Hidden Markov Model Basics

Patrick Gampp, 9931027

Seminar: Advanced Signal Processing, SS 2008
Supervisor: Dr. Michael Pucher

Abstract

This document wants to give a basic introduction to Hidden Markov Models (HMMs) regarding the field of speech communication and speech synthesis, especially.

1 Markov Chain

A Markov chain like depicted in Fig. 1 has a set of N distinct states $S_1, S_2, \ldots, S_N$, at regularly spaced discrete times, the system changes its state. The time instants are denoted with $t = 1, 2, \ldots$. The actual state at time instance t is denoted as q_t. An important feature of this probabilistic model is the Markov property. Given the present state, the future and past states are independent.

$$P[q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \cdots] = P[q_t = S_j | q_{t-1} = S_i] \tag{1}$$

The state transition probability is defined as:

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i], \qquad 1 \leq i,\, j \leq N, \tag{2}$$

with the properties:

$$a_{ij} \geq 0 \tag{3}$$

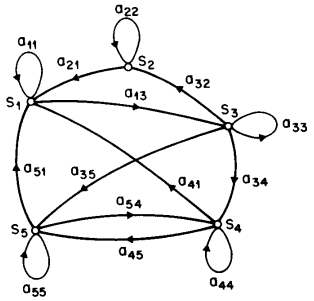

Figure 1: Markov chain with 5 states S_1 to S_5 and state transitions a_{xx}

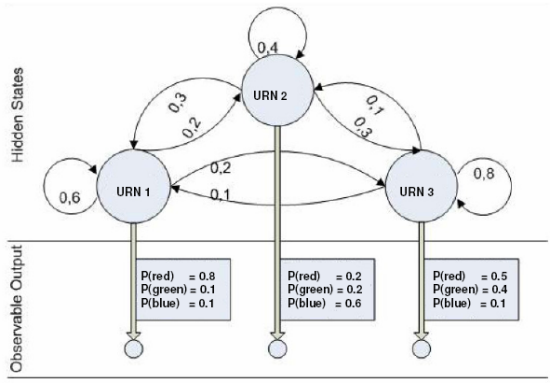

Figure 2: Hidden Markov Model: A doubly embedded stochastic process with an underlying hidden, not observable process, that procudes a sequence of observations.

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{4}$$

An Markov chain could also be called an observable Markov model since the output of the process corresponds with the observed states, which is a physical event.

2 Extension to Hidden Markov Model

The Markov models, where each state corresponds to an observable output is too restrictive to be used for many problems. In speech communication it is not so easy to do one-to-one mapping from speech to a word symbol. There are for instance different symbols that produce the same sound. Furthermore, there is a large variation in speech. Speech is always different for other speakers. Speech even varies for the same speaker speaking in different moods or environments, eg. where the person whispers or screams. As a further difficulty in speech, there are no explicit boundaries that can be detected. Speech waveform is not a concatenation of static patterns.

A hidden Markov model is a doubly embedded stochastic process, where the actual states producing the output are hidden. Additionally, there is a second set of stochastic processes, which produces the sequence of observations. This can be seen in Fig. 2.

3 Elements of an Hidden Markov Model

A HMM can completely described by the following elements.

1. N is the number of the hidden states in the model. Individual states are denoted as $S = \{S_1, S_2, \ldots, S_N\}$, and the state at time t as q_t.

2. M is the number of distinct observation symbols for each state and is called the alphabet size and correspond to the physical output of the system which is modelled.

3. The state transition probability distribution $A = \{a_{ij}\}$, as defined in Eq. (2).

4. The observation symbol probability distribution in state $j, B = \{b_j(k)\}$, where

$$b_j(k) = P[v_k \text{ at } t | q_t = S_j], \qquad 1 \leq j \leq N \quad \text{and} \quad 1 \leq k \leq M \tag{5}$$

5. The initial state distribution, describing the probability of beginning the state sequence in a certain initial state.

$$\pi_i = P[q_1 = S_i], \qquad 1 \leq i \leq N \tag{6}$$

6. The observation sequence is denoted as $O = O_1 O_2 \cdots O_T$.

λ represents the complete parameter set of a model, where $\lambda = (A, B, \pi)$.

4 The Three Basic Problems and their Solutions

In order to work with HMMs three basic problems have to be solved.

- Problem 1: Given the observation sequence O and model λ, how can the probability of a given model producing the output sequence O $P(O|\lambda)$ efficiently be computed? A solution is the Forward algorithm.

- Problem 2: Given the observation sequence O and the model λ, how can a inner state sequence $Q = q_1 q_2 \ldots q_T$ which best explains the obervations O? A solution is the Viterbi algorithm.

- Problem 3: Given the observation sequence O, how can the probability of a observation sequence being produced by a model λ, i.e. how to choose model parameters λ in order to maximize $P(O|\lambda)$?. A solution is the Baum–Welch algorithm.

4.1 Forward-Algorithm

A forward variable is the probability of a partial observation sequence until time t, given the model λ and is defined as:

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i | \lambda) \tag{7}$$

$\alpha_t(i)$ can be solved inductively:

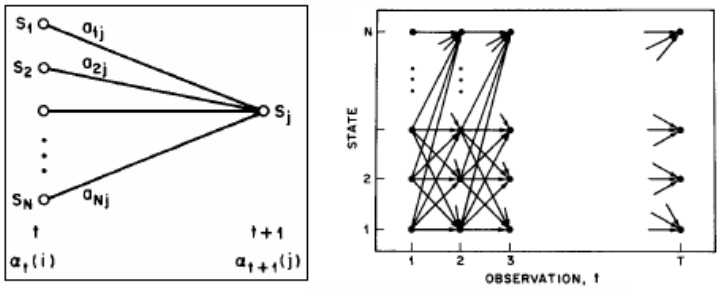

Figure 3: Left: Sequence of operations required for the computation of the forward variable. Right: Computation of forward variable including all states i and times t.

1. Initialization:
$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N. \tag{8}$$

2. Induction:
$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}) \qquad 1 \leq t \leq T-1 \text{ and } 1 \leq j \leq N \tag{9}$$

3. Termination:
$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{10}$$

$\alpha_t(i)$ is computed along t and for every state S as it can be seen in Fig. 3. When the final T is reached, the likelihoods over all states are summed up.

4.2 Viterbi-Algorithm

The highest probability along a single path at time t that ends in state S_i is defined by:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 \ldots q_t = i, O_1 O_2 \cdots O_t | \lambda] \tag{11}$$

By induction we have:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(O_{t+1}). \tag{12}$$

The basic idea of this algorithm is to retrieve the most likely state sequence for the given observation sequence O by backtracking the argument shown in Fig. 4, i.e. which maximized Eq. (12). The array where these values are stored is Ψ. The algorithm consists of the following steps:

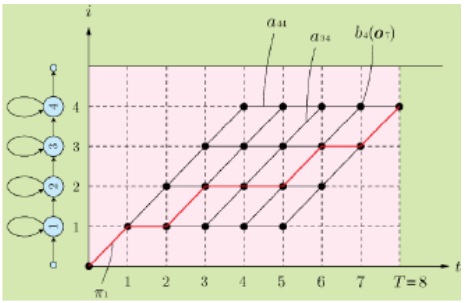

Figure 4: Backtracking of most likely path, indicated by the red line, with Viterbi algorithm.

1. Initialization:

$$\delta_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N \tag{13}$$

$$\psi_1(i) = 0 \tag{14}$$

2. Recursion:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij} b_j(O_t)], \qquad 2 \leq t \leq T \text{ and } 1 \leq j \leq N \tag{15}$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \qquad 2 \leq t \leq T \text{ and } 1 \leq j \leq N \tag{16}$$

3. Termination:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \tag{17}$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)] \tag{18}$$

4. Backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \ldots, 1. \tag{19}$$

4.3 Baum-Welch-Algorithm

The probability of being in state S_i at time t and state S_j at time $t+1$ with given model λ and observation sequence O is:

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \tag{20}$$

According to the forward variable, a backward variable $\beta_t(i)$ is defined:

$$\beta_t(i) = P(O_{t+1} O_{t+2} \cdots O_T | q_t = S_i, \lambda). \tag{21}$$

In terms of forward and backward variables, $xi_t(i,j)$ can be written as:

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \tag{22}$$

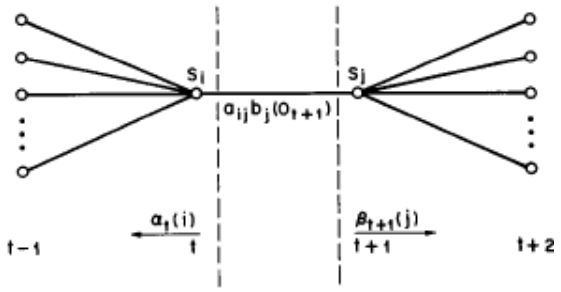

Figure 5: Illustration of the sequence of operations required for the computation of $\xi_t(i,j)$ in the Baum–Welch algorithm.

$\gamma_t(i)$ is the probability of being in state S_i at time t, given observation sequence and the model:

$$\gamma_t(i) = P(q_t = S_i|O,\gamma) \tag{23}$$

The relation between $\gamma_t(i)$ and $\xi_t(i,j)$ is:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{24}$$

Therefore, one can say that $\sum_{t=1}^{T-1} \gamma_t(i)$ is the number of expected number of transitions from S_i and $\sum_{t=1}^{T-1} \xi_t(i,j)$ is the expected number of transitions from S_i to S_j. The parameters of the HMM are iteratively reestimated by the following formulas:

$$\bar{\pi}_i = \gamma_1(i) \tag{25}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{26}$$

$$\bar{b}_j(k) = \frac{\sum_{t=1}^{T} \gamma_t(j) \quad s.t. O_t = v_k}{\sum_{t=1}^{T} \gamma_t(j)} \tag{27}$$

The iterations are profitable since Baum prooved, that $P(O|\bar{\lambda}) \geq P(O|\lambda)$, i.e. the model parameters are optimized with respect to the observation sequence till they reach a limiting point. The result is called maximum likelihood estimate of the HMM. The algorithm leads to a local maximum only, depending on the initialized model parameter values.

The reestimation formulas can directly be derived by maximizing Baum's auxiliary function:

$$Q(\lambda,\bar{\lambda}) = \sum_{Q} P(Q|O,\lambda) \log P(O,Q|\bar{\lambda}) \tag{28}$$

The Baum–Welch algorithm is similar to the EM algorithm of statistics, in which the expectation step corresponds to the calculation of the auxiliary function and the modification step to the reestimation of λ.

References

[1] Rabiner, L.: *A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of the IEEE, VOL.77, NO. 2, February 1989

[2] Cambridge University Engineering Department: *HTK Book*, 2006