Hidden Markov Model: Viterbi Method for decoding

Yang You

1 Introduction

Hidden Markov Model (HMM) is an extension of Markov Chain Model with unobserved states [3], which is a graphical model for time series. In the hidden markov model, we cannot observe the states directly, which means those hidden states are hidden state sequence I and the observations of those states form a sequence called observation sequence O. Each position in the sequence is considered as specified time.

Hidden Markov Model is specified by an initial probability distribution π, a transition probability matrix A, an emission probability (measurement probability) matrix B, a set of all possible states Q and a set of all possible observations V. The definitions of those components are as follows:

$$Q = \{q_1, q_2, q_3 ... q_N\}$$

$$V = \{v_1, v_2, v_3 ... q_M\}$$

The number N in the set Q stands for the maximum number of possible states, M stands for the maximum number of possible measurement (observations) for one state. To a given time series with length T, the hidden state sequence I and observation sequence O are defined as:

$$I = \{i_1, i_2, i_3 ... i_N\}$$

$$O = \{o_1, o_2, o_3 ... o_N\}$$

From a state q_i at time t , the probability of transferring to another state q_j at time $t+1$ can be defined as.

$$a_{ij} = P(i_{t+1} = q_j | i_t = q_i), \quad i = 1, 2, 3...N; j = 1, 2, 3...N$$

The transition matrix A is consist of a_{ij} in each dimension.

$$A = [a_{ij}]_{N*N}, \quad s.t. \sum_{j=1}^{N} a_{ij} = 1, \forall i$$

From each state q_i at time t, there is a probability of having an observation v_k which can be defined as.

$$b_i(k) = P(o_t = v_k | i_t = q_t), \quad k = 1, 2, 3...M; i = 1, 2, 3...N$$

Then the observation (measurement) probability matrix can be define as follows:

$$B = [b_i(k)]_{N*M}, \quad s.t. \sum_{k=1}^{M} b_{ik} = 1, \forall i$$

The initial probability distribution π is defined by the probabilities of all possible states in the initial time, which can be written as:

$$\pi = (\pi_i), \quad \pi_i = P(i_1 = q_i) \quad s.t. \sum_{i=1}^{N} \pi_i = 1$$

2 Problem Setting

Now we consider a problem as follows, there are four boxes, each box contained two types of balls, red and black respectively. The numbers of balls and boxes are given in table 1.

Table 1: Number of balls

Box Reference	1	2	3	4
Number of Red Balls	5	3	6	8
Number of Black Balls	5	7	4	2

At the first time user selects 1 box randomly among those 4 boxes, then pick one ball randomly from the selected box. Then, switch the box in the next time, the rules for switching are as follows: If the current box is the 1st box, then the next time it must be the 2nd box selected. If the current box is the 2nd or 3rd, then there are probabilities of 0.4 and 0.6 switching to their left or right box. If the current box is the 4th one, there are equal possibilities about 0.5 to stay in the 4th box or switch to the 3rd box. The reference of box is not recorded in each time, only the colour of the picked ball is recorded. Thus, the hidden state I is the reference of box and the observation O is the colour of ball.

Now an observation sequence is given as $O = \{Red, Black, Red\}$, what's the most possible hidden state sequence according to the given observation?

3 Solution

According to the Hidden Markov Model, we need an input vector λ to specified the model, which is defined as follows:

$$\lambda = (A, B, \pi)$$

By analyzing the question as fig 1 and fig 2 , we can construct the transition probability matrix A and measurement matrix B as follows:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0.4 & 0 & 0.6 & 0 \\ 0 & 0.4 & 0 & 0.6 \\ 0 & 0 & 0.5 & 0.5 \end{bmatrix} \quad B = \begin{bmatrix} 0.5 & 0.5 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ 0.8 & 0.2 \end{bmatrix}$$

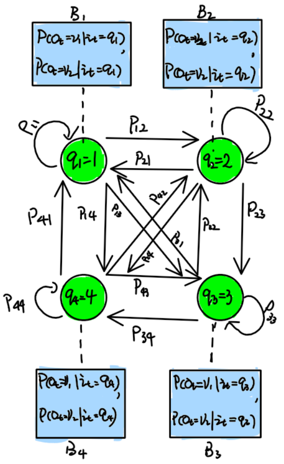

Figure 1: HMM model

$$A = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}$$

$$B = \begin{bmatrix} p(o_t = v_1|i_t = q_1) & p(o_t = v_2|i_t = q_1) \\ p(o_t = v_1|i_t = q_2) & p(o_t = v_2|i_t = q_2) \\ p(o_t = v_1|i_t = q_3) & p(o_t = v_2|i_t = q_3) \\ p(o_t = v_1|i_t = q_4) & p(o_t = v_2|i_t = q_4) \end{bmatrix}$$

Figure 2: Transition matrix and emission matrix

As the first selection of box is randomly, thus all 4 box share the same possibility of being selected. Then the initial distribution π can be defined as follows:

$$\pi = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}$$

3.1 Viterbi Algorithm

Viterbi algorithm is a dynamic programming method to solve the decoding problem of hidden markov model [1]. Using dynamic programming method to find the most likely path among nodes, the node represent the state and the path is the most likely sequence of hidden states.

From the time $t = 1$ and state i , we calculate and select the sub-path which has the biggest probability of transition to the next state until time $t = T$. When we select the optimal sub-path, the nodes i_t^* is stored, then by finding all the optimal nodes in the optimal path, the most likely hidden states sequence $I*$ is deduced as $I^* = \{i_1^*, i_1^*, ..., i_T^*\}$

Following the introduction of HMM[2], first, we need introduce a variable δ to store the probability of the most likely path in the time t and state i generating an observation o_t, which can be defined as:

$$\delta_t(i) = \underset{i_1,\dots,i_{t-1}}{max} P(i_t = i, i_{t-1}, ..., i_1, o_t, o_{t-1}, ..., o_1|A, B, \pi), \quad i = 1, 2, ..., N$$

Then, We can deduce the δ_{t+1} as follows:

$$\delta_{t+1}(i) = \underset{j}{max}[\delta_t(j) \cdot a_ji \cdot b_i(o_{t+1})], \quad i, j \in \{1, 2, ..., N\}; \quad t = 1, 2, ..., T - 1$$

Another variable ψ is used for storing the nodes of the most likely path, which is defined as:

$$\psi_t(i) = \underset{j}{argmax}[\delta_t(j) \cdot a_j i \cdot b_i(o_t)], \quad i,j \in \{1,2,...,N\}; \quad t = 1,2,...,T-1$$

3.2 Implementation in Matlab

According to the Viterbi Algorithm, first we need to initialize the variable $\delta_{t=1}$ and $\psi_{t=1}$.

```
%Initialization
Delta = zeros();
Psi = zeros();
Psi(:, 1) = 0;
O_1 = O(1,1);
for i = 1:M
    Delta(i,1) = Pi(i)*B(i,O_1);
end
```

Then, iterating until time $t = T$, δ_t and ψ_t will be updated each time.

```
% Iteration for calculating the rest values of
    delta
Delta_j = zeros();
for t = 2:K
    for j = 1:N
        for i = 1:M
            Delta_j(i,1) = Delta(i,t-1) * A(i,j)
                * B(j,O(t,1));
        end
        [max_delta_j,psi] = max(Delta_j); %Find
            the maximum probability
        Delta(j,t) = max_delta_j; %Store the
            maximum probability
        Psi(j,t) = psi; %Store the node
    end
end
```

In the last part, we need to record the probability and node in the last iteration, which is time $t = T$, then retracing the path to find each node to construct the most likely hidden states sequence.

```
%Recording the probability and node of the end
    of path
[P_best,psi_k] = max(Delta(:,K));
P = P_best; % The probability of the optimal
    path

%Retracing the path to find each node
I = zeros();
I(K,1) = psi_k;
for t = K-1:-1:1
```

```
    I(t,1) = Psi(I(t+1,1),t+1);
end
```

After comling those codes, we can obtain the hidden states sequence $I = \{1, 2, 3\}$ for a given observation sequence $O = \{Red, Black, Red\}$. The probability of this sequence is also computed which is $P = 0.0315$.

4 Conclusion

In this project, a decoding question has been analyzed based on hidden markov and solved by a dynamic programming method called Viterbi algorithm. The observations are discretized in this project, which could be an continuous distribution as an extension of this model. Moreover, there are other two applications of hidden markov model which are Likehood problem and Learning problem, those applications require different given parameters and use different methods to solve.

References

[1] G. D. Forney. The viterbi algorithm. *Proc. of the IEEE*, 61:268 – 278, March 1973.

[2] Hang Li. *Statistical Learning Methods*, pages 184–185. Tsinghua University Press, 2012.

[3] Lawrence R. Rabiner. Readings in speech recognition. chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

Appendix

```
% Main.m
A = [0, 1, 0, 0;
      0.4, 0, 0.6, 0;
      0, 0.4, 0, 0.6;
      0, 0, 0.5, 0.5;];
B = [0.5, 0.5;
     0.3, 0.7;
     0.6, 0.4;
     0.9, 0.2;];
 Pi = [0.25;
     0.25;
     0.25;
     0.25;];
O = [1;
      2;
      1;];

 [D,Psi,P,I] = Viterbi(A,B,Pi,O);

%Vertibi.m
function [Delta,Psi,I,P] = Viterb(A,B,Pi,O)
%Paramters
 A_size = size(A);
 O_size = size(O);
N = A_size(1,1);
M = A_size(1,2);
K = O_size(1,1);

% Initialization
 Delta = zeros();
 Psi = zeros();
 Psi(:,1) = 0;
 O_1 = O(1,1);
 for i = 1:M
     Delta(i,1) = Pi(i) * B(i,O_1);
```

```
end

% Iteration for calculating the rest values of
    delta
Delta_j = zeros();
for t = 2:K
    for j = 1:N
        for i = 1:M
            Delta_j(i,1) = Delta(i,t-1) * A(i,j)
                * B(j,O(t,1));
        end
        [max_delta_j,psi] = max(Delta_j); %Find
            the maximum probability
        Delta(j,t) = max_delta_j; %Store the
            maximum probability
        Psi(j,t) = psi; %Store the node
    end
end

%Recording the probability and node of the end
    of path
[P_best,psi_k] = max(Delta(:,K));
P = P_best; % The probability of the optimal
    path

%Retracing the path to find each node
I = zeros();
I(K,1) = psi_k;
for t = K-1:-1:1
    I(t,1) = Psi(I(t+1,1),t+1);
end
```