Theoretical Elaboration on the Hidden Markov Model used in some Selected Applications

Xinhao Luo

Shanghai Jiao Tong University

lxh666@sjtu.edu.cn

Abstract

Hidden Markov Model(HMM), which is introduced and studied in the late 1960s and early 1970s, is a type of Markov chain where the states cannot be observed directly, but can be inferred from a sequence of observation vectors. Each observation vector is represented by a probability density distribution of various states, and generated by a corresponding state sequence [1]. Since the 1980s, HMM has been successfully applied in machine recognition due to its abundant mathematical structures. In light of this, I attempt to carefully study the theoretical basis of this type of statistical model and show applications in some selected problems. Code is available at uaenalxh/HMM (github.com).

1 Introduction

The Hidden Markov Model(HMM) is a probabilistic model for time series, which describes the process of generating an **unobservable sequence of random states** by a hidden Markov chain, followed by generating an observed sequence by each state. The sequence of random states generated by the hidden Markov chain is called the **state sequence**, and each state generates an observation, resulting in an observed sequence of random events, called the **observation sequence**. Each position in the sequence can also be viewed as a time step [2].

HMM is determined by its initial probability distribution, state transition probability distribution, and observation probability distribution. The formal definition of Hidden Markov Model (HMM) is as follows:

Let Q be the set of all possible states and let V be the set of all possible

observations:

$$Q = \{q_1, q_2, \cdots, q_N\}, \qquad V = \{v_1, v_2, \cdots, v_N\}$$

Where N is the number of possible states, and M is the number of possible observations.

I is a state sequence of length T and O is an observation sequence corresponding to the state sequence:

$$I = (i_1, i_2, \cdots, i_T), \qquad O = (o_1, o_2, \cdots, o_T)$$

A is the transition probability matrix:

$$A = [a_{ij}]_{N\times N}, a_{ij} = P(i_{t+1} = q_j | i_t = q_t), i = 1,2,\cdots,N, j = 1,2,\cdots,N.$$

Which represents the probability of transitioning from state q_i to state q_j at time *t+1*, given that the system is in state q_i at time *t*.

B is the observation probability matrix:

$$B = [b_j(k)]_{N\times M}, b_j(k) = P(o_t = v_k | i_t = q_j), k = 1,2,\cdots,M, j = 1,2,\cdots,N.$$

π is the initial state probability vector:

$$\pi = (\pi_i), \pi_i = P(i_1 = q_i), i = 1,2,\cdots,N.$$

Which represents the probability of being in state q_i at time $t = 1$ [2].

HMM is determined by the vector of initial state probabilities π, the matrix of state transition probabilities *A*, and the matrix of observation probabilities *B*. The state sequence is determined by π and *A*, and the observation sequence is determined by *B*. Therefore, HMM can be represented using a triad of symbols:

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

Where *A,B* and π are the three elements that constitute the Hidden Markov Model [2].

Example 1 [3]. Suppose there are 4 boxes, each containing red and white balls. The number of red and white balls in each box is listed in the following table:

box	1	2	3	4
red	5	3	6	8
white	5	7	4	2

We generate an observation sequence of ball colors using the following procedure: Firstly, we randomly select one of four boxes with equal probability, and then randomly draw one ball from the selected box, record its color, and push it back. Next, we transition to the next box according to the following rule: if the current box is box 1, then we must transition to box 2; if the current box is box 2 or 3, we

transition to the left or right adjacent box with probabilities of 0.4 and 0.6, respectively; if the current box is box 4, we either stay in box 4 or transition to box 3, both with a probability of 0.5. After determining the next box, we randomly draw one ball from it, record its color, and replace it. This process is repeated five times to obtain an observation sequence of ball colors as:

$$O = (red, red, white, white, red)$$

During this process, we can only observe a sequence of ball colors and cannot observe the sequence of boxes from which the balls were drawn. In other words, the sequence of boxes is unobservable.

In this example, there are two random sequences: one is the sequence of boxes (state sequence), and the other is the sequence of observed colors of the balls (observation sequence). The former is hidden, and only the latter is observable. This is an example of a HMM, for which the state set, observation set, sequence length, and the three elements of the model can be clearly defined based on the given conditions.

The boxes correspond to states, and the set of states can be defined as follows:

$$Q = \{box_1, box_2, box_3, box_4\}, \quad N = 4$$

The colors of the balls correspond to observations, and the set of observations can be defined as follows:

$$V = \{red, white\}, \quad M = 2$$

The lengths of the state sequence and observation sequence can be defined as follows:

$$T = 5$$

The initial probability distribution can be defined as follows:

$$\pi = (0.25, 0.25, 0.25, 0.25)^T$$

The probability distribution of state transition is:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0.4 & 0 & 0.6 & 0 \\ 0 & 0.4 & 0 & 0.6 \\ 0 & 0 & 0.5 & 0.5 \end{bmatrix}$$

The probability distribution of observation is:

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

2 Applications

According to the definition, the HMM makes two fundamental assumptions:

Homogeneous Markov property, which states that the hidden Markov chain's state at any given time t depends solely on its preceding state at time $t-1$, independent of other states and observations at different times, as well as independent of the specific time t:

$$P(i_t|i_{t-1}, o_{t-1}, \cdots, i_1, o_1) = P(i_t|i_{t-1})$$

Observation independence property, which states that at any given time, the observation is solely dependent on the state of the Markov chain at that particular time, irrespective of other observations and states:

$$P(o_t|i_T, o_T, i_{T-1}, O_{T-1}, \cdots, i_{t+1}, o_{t+1}, i_t, i_{t-1}, o_{t-1}, \cdots, i_1, o_1) = P(o_t|i_t)$$

And the HMM encompasses three distinct challenges that need to be addressed: probability computation, learning, and prediction. In my report, considering its relevance to information theory, I will focus on prediction problems.

2.1 Prediction problems

The prediction problem, also referred to as decoding problem, entails finding the most likely state sequence $I = (i_1, i_2, \cdots, i_T)$ that maximizes the conditional probability $P(I/O)$ for a given observed sequence $O = (o_1, o_2, \cdots, o_T)$, where the model parameters $\lambda = (A, B, \pi)$ are known. In other words, the objective is to determine the state sequence that is most probable, given the observed sequence.

And equally inspired by the algorithms course this semester, I will try to solve the prediction problem by Brute force inspired by approximation algorithm and dynamic programming which is also specifically called Viterbi algorithm [3].

Example 2 [3]. Considering the box-and-ball model $\lambda = (A, B, \pi)$, sets of states $Q = \{1,2,3\}$, and sets of observations $V = \{red, white\}$.

$$A = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}, B = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{bmatrix}, \pi = \begin{bmatrix} 0.2 \\ 0.4 \\ 0.4 \end{bmatrix}$$

And given an observed sequence $O = (red, white, red)$, the goal is to determine the optimal sequence of states, also known as the optimal path $I^* = (i_1{}^*, i_2{}^*, i_3{}^*)$.

2.1.1 Brute Force

Firstly, about approximation algorithm, due to the involvement of forward and backward probabilities in probability computation problem, I have only provided a rough idea, with specific solutions and code left for future research. The idea

behind the approximate algorithm is to select, at each time t, choose the state ${i_t}^*$ that is most likely to occur at that moment, thus getting a sequence of states $I^* = ({i_1}^*, {i_2}^*, \cdots, {i_T}^*)$ that serves as the predicted outcome. And inspired by this, I have devised a brute-force approach for exhaustive enumeration. That is, to identify all possible sequences of states and select the group with the highest probability. What we only need to do is enumerating all possible state sequences and selecting the one that maximizes the conditional probability given the observed sequence.

2.1.2 Dynamic Programming

In this algorithm, the state sequence is represented as a path, and the problem is transformed into finding the maximum probability path using dynamic programming which is also specifically called Viterbi algorithm.

According to the principle of dynamic programming, the optimal path possesses the following property: if the optimal path passes through node ${i_t}^*$ at time t, then the sub-path from ${i_t}^*$ to the end node ${i_T}^*$ must be the optimal sub-path among all possible sub-paths from ${i_t}^*$ to ${i_T}^*$. Otherwise, there would exist a better sub-path from ${i_t}^*$ to ${i_T}^*$, which contradicts the assumption that the original path is optimal. Based on this principle, we can recursively compute the maximum probability of each sub-path from time $t = 1$ to $t = T$ where the state is i, until we obtain the maximum probability of each path at time $t = T$ where the state is i. The maximum probability at time $t = T$ corresponds to the probability of the optimal path P^*, and the end node ${i_T}^*$ of the optimal path can be determined simultaneously. To find all the nodes on the optimal path, we can trace back from the end node ${i_T}^*$ to the node ${i_{T-1}}^*, {i_{T-2}}^*, \cdots, {i_1}^*$, and obtain the optimal path $I^* = ({i_1}^*, {i_2}^*, \cdots, {i_T}^*)$[3].

The maximum probability among all different paths $(i_1, i_2, \cdots, i_t)$ with state i at time t is defined as:

$$\delta_t(i) = \max_{i_1, i_2, \cdots, i_{t-1}} P(i_t = i, i_{t-1}, \cdots, i_1, o_t, \cdots, o_1|\lambda), i = 1,2, \cdots N$$

And we can get the recursive formula of δ according to definition:

$$\delta_{t+1}(i) = \max_{i_1, i_2, \cdots, i_t} P(i_{t+1} = i, i_t, \cdots, i_1, o_{t+1}, \cdots, o_1|\lambda)$$

$$= \max_{1 \le j \le N} [\delta_t(j) a_{ji}] b_i(o_{t+1}), i = 1,2, \cdots, N, t = 1,2, \cdots, T-1$$

The $(t-1)_{th}$ node of the path with the maximum probability among all different paths $(i_1, i_2, \cdots, i_{t-1}, i)$ with state i at time t is defined as:

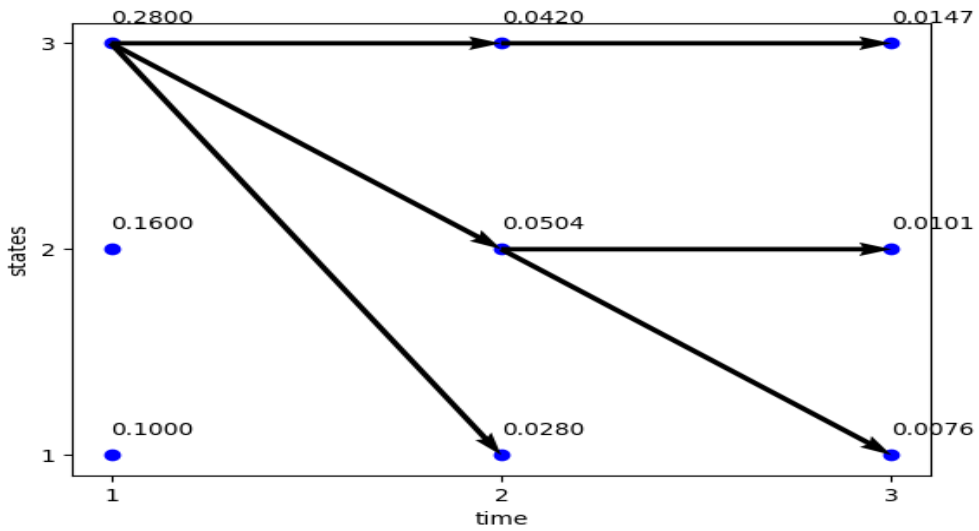
$$\Psi_t(i) = \arg\max_{1\le j\le N}[\delta_{t-1}(j)a_{ji}], i = 1,2,\cdots,N$$

Algorithm 1 Dynamic Programming

Input: model $\lambda = (A, B, \pi)$ and observation $O = (o_1.o_2, \cdots, 0_T)$
Output: the optimal path $I^* = (i_1{}^*, i_2{}^*, \cdots, i_T{}^*)$

Initialization
$\delta_1(i) = \pi_i b_i(o_1),\ i = 1, 2, \cdots, N$
$\Psi_1(i) = 0,\ i = 1, 2, \cdots, N$
Recursion, for $t = 2, 3, \cdots, T$
$\delta_t(i) = \max\limits_{1\le j\le N}[\delta_{t-1}(j)a_{ji}]b_i()o_t,\ i = 1, 2, \cdots, N$
$\Psi_t(i) = \max\limits_{1\le j\le N}[\delta_{t-1}(j)a_{ji}],\ i = 1, 2, \cdots, N$
Stop
$P^* = \max\limits_{1\le i\le N}\delta_T(i)$
$i_T{}^* = \underset{1\le i\le N}{argmax}[\delta_T(i)]$
Tracing back the optimal path, for $t = T-1.T-2.\cdots, 1$
$i_t{}^* = \Psi_{t+1}(i_{t+1}{}^*)$
return $I^* = (i_1{}^*, i_2{}^*, \cdots, i_T{}^*)$

For the example2, applying above algorithm for solving and the path of state diagram and the two-dimensional timeline are illustrated below:



And x axis label is time, y axis label is state, the arrow represents the state transition. The number above every vertex represents $\delta_t(i)$. According to the DP process, we can get the optimal path $I^* = (i_1{}^*, i_2{}^*, \cdots, i_T{}^*) = (3,3,3)$.

3 Future Work

The property of HMM and its abundant mathematical structures can be applied widely in speech recognition, machine translation, Chinese word segmentation, named entity recognition, part-of-speech tagging, and gene recognition. Looking forward to encountering a broader range of relevant knowledge in my future studies and research.

References

[1] LEONARD E. Baum, J. A. EAGON (1966) *An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology*.

[2] RABINER L, JUANG B (1986). *An introduction to hidden Markov Models*[J]. IEEE ASSP Magazine.

[3] Hang Li. (2021) *Machine Learning: An Algorithmic Perspective*. Tsinghua University Press.

[4] Zhihua Zhou. (2015) *Machine Learning*. Tsinghua University Press.