Notes on Automatic Speech Recognition

Automatic Speech Recognition

1 Feature Extraction

The speech features are the most important aspect in ASR. It saves us from the need to work directly on the speech samples. Of course the features, which is a function of the speech samples, can not contain more information about the linguistic content than the raw speech samples. However, it can be designed to be more appropriate for recognition purposes than the raw samples.

1.1 MFCC

MFCC stands for Mel-Frequency Cepstral Coefficients. The Mel-scale is defined by

$$Mel(f) = 2595 \log_{10}(1 + \frac{f}{700}). \tag{1}$$

MFCC feature extraction steps are:

1. pre-processing
2. windowing
3. FFT for spectrum
4. mel-scale filter binning
5. compression (taking logarithm)
6. inverse DFT or discrete cosine transform
7. cepstral truncation

1.2 Dynamic Features

MFCC is a set of the static features, since it only depends on a single processing window. From the static features time sequence, one can derive the

dynamic features. Specifically, the *delta* features estimates the rate of change of the static features. The basic definition is

$$\triangle F(t) = \lim_{dt \to 0} \frac{F(t+dt) - F(t)}{dt}. \tag{2}$$

This is typically approximated by

$$\triangle F(n) = \frac{\sum_{k=-w}^{w} kF(n+k)}{\sum_{k=-w}^{w} k^2}, \tag{3}$$

so this is simply computed from the static features. The *acceleration* features is defined as the dynamic features of the dynamic features, and they can be computed in the same way.

1.3 Parameters in Feature Extraction

From the above discussion, we know the basic idea of feature extraction. Exactly what is used as the speech data representation in the recognition module is determined by the feature kind *and* the following parameters that need to be considered and specified:

- window size
- frame rate
- feature dimension
- dynamic features

1.4 Phones, Phonemes, and Articulatory Features

Phoneme is the logical representation while phone is the physical realization. The same phoneme may be realized differently in different words, and leading to two different phones.

Articulatory features are used to describe how a phone is articulated. produced. For consonants, this is described by *the place of articulation*, the point of closest constriction in the oral cavity, such as

- bilabial
- labiodental

- interdental
- alveolar
- palatal-alveolar
- palatal
- velar
- labiovelar
- uvular
- glottal,

and *the manner of articulation*, referring to the amount of constriction in the articulation, such as

- stops
- fricatives
- affricates
- nasals
- liquids or glides.

For vowels, this is described by

- the frontness (front, central or back): where is the greatest constriction in the oval cavity
- the height (high or low): how far the lower jaw is from the upper
- the roundness (rounded or unrounded): whether the lips have been rounded.

2 Basic Acoustic Modeling

In building up ASR systems, we need to associate data with their corresponding linguistic classes. This is called *modeling the linguistic units*. According the the linguistic units, the common models are:

- phone models
- word models
- syllable models
- head/tail models

2.1 Hidden Markov Models

Once the base units are decided, we need to model the relationship between the classes and the speech features. The most frequently used models are the hidden Markov Models (HMMs).

An HMM consists of

1. a Markov chain, whose probability is given by

$$P(q_1, q_2, \ldots, q_N) = P(q_1) \prod_{n=2}^{N} P(q_n|q_{n-1}), \tag{4}$$

   where q_n is the state at time index n.

2. Emitting density functions, which are the state-conditional probability density of the observation, $f(x|q)$. The Gaussian models are often used for the emitting densities, with the means and variances depend on the state. I.e.,

$$f(x|q) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_q|^{\frac{1}{2}}} e^{\frac{-1}{2}(x-\mu_q)'\Sigma_q^{-1}(x-\mu_q)}, \tag{5}$$

   where d is the dimension of feature vector x.

Thus the joint probability of a state and feature sequence is given by

$$p(x_{1:N}, q_{1:N}) = P(q_1) f(x_1|q_1) \prod_{n=2}^{N} P(q_n|q_{n-1}) f(x_n|q_n). \tag{6}$$

It follows that the posterior probability of a state sequence, which is not observable, is

$$p(q_{1:N}|x_{1:N}) = \frac{p(x_{1:N}, q_{1:N})}{p(x_{1:N})} = \frac{p(x_{1:N}, q_{1:N})}{\sum_{q'_{1:N}} p(x_{1:N}, q'_{1:N})} \tag{7}$$

The parameters in HMMs with Gaussian emitting densities include the initial probability of the Markov chain (denoted here by π), the transition probability matrix (denoted by T), the means and variances in the state-dependent emitting densities (denoted by $\mu'_q s$ and $\Sigma'_q s$).

2.1.1 HMM Parameter Learning

Since the state is hidden, we cannot identify which state the Markov chain is in at any given time. Instead, we use the probablistic counts. For the initial state and the state transitions, we need to compute

$$P(Q_1 = i|x_{1:N}), \tag{8}$$

and

$$P(Q_n = i, Q_{n+1} = j|x_{1:N}) \text{ for all } n \text{ and } i, j. \tag{9}$$

For $\mu'_q s$ and $\Sigma'_q s$, we need to compute

$$P(Q_n = i|x_{1:N}) \text{ for all } n \text{ and } i. \tag{10}$$

The estimation of these parameters follows the idea given in the exercises. The maximum-likelihood estimators for π is simply (8). For T, it is

$$\begin{aligned} T_{i,j} = & \text{ transition probability from state } i \text{ to state } j \\ = & \frac{\sum_t P(Q_n = i, Q_{n+1} = j|x_{1:N})}{\sum_t P(Q_n = i|x_{1:N})} \end{aligned} \tag{11}$$

For the means and variances, it is

$$\begin{aligned} \mu_q &= \frac{\sum_n P(Q_n = q|x_{1:N}) x_n}{\sum_n P(Q_n = q|x_{1:N})} \\ \Sigma_q &= \frac{\sum_n P(Q_n = q|x_{1:N})(x_n - \mu_q)(x_n - \mu_q)'}{\sum_t P(Q_n = q|x_{1:N})} \end{aligned} \tag{12}$$

How do we compute the quantities in (8), (9) and (10)? This is answered by the forward-backward recursion. Specifically, define the α probability

$$\alpha(n, l) = p(x_1, \ldots, x_n, Q_n = l), \tag{13}$$

where x_t is the feature vector at time t, Q_t is the hidden state at time t. It follows from the conditional independent assumption of HMM that

$$\alpha(n,l) = \sum_k \alpha(n-1,k) P(Q_n = l | Q_{n-1} = k) p(x_n | Q_n = l). \tag{14}$$

Similarly, define the β probability

$$\beta(n,l) = p(x_{n+1}, \ldots, x_N | Q_n = l, x_1, \ldots, x_n), \tag{15}$$

and it follows that

$$\beta(n,l) = \sum_k \beta(n+1,k) P(Q_{n+1} = k | Q_n = l) p(x_{n+1} | Q_{n+1} = k). \tag{16}$$

From (13) and (15), it follows that the joint probability of Q_n and $x_1, \ldots, x_N$ is given by

$$p(Q_n = l, x_1, \ldots, x_N) = \alpha(n,l)\beta(n,l), \tag{17}$$

and the conditional probability of Q_n on $x_1, \ldots, x_N$ is given by

$$P(Q_n = l | x_1, \ldots, x_N) = \frac{\alpha(n,l)\beta(n,l)}{\sum_{l'} \alpha(n,l')\beta(n,l')}. \tag{18}$$

The joint probability of Q_n, Q_{n-1} and $x_1, \ldots, x_N$ is given by

$$\begin{aligned} &p(Q_n = l, Q_{n-1} = k, x_1, \ldots, x_N) \\ &= \beta(n,l) p(x_n | Q_n = l) P(Q_n = l | Q_{n-1} = k) \alpha(n-1,k). \end{aligned} \tag{19}$$

To compute the α and β, the current parameter values are needed. Therefore, the parameter estimation is an iterative algorithm. In fact it is the EM algorithm as we show earlier in the course.

3 Decoding Speech

The speech signal is represented as sequence of speech features in an ASR system. To decide the words in the speech signal, accumulation of information in the speech feature sequence is necessary for the best overall hypothesis. There are two methods to be introduced here: the dynamic time warping (DTW) method and the statistical sequence recognition.

3.1 DTW

In DTW, the segmentation and classification are done deterministically and simultaneously. Let $X = \{x_1, \ldots, x_N\}$ be the feature vector sequence, we want to determine the best $Q = \{q_1, \ldots, q_N\}$, each q associating an x.

We assume that X is to be determined as one of the K hypotheses. Let each hypothesis has a reference sequence X_k. In general, the length in a reference sequence is not equal to the length of the unknown sequence. Therefore, for each X_k, the observation X is "warped" to match X_k in the best way, resulting a score s_k. Then

$$k^* = \arg\max_k s_k. \tag{20}$$

Let the distance function be defined as

$$D(i,j) = d(i,j) + \min_{p(i,j)} \{D[p(i,j)] + T[(i,j), p(i,j)]\}, \tag{21}$$

where $d(i,j)$ is the local distance between input frame i and template frame j, $p(i,j)$ is the possible predecessor of (i,j), and T is the transition cost between two points in the two-dimension space of the frame indexes. Starting from $(1,j)$, updating the (cumulative) distance function this way and we will finally reach (I,j). $D(I,j)$ gives the minimum distance between X and X_k for each j. This is the well-known dynamic programming method.

3.2 Statistical Sequence Recognition

In statistical sequence recognition, we replace the K template sequences with K models, M_k. The score of each model is the likelihood of the data given the model. That is $s_k = P(X|M_k)$. The data-likelihood, once the parameter is learned, can be computed by the forward/backward recursion. Alternatively, one can approximate the data-likelihood by the best-path. I.e.,

$$p(X|M_k) = \sum_q p(X, Q = q|M_k) \sim \max_q p(X, Q = q|M_k). \tag{22}$$

One can relate (22) to the DTW. To see this, note that

$$\begin{aligned} &\max_{q_{1:N}} p(x_{1:N}, q_{1:N}|M) \\ =&\max_{q_{1:N}} P(q_{1:N}|M) p(x_{1:N}|q_{1:N}, M) \\ =&\max_{q_{1:N}} P(q_1|M) p(x_1|q_1, M) \prod_{n=2}^{N} P(q_n|q_{n-1}, M) p(x_n|q_n, M). \end{aligned} \tag{23}$$

Since for any state sequence $q_{1:N}$, we have

$$p(x_{1:N}, q_{1:N}|M) = p(x_N|q_N, M)P(q_N|q_{N-1}, M)p(x_{1:N-1}, q_{1:N-1}|M),$$

or

$$\log(x_{1:N}, q_{1:N}|M) = \log p(x_N|q_N, M) + \log P(q_N|q_{N-1}, M) + \log p(x_{1:N-1}, q_{1:N-1}|M),$$

The dynamic programming method can be used by identifying the local cost to be $\log p(x_n|q_n, M)$, transition cost to be $\log P(q_n|q_{n-1}, M)$ and the past cumulative cost upto time $n-1$ to be $\log p(x_{1:n-1}, q_{1:n-1}|M)$.