Lecture 2: Hidden Markov Models

A hidden Markov model (HMM) consists of a set of internal states and a set of observable tokens. A run of a hidden Markov model generates a hidden state sequence s_1, …, s_T and a sequence of observable tokens a_1, …, a_T.

$$
\begin{aligned}
\pi(s) &= P(s_1 = s),\ \sum_s \pi(s) = 1 \\
T(w|s) &= P(s_{t+1} = w | s_t = s),\ \sum_w T(w|s) = 1 \\
O(a|s) &= P(a_t = a | s_t = s),\ \sum_a O(a|s) = 1 \\
P(s_1, \ldots, s_T, a_1, \ldots, a_T) &= \pi(s_1) \left(\Pi_{t=1}^{T} O(a_t|s_t)\right) \left(\Pi_{t=1}^{T-1} T(s_{t+1}|s_t)\right)
\end{aligned}
$$

Applications of HMMs:

- Speech Recognition. The hidden states are word positions and the observable tokens are accoustic feature vectors.

- Part of speech tagging. The hidden states are the parts of speech (noun, verb, adjective, and so on).

- DNA sequence analysis. The hidden states might be protien secondary structure or a position in a homologous sequence.

1 The Viterbi Algorithm

$$\begin{aligned}
\mathrm{Viterbi[s,t]} &= \max_{s_1,\ldots,s_{t-1}} P(s_1,\ldots,s_{t-1},s,a_1,\ldots,a_{t-1}) \\
\mathrm{Viterbi[s,1]} &= \pi(s) \\
\mathrm{Viterbi[w,t+1]} &= \max_{s}\ \mathrm{Viterbi[s,t]O(a_t|s)T(w|s)} \\
s_T^* &= \underset{s}{\mathrm{argmax}}\ \mathrm{Viterbi(s,T)O(a_T|s)}
\end{aligned}$$

The best predicessor s_t can be recorded for each possible value of s_{t+1} and the best path can be constructed by working backward from s_T^* through best predicessors.

2 The Forward-Backward Procedure

$$\begin{aligned}
\mathrm{Forward[s,t]} &= P(a_1,\ldots,a_{t-1},s_t=s) \\
\mathrm{Backward[s,t]} &= P(a_t,\ldots,a_T \mid s_t=s)
\end{aligned}$$

$$\begin{aligned}
\mathrm{Forward[s,1]} &= \pi(s) \\
\mathrm{Forward[w,t+1]} &= \sum_s \mathrm{forward[s,t]O(a_t|s)T(w|s)}
\end{aligned}$$

$$\begin{aligned}
\mathrm{Backward[s,T]} &= O(a_T|s) \\
\mathrm{Backward[s,t]} &= O(a_t|s)\sum_w T(w|s)Backward[s,t+1]
\end{aligned}$$

$$\begin{aligned}
P(a_1,\ldots,a_T) &= \sum_s \pi(s)\mathrm{Backward[s,1]} \\
P(s_t=s \mid a_1,\ldots,a_T) &= \frac{\mathrm{Forward(s,t)Backward[s,t]}}{P(a_1,\ldots,a_T)}
\end{aligned}$$

3 Trigram Language Models

Let $\#(w)$ be the number of times that the word w appears in a certain training corpus. Let $\#(w_1 w_1 2)$ be the number of times that the pair of words w_1‘w_2 occurs and similarly for $\#(w_1, w_2, w_3)$ for the triple of words w_1, w_2, w_3. Let N be the total number of word occurances. A interpolated trigram model predicts the word w_3 following a given pair w_1, w_2 as follows.

$$P(w_3|w_1, w_2) = \lambda_1 \left(\frac{\#(w_1, w_2, w_3)}{\#(w_1, w_2)} \right) + \lambda_2 \left(\frac{\#(, w_2, w_3)}{\#(w_2)} \right) + \lambda_3 \left(\frac{\#(w_3)}{N} \right) \quad (1)$$

Here λ_1, λ_2, and λ_3 are non-negative weights which som uto one:

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

A weighted sum, such as (**??**), where the weights are non-negative and sum to one, is called a *convex combination*. Any convex combination of probability distributions is also a probability distribution. A convex combination of distributions is often called an *interpolated* model. In a trigram language model the weights λ_1, λ_2 and λ_3 are usually taken to depend on some way on the pair w_1, w_2. This is ok since we can hold w_1, w_2 fixed in defining the conditional distribution $P(w_3|w_1, w_2)$.

A trigram language model defines the hiddens states used in standard HMM-based speech recognition. A hidden state is a triple of words w_1, w_2, w_3 together with a index for a position in the last word. For example “we the pe*ople” is a state specifing that the two preceeding words were “we” and “the” and that we are currently at the o in the word “people” so we should expect to be hearing an “o” sound. The state transition probabilities can be taken to be the following.

$$T(w_1, w_2, [\alpha_1 \ldots \alpha_i * \alpha_{i+1}\alpha_{i+2} \ldots \alpha_k] \mid w_1, w_2, [\alpha_1 \ldots \alpha_i * \alpha_{i+1}\alpha_{i+2} \ldots \alpha_k]) = 1/2$$

$$T(w_1, w_2, [\alpha_1 \ldots \alpha_i\alpha_{i+1} * \alpha_{i+2} \ldots \alpha_k] \mid w_1, w_2, [\alpha_1 \ldots \alpha_i * \alpha_{i+1} \ldots \alpha_k]) = 1/2$$

$$T(w_2, w_3, *w_4 \mid w_1, w_2, w_3*) = P(w_4|w_2, w_3)$$

The output probablities are determined by a "accoustic model" specifying the probability distribution over accoustic feature vectors given the current phoneme of the hidden state. Actually, the distribution on accoustic features is usually taken to depend on a "triphone" — the preceding phoneme, the current phoneme, and the next phoneme. Pentaphones are even used in some systems.

4 Problem

Suppose that we have two hidden states A and B and two observable symbols a and b and an HMM defined by the following probabilities.

$$\pi(A) = \pi(B) = 1/2$$

$$T(A|A) = T(B|B) = 1 - \epsilon$$

$$T(B|A) = T(A|B) = \epsilon$$

$$O(a|A) = O(b|B) = 1 - \delta$$

$$O(b|A) = O(a|B) = \delta$$

Now suppose that we observe a sequence of T a's. Let $F(A,t)$ abbreviate Forward(A, t) and similarly for $F(B,t)$ Give the values of $F(A,1)$ and $F(B,1)$ and give equation for computing $F(A,t{+}1)$ and $F(B,t{+}1)$ as a function of $F(A,t)$ and $F(B,t)$. Similarly let $B(A,t)$ abbreviate Backward(A, t). Give the values of $B(A,T)$ and $B(B,t)$ and give equation for computing $B(A,t)$ and $B(B,t)$ as a function of $F(A,t+1)$ and $F(B,t+1)$.

Extra Credit: Solve for $F(A,t)$, $B(A,t)$ and $P(a_1,\ldots,a_T)$. Graph $P(s_t = A)$ as a function of t for $\epsilon = \delta = 1/4$, and $T = 20$ (you should not calculate the actual numbers for $P(s_t = A)$ if you can see qualitatively what the graph must look like).