A Hidden Markov Model (HMM) is a statistical model that represents systems with underlying unobservable (hidden) states. HMMs are particularly used for temporal data where the state of the system evolves over time according to some probabilistic rules. Here's an explanation of HMM with the appropriate mathematics:

Components of an HMM

1. **States**: The model consists of N states, which are hidden. These states can be denoted as $S = \{S_1, S_2, \ldots, S_N\}$.

2. **Observations**: The model generates a sequence of observations. Each observation comes from a set $V = \{v_1, v_2, \ldots, v_M\}$.

3. **Transition Probabilities**: The probability of transitioning from state i to state j is given by $A = \{a_{ij}\}$, where:

$$a_{ij} = P(S_{t+1} = j | S_t = i)$$

   and $\sum_{j=1}^{N} a_{ij} = 1$ for all i.

4. **Emission Probabilities**: The probability of observing v_k from state j is given by $B = \{b_j(k)\}$, where:

$$b_j(k) = P(O_t = v_k | S_t = j)$$

   and $\sum_{k=1}^{M} b_j(k) = 1$ for all j.

5. **Initial State Probabilities**: The probability of starting in state i is given by $\pi = \{\pi_i\}$, where:

$$\pi_i = P(S_1 = i)$$

   and $\sum_{i=1}^{N} \pi_i = 1$.

The Three Fundamental Problems of HMM

1. **Evaluation (Likelihood)**: Given an observation sequence $O = (O_1, O_2, \ldots, O_T)$ and the model $\lambda = (A, B, \pi)$, compute the probability of the observation sequence given the model, $P(O|\lambda)$.

   This is solved using the **Forward Algorithm**:

$$\alpha_t(j) = P(O_1, O_2, \ldots, O_t, S_t = j | \lambda)$$

   with the recursive relation:

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j(O_t)$$

where the initialization is:

$$\alpha_1(j) = \pi_j b_j(O_1)$$

and the final probability is:

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_T(j)$$

2. **Decoding (Most Likely Sequence)**: Given the observation sequence O and the model λ, find the most likely sequence of states that produced the observations.

   This is solved using the **Viterbi Algorithm**:

   $$\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] b_j(O_t)$$

   with initialization:

   $$\delta_1(j) = \pi_j b_j(O_1)$$

   and backtracking to find the state sequence.

3. **Learning (Training)**: Given the observation sequence O and the model structure, adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$.

   This is solved using the **Baum-Welch Algorithm**, an Expectation-Maximization (EM) algorithm:

   - **Expectation Step**: Compute the forward ($\alpha_t(i)$) and backward ($\beta_t(i)$) variables.

     $$\beta_t(i) = P(O_{t+1}, O_{t+2}, \ldots, O_T | S_t = i, \lambda)$$

     with the recursive relation:

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

   - **Maximization Step**: Re-estimate the model parameters A, B, π using the α and β variables.

Summary

An HMM is defined by its states, observations, transition probabilities, emission probabilities, and initial state probabilities. The three main tasks are to evaluate the likelihood of an observation sequence given the model, to find the most probable sequence of hidden states, and to adjust the model parameters to best fit the observed data. Each task has specific algorithms: Forward Algorithm for evaluation, Viterbi Algorithm for decoding, and Baum-Welch Algorithm for learning.