Chapter 1

Manish Panda(6614056) Contribution : Computing Likelihoods and Accumulating Parameters

1.1 Intorduction to HMM

Strong statistical tools used in academic research to explain observational sequences are Hidden Markov Models (HMMs). This is especially useful when the underlying system is assumed to follow a Markov process with non-observable states. Hidden Markov Models (HMMs) are a class of latent states that capture the system's intrinsic, imperceptible dynamics. All of these states add up to a Markov chain, which means that the likelihood of changing from one state to another depends only on the state at hand. Every hidden state is associated with observable outputs or emissions; these emissions do not explicitly reveal the underlying state, which introduces the model's covert character.

Transition probabilities, which are represented as a matrix and indicate the probability of changing from one state to another, control the transitions between concealed states. The likelihood of witnessing a certain output based on the present concealed state is determined by emission probabilities. Usually, a matrix or distribution is used to represent these probabilities, which are associated with every state.

1.2 Background

As a crucial probabilistic framework required for the representation of sequential data, Hidden Markov Models (HMMs) are important. Three essential elements serve as the modeling process' compass inside the HMM paradigm.

1. States

Unique Configurations: States in the context of the Hidden Markov Model (HMM) are discrete arrangements or circumstances that are intrinsic in a system. Even though they are not seen directly, these configurations have an impact on the observable results.

Latent Variables: Every state function as a latent variable that contains the unobservable variables that control the behavior of the system. Including these latent variables is essential to understanding the tremendous variability and complexity found in sequential data.

2. Transition

Probabilistic Associations: In HMMs, transitions create probabilistic connections between states, forming a Markov chain. With regard to uncertainties and fluctuations that are inherent in the system's evolution, the transition probabilities outline the probability of changing from one state to another.

Markov Property: The basic Markov property states that the next state depends only on the current state, which makes modeling dynamic processes easier. This feature simplifies the way that sequential dependencies are represented, making it possible to make predictions based on the current state without having to know the system's whole history trajectory.

3. Emissions

Observable Outputs: The observable manifestations linked to every state in the model are the quantifiable and palpable results, sometimes called emissions in the context of Hidden Markov Models (HMMs). Observable data or outputs are produced by a system when it transitions between several states over time. Any tangible data that represents the system's current state could be one of these outputs, as could a variety of measurements or observations. The important thing to remember about emissions in HMMs is that, although they offer clear, quantifiable information about the system, they never expose the hidden or underlying state of the system at any given time. As a result, modeling becomes more complex because the objective is frequently to infer the sequence of hidden states from the observable outputs.

Concealment of Underlying State: One important feature of HMMs is that the underlying state is hidden. Emissions do not clearly reveal the current concealed condition, even if they are directly observable. This adds another level of complexity, since the hard part is figuring out the latent dynamics based on the visible outputs, which makes the inference process more difficult.

1.3 Markov Chain and Transition Probabilities

1.3.1 Markov Property

An essential idea in the study of stochastic processes, the Markov property is helpful in many scientific domains as well as statistics, economics, and game theory. The idea behind the Markov property is that a process's behavior in the future is solely determined by its current state, not by the circumstances surrounding its arrival at that state. A stochastic process X_t (where t represents discrete time steps) exhibits the Markov property if, for any sequence of states and any time t, the conditional probability of moving to the next state X_{t+1} depends only on the present state X_t and not on the sequence of events or states that preceded it. Mathematically, this is expressed as:

$$P(X_{t+1} = x \mid X_t = x_t, X_{t-1} = x_{t-1}, \ldots, X_0 = x_0) = P(X_{t+1} = x \mid X_t = x_t) \tag{1.1}$$

This equation articulates that the probability of transitioning to a future state X_{t+1} is independent of all past states $(X_{t-1}, X_{t-2}, \ldots, X_0)$ given the current state X_t.

Features :

1. Because the stochastic process is memoryless, previous states only affect the current state and have no bearing on future ones. Because of this property, Markov processes are very easy to model and analyze because the dependencies' complexities are reduced.

2. In Markov processes, the transition between states is often described using either a transition matrix in the case of a finite state space or, in a more general setting, transition probabilities. This ensemble of probabilities, or matrix, captures the basic dynamics that are part of the process.

3. Analysis of Markov processes includes the application of techniques like state-transition diagrams, behavior analysis at steady states, and convergence property research.

1.3.2 Transition Probabilities in HMMs

One fundamental feature of Hidden Markov Models (HMMs) is the probabilistic depiction of transitions between hidden states. These transition probabilities are essential elements in the HMM framework that are controlled by the Markov property. Formally, if we denote the concealed states as S_i and S_j, the likelihood of transitioning from state S_i to S_j is expressed by the transition probability a_{ij}. In mathematical terms, this can be expressed as follows:

$$a_{ij} = P(S_{t+1} = S_j \mid S_t = S_i) \tag{1.2}$$

Here, a_{ij} represents the probability of the system transitioning to state S_j at time $t+1$ given that it is in state S_i at time t. This formulation adheres strictly to the Markov property, as the future state S_{t+1} depends solely on the present state S_t and not on any previous states.

Transition Matrix

The collection of transition probabilities together form a crucial component that is referred to as the transition matrix A, which is used to describe the dynamics of the Hidden Markov Model (HMM). This matrix, represented by A, and is made up of the probabilities a_{ij} that represent each and every possible state transition. Typically square in form, the size of this matrix depends on the number of hidden states that the model contains. The following requirements are met by the matrix elements:

1. Non-negativity: Each element a_{ij} is non-negative, i.e., $a_{ij} \geq 0$.

2. Normalization: The sum of the probabilities of transitioning from a given state to all possible states (including itself) equals 1, i.e., $\sum_j a_{ij} = 1$ for all i.

```
Transition Matrix (A):
0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.800000 0.200000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.800000 0.200000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.800000 0.200000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.800000 0.200000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.800000 0.200000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.800000 0.200000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.800000 0.200000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.800000 0.200000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
```

Figure 1.1: Transition Matrix.

1.4 Forward Likelihood

The Forward Algorithm is used to determine the probability of an observed series of symbols (emissions) up to a given time point while accounting for the hidden states in the Hidden Markov Model (HMM). Since there are exponentially many possible state sequences, direct calculation of these probabilities is computationally impractical. Instead, the basic idea is to use dynamic programming techniques to determine these probabilities.

1.4.1 Implementation of forward algorithm

1. Initialization

The algorithm starts by setting the initial forward probabilities for each state at the first time step. Given a state i, the initial forward probability for that state is equal to the product of the initial probability of that state and the likelihood of seeing the sequence's beginning symbol.

Mathematically, this is represented as:

$$\alpha_1(i) = \pi_i \cdot b_i(O_1) \tag{1.3}$$

Here, $\alpha_1(i)$ is the forward probability for state i at time 1, π_i is the initial state probability for state i, and $b_i(O_1)$ is the emission probability of observing the first symbol in the sequence given state i.

2. Recursion This is an important phase because it calculates the probability of every condition at a given time by iteratively taking into account all of the observations made up to that point in time.

The recursion formula in the Forward Algorithm is:

$$\alpha_t(i) = \left[\sum_{j=1}^{N} \alpha_{t-1}(j) \cdot a_{ji}\right] \cdot b_i(O_t) \tag{1.4}$$

where:

- $\alpha_t(i)$: This is the forward variable for state S_i at time t. It represents the probability of ending up in state S_i at time t after observing the sequence $O_1, O_2, \ldots, O_t$.
- The summation $\sum_{j=1}^{N} \alpha_{t-1}(j) \cdot a_{ji}$: This is the core of the recursion. It calculates the probability of transitioning into state S_i at time t from any possible state at time $t-1$.
- $\alpha_{t-1}(j)$: This is the probability of being in state S_j at time $t-1$ after observing $O_1, O_2, \ldots, O_{t-1}$.
- a_{ji}: This is the transition probability from state S_j to state S_i.
- By summing over all states j, we accumulate the total probability of arriving at state S_i from any state at the previous time step.
- $b_i(O_t)$: This is the emission probability. It gives the probability of observing the current observation O_t from state S_i. This factor incorporates the influence of the latest observation on the state probabilities.

3. Finalise The final step of the Forward Algorithm is given by:

$$P(O \mid \text{Model}) = \sum_{i=1}^{N} \alpha_T(i) \tag{1.5}$$

where:

- $P(O \mid \text{Model})$: This is the probability of the observed sequence O given the model.
- The summation $\sum_{i=1}^{N} \alpha_T(i)$: This represents the sum of the forward variables at the final time step T. It gives the total probability of the observation sequence given the model.

1.5 Backward Likelihood

The Backward Algorithm is a fundamental technique used in the field of Hidden Markov Models (HMMs), which is fundamentally similar to the Forward Algorithm. While the Forward Algorithm calculates the probability of an observation sequence's first part up to a given point, the Backward Algorithm calculates the probability of the observation sequence's last part, starting from a predetermined point in the model and going all the way to the end of the sequence. This algorithm is essential to the computation of posterior probabilities and is especially useful inside the Baum-Welch Algorithm framework, which is used to train HMMs.

1.5.1 Implementation of backward algorithm

1. Initialization At the final time step T, for each state S_i:

$$\beta_T(i) = 1 \tag{1.6}$$

This initialization asserts that the probability of ending the series in any state at the end is unity, indicating that the sequence has been completed by this point.

2. Recursion For each time step $t = T-1, T-2, \ldots, 1$ and for each state S_i:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \cdot b_j(O_{t+1}) \cdot \beta_{t+1}(j) \tag{1.7}$$

This step computes the probability of observing the sequence from time $t+1$ to T, given that we are in state S_i at time t. It aggregates over all possible states at time $t+1$, taking into account the transition probability a_{ij}, the emission probability $b_j(O_{t+1})$, and the backward probability $\beta_{t+1}(j)$.

3. Finalise To find the probability of the observed sequence O, we use:

$$P(O \mid \text{Model}) = \sum_{i=1}^{N} \pi_i \cdot b_i(O_1) \cdot \beta_1(i) \tag{1.8}$$

The backward probabilities, the likelihood of the original observation, and the probabilities associated with the starting states are all integrated in this formulation.

1.6 Occupational Likelihood

Hidden Markov Models (HMMs) use occupation likelihoods to express the chances of occupying a particular state at a given moment, conditional on the complete observation sequence. For HMM parameter estimation, these probabilities play a crucial role in the Baum-Welch algorithm, a variation of the Expectation-Maximization process.

The recalibration of the model's parameters, such as state transition and emission probabilities, emphasizes how important they are. The model can be improved to more closely mimic the observed sequences by quantifying the likelihood of each state at each temporal instance, dependent on the observed data.

The occupation likelihood for state i at time t is computed as:

$$\gamma_t(i) = \frac{\alpha_t(i) \cdot \beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j) \cdot \beta_t(j)} \tag{1.9}$$

This formula effectively combines the forward and backward probabilities, normalized by the sum of these products for all states, to ensure that the probabilities sum to 1.

1.7 Transitional Likelihood

The probabilities of changing from one state to another are known as transition likelihoods in Hidden Markov Models (HMMs), and they are essential for clarifying the dynamics of state changes within the model and showing the system's temporal evolution.

These likelihoods are crucial for updating the model's transition probabilities during HMM training. In the context of the entire observation sequence, they provide a quantitative measure of the frequency of particular state transitions.

The likelihood of transitioning from state i to state j at time t is given by:

$$\xi_t(i,j) = \frac{\alpha_t(i) \cdot a_{ij} \cdot b_j(O_{t+1}) \cdot \beta_{t+1}(j)}{P(O \mid \text{Model})} \tag{1.10}$$

Where:

- $\alpha_t(i)$ is the forward probability for state i at time t.
- a_{ij} is the transition probability from state i to state j.
- $b_j(O_{t+1})$ is the emission probability of observing O_{t+1} from state j.
- $\beta_{t+1}(j)$ is the backward probability for state j at time $t+1$.
- $P(O \mid \text{Model})$ is the probability of the observation sequence given the model, which can be computed using the forward or backward algorithm.

1.8 Forward and Backward Procedure Example

Example : Rainy Day Detection [1]

1.8.1 Forward procedure

States

$S = \{S\text{sunny}, S\text{rainy}\}$ (Hidden States)

Observables

$O = \{Oclean, Obike, Oshop, Opaint\}$ (Observables)

Initial Probabilities

$\pi = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}$ (Initial Probabilities)

Transition Probabilities

$A = \begin{bmatrix} 0.4 & 0.3 & 0.8 & 0.2 \\ 0.4 & 0.6 & 0.4 & 0.6 \end{bmatrix}$ (Transition Probabilities)

Emission Probabilities

$B = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.45 & 0.2 & 0.05 \end{bmatrix}$ (Emission Probabilities)

$\alpha_1(1) = 0.24$

$\alpha_1(2) = 0.12$

$\alpha_2(1) = 0.24 * 0.8 * 0.1 = 0.0192$

$\alpha_2(2) = 0.12 * 0.4 * 0.1 = 0.0048$

$\alpha = 0.0192 + 0.0048 = 0.024$

Finalise

$P(O \mid \lambda) = 0.0028512 + 0.0003048 = 0.003156$

1.8.2 Backward procedure

sunny state

$\beta_2(1) = 0.8 * 0.3 * 1 = 0.24$

$\beta_2(2) = 0.2 * 0.05 * 1 = 0.01$

$\beta = 0.24 + 0.01 = 0.25$

Rainy state

$\beta_1(1) = 0.4 * 0.3 * 1 = 0.12$

$\beta_1(2) = 0.6 * 0.05 * 1 = 0.03$

$\beta = 0.12 + 0.03 = 0.15$

Finalise

$P(O \mid \lambda) = (0.6 * 0.4 * 0.0071) + (0.4 * 0.3 * 0.0121) = 0.003156$

1.9 Code snippets and Results

1.9.1 Forward likelihood

```
function [forward_likelihood, scaling_factors] = forward_algorithm(observation_sequence, initial_prob, transition_prob, emission_prob)
    num_states = size(transition_prob, 1);
    seq_length = length(observation_sequence);

    forward_likelihood = zeros(num_states, seq_length);
    scaling_factors = zeros(1, seq_length);

    % Initialization step
    forward_likelihood(:, 1) = initial_prob .* emission_prob(:, observation_sequence(1));
    scaling_factors(1) = 1 / sum(forward_likelihood(:, 1));
    forward_likelihood(:, 1) = forward_likelihood(:, 1) * scaling_factors(1);

    % Recursion step
    for t = 2:seq_length
        for j = 1:num_states
            sum_prob = sum(forward_likelihood(:, t - 1) .* transition_prob(:, j));
            forward_likelihood(j, t) = sum_prob * emission_prob(j, observation_sequence(t));
        end
        scaling_factors(t) = 1 / sum(forward_likelihood(:, t));
        forward_likelihood(:, t) = forward_likelihood(:, t) * scaling_factors(t);
    end
end
```

Figure 1.2: Function for computing forward likelihood.

1.9.2 Backward likelihood

```
function backward_likelihood = backward_algorithm(observation_sequence, transition_prob, emission_prob, scaling_factors)
    num_states = size(transition_prob, 1);
    seq_length = length(observation_sequence);

    backward_likelihood = zeros(num_states, seq_length);
    backward_likelihood(:, seq_length) = 1 * scaling_factors(seq_length);

    % Recursion step
    for t = seq_length - 1:-1:1
        for i = 1:num_states
            sum_prob = sum(transition_prob(i, :) .* emission_prob(:, observation_sequence(t + 1))' .* backward_likelihood(:, t + 1)');
            backward_likelihood(i, t) = sum_prob * scaling_factors(t);
        end
    end
end
```

Figure 1.3: Function for computing backward likelihood.

1.9.3 Forward and Backward likelihood computational results

```
Iteration 1
Sequence 1 - Total Probability (Forward): 1.000000
Sequence 1 - Total Probability (Backward): 1.000000
The forward and backward probabilities are consistent for this sequence.
Sequence 2 - Total Probability (Forward): 1.000000
Sequence 2 - Total Probability (Backward): 1.000000
The forward and backward probabilities are consistent for this sequence.
Sequence 3 - Total Probability (Forward): 1.000000
Sequence 3 - Total Probability (Backward): 1.000000
The forward and backward probabilities are consistent for this sequence.
Sequence 4 - Total Probability (Forward): 1.000000
Sequence 4 - Total Probability (Backward): 1.000000
The forward and backward probabilities are consistent for this sequence.
Sequence 5 - Total Probability (Forward): 1.000000
Sequence 5 - Total Probability (Backward): 1.000000
The forward and backward probabilities are consistent for this sequence.
Sequence 6 - Total Probability (Forward): 1.000000
Sequence 6 - Total Probability (Backward): 1.000000
The forward and backward probabilities are consistent for this sequence.
Sequence 7 - Total Probability (Forward): 1.000000
Sequence 7 - Total Probability (Backward): 1.000000
The forward and backward probabilities are consistent for this sequence.
Sequence 8 - Total Probability (Forward): 1.000000
```

Figure 1.4: Forward and Backward computational result.

1.9.4 Occupational and Transition likelihood

```
function [occupation_likelihood, transition_likelihood] = compute_likelihoods(observation_sequence, forward_likelihood, backward_likelihood, transition_prob, emission_prob)
    num_states = size(transition_prob, 1);
    seq_length = length(observation_sequence);

    % Initialize matrices for occupation and transition likelihoods
    occupation_likelihood = zeros(num_states, seq_length);
    transition_likelihood = zeros(num_states, num_states, seq_length - 1);

    % Calculate occupation likelihoods
    for t = 1:seq_length
        occupation_likelihood(:, t) = (forward_likelihood(:, t) .* backward_likelihood(:, t)) / sum(forward_likelihood(:, t) .* backward_likelihood(:, t));
    end

    % Calculate transition likelihoods
    for t = 1:(seq_length - 1)
        for i = 1:num_states
            for j = 1:num_states
                transition_likelihood(i, j, t) = (forward_likelihood(i, t) * transition_prob(i, j) * emission_prob(j, observation_sequence(t + 1)) * backward_likelihood(j, t + 1)) / sum(forward_likelihood(:, t) .* backward_likelihood(:, t));
            end
        end
    end
end
```

Figure 1.5: Function for computing occupational and transition likelihood.

1.9.5 Occupational and Transition likelihood computational results

occupation_likelihood | total_occupation_likelihood

8x100 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.8717	0.5051	0.5580	0.4554	0.4851	0.4795	0.5020	0.4904	0.4235	0.5239	0.4160	0.5015	0.5359	0.4902	0.
2	1.0925	1.3889	1.0230	0.8726	0.9542	1.1052	1.0395	0.9206	1.0784	1.0998	1.0035	0.9487	1.1094	1.1213	0.
3	0.8206	1.6475	1.3901	1.9830	1.4071	1.8387	1.3330	2.1199	1.5037	1.5531	1.4295	1.4030	1.7135	1.4227	1.
4	2.0214	1.0152	0.5979	0.7176	1.1501	1.0064	0.7351	0.8705	1.0823	0.8132	0.8903	0.8278	1.1260	0.8391	0.
5	1.3229	1.7535	2.6405	1.2363	1.6584	1.5708	2.0680	1.3860	1.9460	2.3197	1.9524	1.5835	1.6383	2.2460	1.
6	2.0106	1.5347	1.4396	2.1793	1.5437	1.6063	1.4928	1.8240	1.4057	1.1366	1.9784	1.5911	1.4768	1.1836	1.
7	1.2833	0.8863	0.6626	1.0329	1.1386	1.2152	0.6976	1.1458	0.7854	0.8358	0.7638	0.8209	1.6039	0.8714	0.
8	0.5769	1.2687	1.6883	1.5229	1.6628	1.1779	2.1319	1.2427	1.7751	1.7178	1.5661	2.3235	0.7962	1.8257	1.
9															
10															
11															

Figure 1.6: Occupational likelihoods.

transition_prob

8x8 double

	1	2	3	4	5	6	7	8	9
1	0.3668	0.8528	0.5040	0.0706	0.1845	0.8067	0.3814	0.0500	
2	0.5155	0.6131	0.4462	0.2645	0.0290	0.2229	0.7525	0.9359	
3	0.0220	0.4627	0.3321	0.8016	0.1350	0.1635	0.3374	0.7285	
4	0.4785	0.1028	0.6646	0.9533	0.4893	0.2424	0.5800	0.7303	
5	0.7902	0.7491	0.6378	0.0122	0.7966	0.3464	0.3197	0.7249	
6	0.9935	0.4933	0.9244	0.1121	0.1745	0.9088	0.0189	0.4938	
7	0.7174	0.4020	0.4113	0.9971	0.7083	0.0319	0.6057	0.8718	
8	0.6551	0.0016	0.3444	0.0355	0.6289	0.7884	0.4591	0.3578	

Figure 1.7: Transition probabilities.

1.10 Conclusion

The forward and backward likelihoods in Hidden Markov Models (HMMs) offer a comprehensive probabilistic analysis of sequential data from different time-based viewpoints. While backward likelihoods focus on the probability from a particular moment onward to the end of the sequence, forward likelihoods explain the likelihood of a sequence's occurrence up to a particular temporal point. An important connection between these temporal evaluations is occupation likelihoods, which relate them to the probability that the system will occupy a specific state at a given time. Combining the forward and backward perspectives is necessary to have a thorough grasp of state occupancy throughout the sequence.

Furthermore, transition likelihoods provide a deep understanding of the dynamics of the model's state transitions. By calculating the probability of changing states, they shed light on the random mechanisms guiding the sequence's development. The foundation of HMM analysis is this comprehensive framework, which incorporates occupation, transition, forward, and backward probabilities. It makes it easier to interpret the data's temporal development and state behavior in a sophisticated way, which improves our comprehension of the dynamics behind the model.

References

[1] Ayra Lux. (2016) *Hidden Markov Models Part 1: The Likelihood Problem.* Retrieved from https://medium.com/@Ayra_Lux/hidden-markov-models-part-1-the-likelihood-problem-8dd1066a784e