seq_Index = [3 3 2 1 2 4 3 1 1];

State_Seq = ['T', 'C', 'A', 'G', 'C', 'G', 'G', 'C', 'T' ];
State_Seq_Index = [4 2 1 3 2 3 3 2 4];
size = length(State_Seq); %Length of the given sequence

%Probabilities of the HMM states
Prob_iH = -1; %initial to High
Prob_iL = -1; %initial to Low
Prob_HH = -1; %High to High
Prob_HL = -1; %High to Low
Prob_LL = log2(0.6); %Low to Low
Prob_LH = log2(0.4); %Low to High
Prob_H = [log2(0.2), log2(0.3), log2(0.3), log2(0.2)]; % State = [A C G T] = [1 2 3 4]
Prob_L = [log2(0.3), log2(0.2), log2(0.2), log2(0.3)]; % State = [A C G T] = [1 2 3 4]


%Matrix to store the result
%row 1 = prob. at H state
%row 2 = prob. at L state
%row 3 = max(row1,row2) at each index
```

```matlab
%row 4 = 0/1 (1 = H state;0 = L state)
Result = zeros(4, size);

%Loop through the given sequence to calulate the max. prob. at High and Low States
for i = 1:size

        %Transition from intial State
        if( i == 1)
                Result(1,i) = Prob_iH + Prob_H(State_Seq_Index(i)); %initial -> H ->State(i)
        Result(2,i) = Prob_iL + Prob_L(State_Seq_Index(i)); %initial -> L ->State(i)

                %select the state with highest prob.
                if(Result(1,i) > Result(2,i))
                        Result(3,i) = Result(1,i);
            Result(4,i) = 1;
        else
            Result(3,i) = Result(2,i);
            Result(4,i) = 0;
        end

    else % Transition from State(i-1) -> State(i)
            Result(1,i) = Prob_H(State_Seq_Index(i)) + max( (Result(1, i-1) + Prob_HH), (Result(2, i-1) + Prob_LH));
            Result(2,i) = Prob_L(State_Seq_Index(i)) + max( (Result(1, i-1) + Prob_HL), (Result(2, i-1) + Prob_LL));

                    %select the state with highest prob.
            if(Result(1,i) == Result(2,i)) %if both states have equal prob,stay with the previous state
                Result(3,i) = Result(1,i);
                Result(4,i) = 2;
            elseif( Result(1,i) < Result(2,i))
                Result(3,i) = Result(2,i);
                Result(4,i) = 0;
            else
                Result(3,i) = Result(1,i);
                 Result(4,i) = 1;
            end
     end
end
```

```
%back tracking  to find  the  path  which  corresponds  to  the
   highest probability
for i = size:-1:1
   if(Result(4,i) == 2 && i ~= size)
        Result(4,i) = Result(4,i+1);
   elseif(Result(4,i) == 2 && i == size)
        Result(4,i) = Result(4,i-1);
   end
end


 disp("Most  probable state path  is ");
    for i = 1:size
       if(Result(4,i) == 1)
           disp('H');
       elseif(Result(4,i) == 0)
            disp('L');
       end
    end

  disp("Result Matrix = ");
  disp(Result);
```

References

1.Viterbi Algorithm from Wikipedia
https://en.wikipedia.org/wiki/Viterbi

2.TeX Live - TeX Users Group
https://www.tug.org/texlive/