CMSC35000-1 Introduction to Artificial Intelligence **Winter 2007**

Lecture 16: 3/1/2007

Lecturer: Partha Niyogi *Scribe: X. Zhou and S. Reddy*

HMM and Speech

16.1 Speech

The process of sound production by human is:
brain ⇒ mouth/lung/etc ⇒ sequences of articulacy gestures ⇒ sequences of articulacy shapes ⇒ sequences of articulacy sound.

The configurations of the articulatory canal (from the larynx to the lips) determine the phoneme being produced by the speech signal. Hence, given a speech signal, we can identify the constituent phonemes if we have an accurate representation of the sequency of articulatory configurations. In other words, there is a set of hidden variables that represent the phoneme-sequence in the signal. This is the intuition behind using Hidden Markov Models for speech recognition.

One big problem in speech recognition is that the sounds for the same word by different people or even by the same person may have different lengths in time during different uteerances. It is thus not appropriate to deal with this problem in time space directly.

Hence, speech recognizers use Fourier transforms to convert the signal in time space into frequency space, and use the limited bandwidth to frame the signal. (Human ears can only recognize sound in a fixed frequency bandwidth: 20Hz-20000Hz).

Let the signal in time space be $x(t)$, and in frequency space be $X(\omega)$. Then we can use vector $\mathbf{F}$ in frequency space to represent $x(t)$, where the i^{th} element in vector $\mathbf{F}$ is

$$F_i = \int_{\omega_{i-1}}^{\omega_i} \mid X(\omega) \mid^2 d\omega \tag{16.1}$$

16.2 HMMs

Assuming X: sequences of frequency vectors $O_1, \cdots, O_T$, where $O_i \in \mathcal{R}^k$, Y:{-1,+1}. Speech recognition can be viewed as a classification problem from X to Y. Our aim is to find out $P(O_1, \cdots, O_T| -1)$ and $P(O_1, \cdots, O_T| +1)$.

What is an HMM?

A Hidden Markov Model is a finite state, first order Markov Chain. Assuming n state $\{1 \cdots n\}$, then we have a $n \times n$ transition matrix A, where $A_{ij} = P(j|i)$. Let S_t be the state at time t, O_t be the observation at time t, and $t = 1 \cdots T$. Let b_i be a probability distribution on observable space O, where $i = 1 \cdots n$. π_i is the starting probability at state i. So an HMM is determined by $\lambda = \{A, \mathbf{b}, \pi\}$.

Obs:

$$P(O_1,\cdots,O_T|\lambda)=\sum_{S_1\cdots S_T}P(O_1,\cdots,O_T,S_1,\cdots,S_T|\lambda) \tag{16.2}$$

There are a total n^T possible sequences of length T. The probability for each one can be computed by:

$$\pi(\prod_{i=1}^{T-1}b_i(O_i)A_{i,(i+1)})b_T(O_T) \tag{16.3}$$

Two important problems need to be solved:

1. how can we calculate $P(O_1,\cdots,O_T|\lambda)$ efficiently ?
2. how can we find out the optimal state-sequences?

For the first problem, we can use *Forward Algorithm.* Define $\alpha_t(i)=P(P(O_1,\cdots,O_t,S_t=i|\lambda)$. We omit λ in the following part. So $\alpha_t(i)=P(P(O_1,\cdots,O_t,S_t=i)$. Then we have the following results:

$$\begin{aligned}
P(O_1,\cdots,O_T) &= \textstyle\sum_{i=1}^n \alpha_T(i)\\
\alpha_1(i) &= P(O_1,S_1=i)=\pi_i b_i(O_i)\\
\alpha_{t+1}(i) &= P(O_1\cdots O_{t+1},S_{t+1}=i)\\
&= \textstyle\sum_{j=1}^n P(O_1\cdots O_{t+1},S_t=j,S_{t+1}=i)\\
&= \textstyle\sum_{j=1}^n P(O_1\cdots O_{t+1},S_t=j)P(S_{t+1}=i,O_{t+1}|O_1\cdots O_t,S_t=j)\\
&= \textstyle\sum_{j=1}^n \alpha_t(j)P(S_{t+1}=i|O_1\cdots O_t,S_t=j)P(O_{i+1}|S_{t+1}=i,O_1\cdots O_t,S_t=j)\\
&= \textstyle\sum_{j=1}^n \alpha_t(j)A_{ji}b_i(O_{t+1})
\end{aligned} \tag{16.4}$$

Therefore, we have the following equation

$$\alpha_{t+1}(i)=\sum_{j=1}^n \alpha_t(j)A_{ji}b_i(O_{t+1}) \tag{16.5}$$

The total computation needed is: Tn^2 additions and $2Tn^2$ multiplications. Therefore, it is a polynomial algorithm.

In practice, people use a small window to frame the signal in time space, then convert it into frequency vector using FFT.

From the above analysis we can see that the probability at time $t+1$ can be efficiently computed based on the result at time t. This implies a natural subproblem for the whole problem of computing the probability of a observed sequence. We can use this natural subproblem to design a dynamic programming algorithm to find the sequence of states that has the highest probability to generate the given sequences (more in the next lecture).