Viterbi Algorithm

Lab-7

Submitted by:

Priyanshu Rawat

---

1 Introduction

The **Viterbi algorithm** is based on dynamic programming for obtaining the maximum posteriori probability estimate of the most likely sequence of hidden states; called the Vertibi path - a sequence of observed events in the context of the hidden Markov models (HMM).
The Viterbi algorithm is commonly used in speech synthesis, speech recognition, keyword spotting, computational linguistics and other similar applications. For instance, in speech recognition (speech-to-text) the linguistic signal is treated as the observed sequence of events, and a string of text is taken to be as the 'hidden cause' of the acoustic signal. Eventually, the Vertibi algorithm finds the most likely string of text given the acoustic signal.

1.1 Problem Statement

The first problem consists of two states, labeled H and L, which can be given numerical values of 0 and 1. The prior probabilties are 0.5, 0.5. The state transition probabilties are 0.5, 0.5 for state 0 and 0.4, 0.6 for state 1. Each state observes a discrete value that takes on one of four values A, C, G, T that can be given numerical values 0, 1, 2, 3. The emission probabilities of these values are 0.2, 0.3, 0.3, 0.2 for state 0 and 0.3, 0.2, 0.2, 0.3 for state 1.
The Viterbi algorithm should be run twice. On the first run, the input should be the same as given in the example: GGCACTGAA. On the second run, the input should be: TCAGCGGCT. For both runs, the table of max probabilities at each data index and the most probable path must be presented.

2 Results and discussion

2.1 Case I: For observation sequence - GGCACTGAA

The most probable path is: **HHHLLLLLL**

The probabilty of the most probable path is: $4.25e^{-08}$

The table of maximum probabilities at each data index in Logarithmic scale:

	G	G	C	A	C	T	G	A	A
H	-2.737	-5.474	-8.211	-11.533	-14.007	-17.329	-19.540	-22.862	-25.657
L	-3.322	-6.059	-8.796	-10.948	-14.007	-16.481	-19.540	-22.014	-24.487

2.2 Case II: For observation sequence - TCAGCGGCT

The most probable path is: **LLLLHHHHL**

The probabilty of the most probable path is: $2.95e^{-08}$

The table of maximum probabilities at each data index in Logarithmic scale:

	T	C	A	G	C	G	G	C	T
H	-3.322	-5.796	-9.118	-11.329	-14.066	-16.803	-19.540	-22.277	-25.598
L	-2.737	-5.796	-8.270	-11.329	-14.388	-17.388	-20.125	-22.862	-25.014

3 Conclusion

The Viterbi algorithm uses the parameters of the HMM model and an output sequence to find the most likely state sequences that have generated that output sequence. Additionaly, it computes the most probable path as well its probability. The algorithm works by finding the maximum value over all possible states sequences.
Sometimes, there are often many state sequences that can produce the same specified output sequence, but with different probabilities. Using the Vertibi algorithm, it is possible to calculate the probability for the HMM model to generate that output sequence by doing the summation over all possible state sequences. This also can be done efficiently using the Forward algorithm or the Backward algorithm, which is also a dynamical programming algorithm.

4 Appendix (MATLAB script)

```
clc;
clear;

states = [0 1]; % H=0, L=1
statevalues = [0 1 2 3]; % A=0, C=1, G=2, T=3

% Emission probabilites
statevalues_H = log2([0.2 0.3 0.3 0.2]);
statevalues_L = log2([0.3 0.2 0.2 0.3]);

% State transitions proabilites
statetrans_HH = log2([0.5]);
statetrans_HL = log2([0.5]);
statetrans_LL = log2([0.6]);
statetrans_LH = log2([0.4]);

% Intial probabilties
initial_H = log2([0.5]);
initial_L = log2([0.5]);

% 1st Observations
obs = ['G' 'G' 'C' 'A' 'C' 'T' 'G' 'A' 'A'];

% 2nd Observations
% obs = ['T' 'C' 'A' 'G' 'C' 'G' 'G' 'C' 'T'];

% Observations to values
obs(obs=='A')='0';
obs(obs=='C')='1';
obs(obs=='G')='2';
obs(obs=='T')='3';
obs = str2num(obs);
obsvalues = num2str(obs)-'0';

% Vertibi Algorithm
for i=1:9
    if i==1
        switch obsvalues(i)
            case 0
                p(1,i) = initial_H + statevalues_H(1); % H
                p(2,i) = initial_L + statevalues_L(1); % L
            case 1
                p(1,i) = initial_H + statevalues_H(2); % H
                p(2,i) = initial_L + statevalues_L(2); % L
```

```
            case 2
                p(1,i) = initial_H + statevalues_H(3); % H
                p(2,i) = initial_L + statevalues_L(3); % L
            case 3
                p(1,i) = initial_H + statevalues_H(4); % H
                p(2,i) = initial_L + statevalues_L(4); % L
        end
    else
        switch obsvalues(i)
            case 0
                p(1,i)=statevalues_H(1)+max(p(1,i-1)+statetrans_HH,p(2,i-1)+statetrans_LH);
                p(2,i)=statevalues_L(1)+max(p(1,i-1)+statetrans_HL,p(2,i-1)+statetrans_LL);
            case 1
                p(1,i)=statevalues_H(2)+max(p(1,i-1)+statetrans_HH,p(2,i-1)+statetrans_LH);
                p(2,i)=statevalues_L(2)+max(p(1,i-1)+statetrans_HL,p(2,i-1)+statetrans_LL);
            case 2
                p(1,i)=statevalues_H(3)+max(p(1,i-1)+statetrans_HH,p(2,i-1)+statetrans_LH);
                p(2,i)=statevalues_L(3)+max(p(1,i-1)+statetrans_HL,p(2,i-1)+statetrans_LL);
            case 3
                p(1,i)=statevalues_H(4)+max(p(1,i-1)+statetrans_HH,p(2,i-1)+statetrans_LH);
                p(2,i)=statevalues_L(4)+max(p(1,i-1)+statetrans_HL,p(2,i-1)+statetrans_LL);
        end
    end
end

% Getting the max probabilites at every index
for j = 1:length(p)
    k(j) = max(p(1,j),p(2,j));
    seq(:,j) = k(j)==p(:,j);
end

% Getting the most probable path
for k = 1:length(seq)
    if seq(1,k)==1
    seqstates(k) = 'H';
    end
    if seq(2,k)==1
    seqstates(k) = 'L';
    end
end

% -- Printing stuff -- %

% Most probable path
fprintf("The most probable path is %s.", seqstates);
```

```
% Probability of most probable path
if seqstates(end) == 'L'
    fprintf("The probabilty of the most probable path is: \n %i", 2.^p(2,9));
elseif seqstates(end) == 'H'
    fprintf("The probabilty of the most probable path is: \n %i", 2.^p(1,9));
end

% Table of max probabilities in logarithmic scale
fprintf("The table of max probabilities in logarithmic scale:
\n %i %i %i %i %i %i %i %i %i \n %i %i %i %i %i %i %i %i %i", p);

% Table of max probabilities
fprintf("The table of max probabilities:
\n %i %i %i %i %i %i %i %i %i \n %i %i %i %i %i %i %i %i %i", 2.^p);
```