transitions directly to the emitting state of the preceding and following phone, leaving only two non-emitting states for the entire word. Fig. 9.7 shows the expanded word.

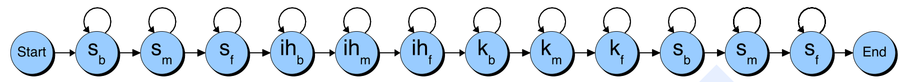


Figure 9.7 A composite word model for "six", [s ih k s], formed by concatenating four phone models, each with three emitting states.

In summary, an HMM model of speech recognition is parameterized by:

$Q = q_1 q_2 \ldots q_N$ a set of **states** corresponding to **subphones**

$A = a_{01} a_{02} \ldots a_{n1} \ldots a_{nn}$ a **transition probability matrix** A, each a_{ij} representing the probability for each subphone of taking a **self-loop** or going to the next subphone.

$B = b_i(o_t)$ A set of **observation likelihoods:**, also called **emission probabilities**, each expressing the probability of a cepstral feature vector (observation o_t) being generated from subphone state i.

Another way of looking at the A probabilities and the states Q is that together they represent a **lexicon**: a set of pronunciations for words, each pronunciation consisting of a set of subphones, with the order of the subphones specified by the transition probabilities A.

We have now covered the basic structure of HMM states for representing phones and words in speech recognition. Later in this chapter we will see further augmentations of the HMM word model shown in Fig. 9.7, such as the use of triphone models which make use of phone context, and the use of special phones to model silence. First, though, we need to turn to the next component of HMMs for speech recognition: the observation likelihoods. And in order to discuss observation likelihoods, we first need to introduce the actual acoustic observations: feature vectors. After discussing these in Sec. 9.3, we turn in Sec. 9.4 the acoustic model and details of observation likelihood computation. We then re-introduce Viterbi decoding and show how the acoustic model and language model are combined to choose the best sentence.

9.3 Feature Extraction: MFCC vectors

FEATURE VECTORS

MFCC
MEL FREQUENCY CEPSTRAL COEFFICIENTS
CEPSTRUM

Our goal in this section is to describe how we transform the input waveform into a sequence of acoustic **feature vectors**, each vector representing the information in a small time window of the signal. While there are many possible such feature representations, by far the most common in speech recognition is the **MFCC**, the **mel frequency cepstral coefficients**. These are based on the important idea of the **cepstrum**. We will give a relatively high-level description of the process of extraction of MFCCs from a

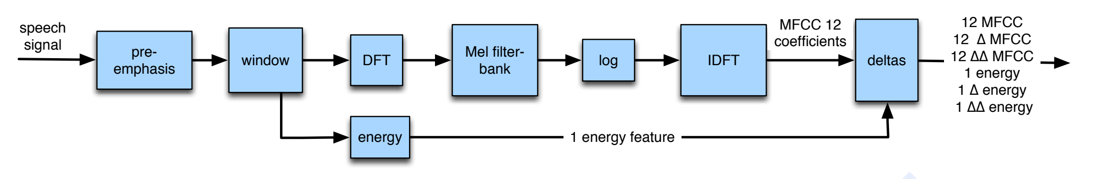


Figure 9.8 Extracting a sequence of 39-dimensional MFCC feature vectors from a quantized digitized waveform

waveform; we strongly encourage students interested in more detail to follow up with a speech signal processing course.

SAMPLING
SAMPLING RATE
NYQUIST FREQUENCY
TELEPHONE-BANDWIDTH
WIDEBAND

We begin by repeating from Sec. **??** the process of digitizing and quantizing an analog speech waveform. Recall that the first step in processing speech is to convert the analog representations (first air pressure, and then analog electric signals in a microphone), into a digital signal. This process of **analog-to-digital conversion** has two steps: **sampling** and **quantization**. A signal is sampled by measuring its amplitude at a particular time; the **sampling rate** is the number of samples taken per second. In order to accurately measure a wave, it is necessary to have at least two samples in each cycle: one measuring the positive part of the wave and one measuring the negative part. More than two samples per cycle increases the amplitude accuracy, but less than two samples will cause the frequency of the wave to be completely missed. Thus the maximum frequency wave that can be measured is one whose frequency is half the sample rate (since every cycle needs two samples). This maximum frequency for a given sampling rate is called the **Nyquist frequency**. Most information in human speech is in frequencies below 10,000 Hz; thus a 20,000 Hz sampling rate would be necessary for complete accuracy. But telephone speech is filtered by the switching network, and only frequencies less than 4,000 Hz are transmitted by telephones. Thus an 8,000 Hz sampling rate is sufficient for **telephone-bandwidth** speech like the Switchboard corpus. A 16,000 Hz sampling rate (sometimes called **wideband**) is often used for microphone speech.

QUANTIZATION

Even an 8,000 Hz sampling rate requires 8000 amplitude measurements for each second of speech, and so it is important to store the amplitude measurement efficiently. They are usually stored as integers, either 8-bit (values from -128–127) or 16 bit (values from -32768–32767). This process of representing real-valued numbers as integers is called **quantization** because there is a minimum granularity (the quantum size) and all values which are closer together than this quantum size are represented identically.

We refer to each sample in the digitized quantized waveform as $x[n]$, where n is an index over time. Now that we have a digitized, quantized representation of the waveform, we are ready to extract MFCC features. The seven steps of this process are shown in Fig. 9.8 and individually described in each of the following sections.

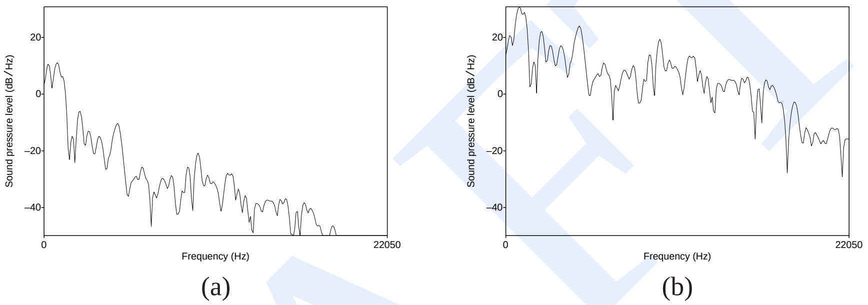
9.3.1 Preemphasis

The first stage in MFCC feature extraction is to boost the amount of energy in the high frequencies. It turns out that if we look at the spectrum for voiced segments like

SPECTRAL TILT

vowels, there is more energy at the lower frequencies than the higher frequencies. This drop in energy across frequencies (which is called **spectral tilt**) is caused by the nature of the glottal pulse. Boosting the high frequency energy makes information from these higher formants more available to the acoustic model and improves phone detection accuracy.

This preemphasis is done by using a filter[1] Fig. 9.9 shows an example of a spectral slice from the first author's pronunciation of the single vowel [aa] before and after preemphasis.



(a) (b)

Figure 9.9 A spectral slice from the vowel [aa] before (a) and after (b) preemphasis.

9.3.2 Windowing

NON-STATIONARY

STATIONARY

Recall that the goal of feature extraction is to provide spectral features that can help us build phone or subphone classifiers. We therefore don't want to extract our spectral features from an entire utterance or conversation, because the spectrum changes very quickly. Technically, we say that speech is a **non-stationary** signal, meaning that its statistical properties are not constant across time. Instead, we want to extract spectral features from a small **window** of speech that characterizes a particular subphone and for which we can make the (rough) assumption that the signal is **stationary** (i.e. its statistical properties are constant within this region).

We'll do this by using a window which is non-zero inside some region and zero elsewhere, running this window across the speech signal, and extracting the waveform inside this window.

FRAME

FRAME SIZE

FRAME SHIFT

We can characterize such a windowing process by three parameters: how **wide** is the window (in milliseconds), what is the **offset** between successive windows, and what is the **shape** of the window. We call the speech extracted from each window a **frame**, and we call the number of milliseconds in the frame the **frame size** and the number of milliseconds between the left edges of successive windows the **frame shift**.

The extraction of the signal takes place by multiplying the value of the signal at

[1] For students who have had signal processing: this preemphasis filter is a first-order high-pass filter. In the time domain, with input $x[n]$ and $0.9 \leq \alpha \leq 1.0$, the filter equation is $y[n] = x[n] - \alpha x[n-1]$.

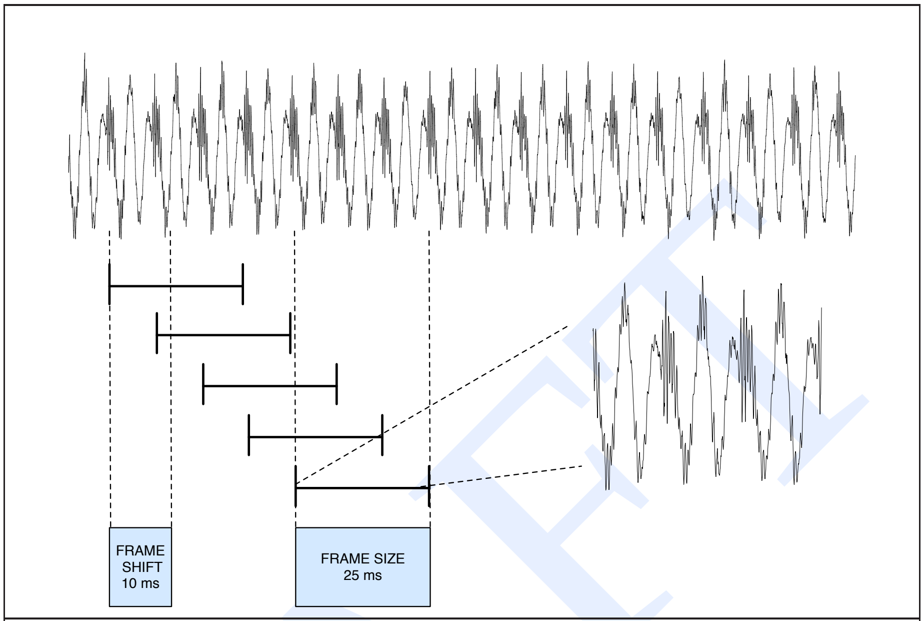


Figure 9.10 The windowing process, showing the frame shift and frame size, assuming a frame shift of 10ms, a frame size of 25 ms, and a rectangular window. After a figure by Bryan Pellom.

time n, $s[n]$, with the value of the window at time n, $w[n]$:

$$y[n] = w[n]s[n] \tag{9.9}$$

Figure 9.10 suggests that these window shapes are rectangular, since the extracted windowed signal looks just like the original signal. Indeed the simplest window is the RECTANGULAR **rectangular** window. The rectangular window can cause problems, however, because it abruptly cuts of the signal at its boundaries. These discontinuities create problems when we do Fourier analysis. For this reason, a more common window used in MFCC extraction is the HAMMING **Hamming** window, which shrinks the values of the signal toward zero at the window boundaries, avoiding discontinuities. Fig. 9.11 shows both of these windows; the equations are as follows (assuming a window that is L frames long):

$$rectangular \qquad w[n] = \begin{cases} 1 & 0 \le n \le L-1 \\ 0 & \text{otherwise} \end{cases} \tag{9.10}$$

$$hamming \qquad w[n] = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi n}{L}) & 0 \le n \le L-1 \\ 0 & \text{otherwise} \end{cases} \tag{9.11}$$

9.3.3 Discrete Fourier Transform

The next step is to extract spectral information for our windowed signal; we need to know how much energy the signal contains at different frequency bands. The tool for

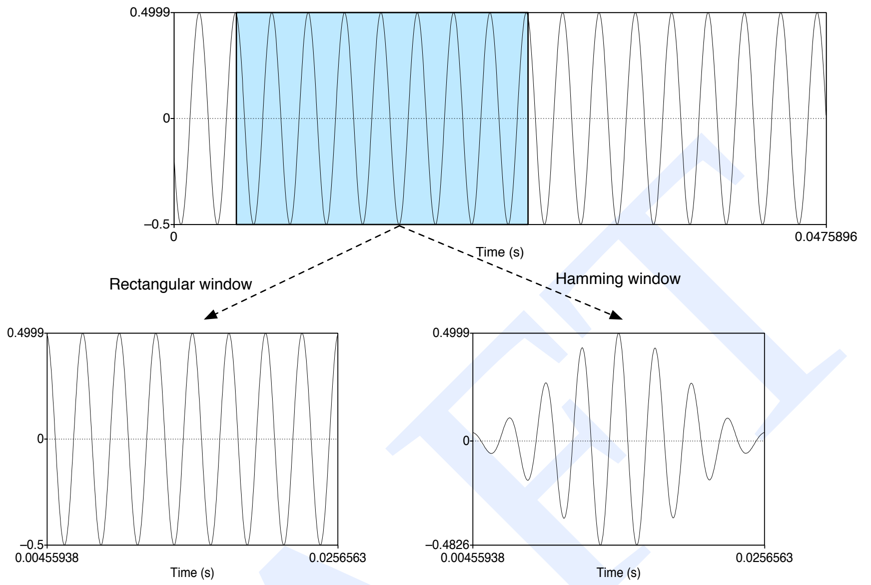


Figure 9.11 Windowing a portion of a pure sine wave with the rectangular and Hamming windows.

extracting spectral information for discrete frequency bands for a discrete-time (sampled) signal is the **Discrete Fourier Transform** or **DFT**.

DISCRETE FOURIER TRANSFORM

DFT

The input to the DFT is a windowed signal $x[n]...x[m]$, and the output, for each of N discrete frequency bands, is a complex number $X[k]$ representing the magnitude and phase of that frequency component in the original signal. If we plot the magnitude against the frequency, we can visualize the **spectrum** that we introduced in Ch. 7. For example, Fig. 9.12 shows a 25 ms Hamming-windowed portion of a signal and its spectrum as computed by a DFT (with some additional smoothing).

We will not introduce the mathematical details of the DFT here, except to note that Fourier analysis in general relies on **Euler's formula**:

EULER'S FORMULA

$$e^{j\theta} = \cos\theta + j\sin\theta \tag{9.12}$$

As a brief reminder for those students who have already had signal processing, the DFT is defined as follows:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\frac{\pi}{N}kn} \tag{9.13}$$

A commonly used algorithm for computing the DFT is the the **Fast Fourier Transform** or **FFT**. This implementation of the DFT is very efficient, but only works for values of N which are powers of two.

FAST FOURIER TRANSFORM

FFT

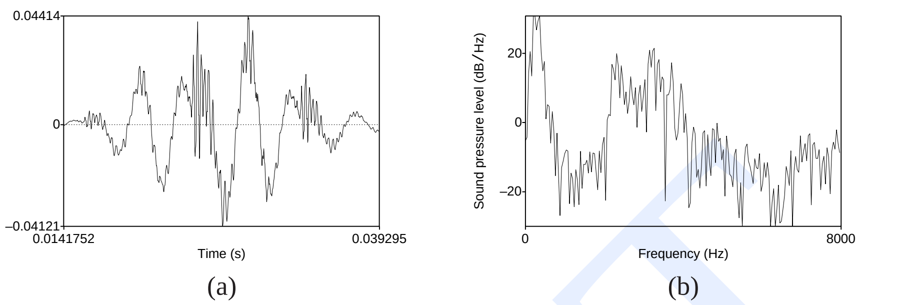


Figure 9.12 (a) A 25 ms Hamming-windowed portion of a signal from the vowel [iy] and (b) its spectrum computed by a DFT.

9.3.4 Mel filter bank and log

The results of the FFT will be information about the amount of energy at each frequency band. Human hearing, however, is not equally sensitive at all frequency bands. It is less sensitive at higher frequencies, roughly above 1000 Hertz. It turns out that modeling this property of human hearing during feature extraction improves speech recognition performance. The form of the model used in MFCCs is to warp the frequencies output by the DFT onto the **mel** scale mentioned in Ch. 7. A **mel** (Stevens et al., 1937; Stevens and Volkmann, 1940) is a unit of pitch defined so that pairs of sounds which are perceptually equidistant in pitch are separated by an equal number of mels. The mapping between frequency in Hertz and the mel scale is linear below 1000 Hz and the logarithmic above 1000 Hz. The mel frequency *m* can be computed from the raw acoustic frequency as follows:

MEL

$$mel(f) = 1127\ln(1+\frac{f}{700}) \qquad (9.14)$$

During MFCC computation, this intuition is implemented by creating a bank of filters which collect energy from each frequency band, with 10 filters spaced linearly below 1000 Hz, and the remaining filters spread logarithmically above 1000 Hz. Fig. 9.13 shows the bank of triangular filters that implement this idea.

Finally, we take the log of each of the mel spectrum values. In general the human response to signal level is logarithmic; humans are less sensitive to slight differences in amplitude at high amplitudes than at low amplitudes. In addition, using a log makes the feature estimates less sensitive to variations in input (for example power variations due to the speaker's mouth moving closer or further from the microphone).

9.3.5 The Cepstrum: Inverse Discrete Fourier Transform

While it would be possible to use the mel spectrum by itself as a feature representation for phone detection, the spectrum also has some problems, as we will see. For this reason, the next step in MFCC feature extraction is the computation of the **cepstrum**. The

CEPSTRUM

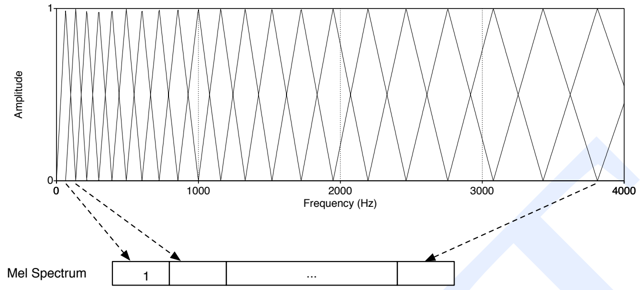
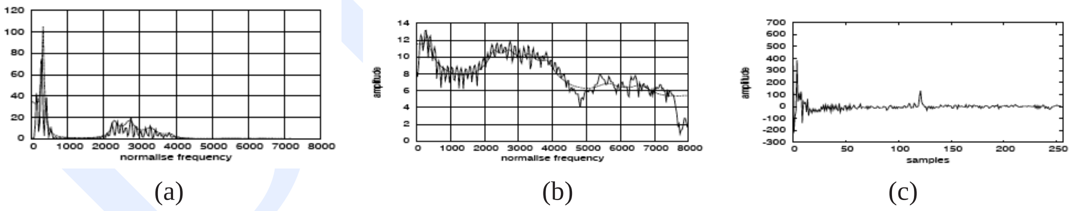


Figure 9.13 The Mel filter bank, after Davis and Mermelstein (1980). Each triangular filter collects energy from a given frequency range. Filters are spaced linearly below 1000 Hz, and logarithmically above 1000 Hz.

cepstrum has a number of useful processing advantages and also significantly improves phone recognition performance.

One way to think about the cepstrum is as a useful way of separating the **source** and **filter**. Recall from Sec. **??** that the speech waveform is created when a glottal source waveform of a particular fundamental frequency is passed through the vocal tract, which because of its shape has a particular filtering characteristic. But many characteristics of the glottal **source** (its fundamental frequency, the details of the glottal pulse, etc) are not important for distinguishing different phones. Instead, the most useful information for phone detection is the **filter**, i.e. the exact position of the vocal tract. If we knew the shape of the vocal tract, we would know which phone was being produced. This suggests that useful features for phone detection would find a way to deconvolve (separate) the source and filter and show us only the vocal tract filter. It turns out that the cepstrum is one way to do this.



(a) (b) (c)

Figure 9.14 PLACEHOLDER FIGURE. The magnitude spectrum (a), the log magnitude spectrum (b), and the cepstrum (c). From Taylor (2008). The two spectra have a smoothed spectral enveloped laid on top of them to help visualize the spectrum.

For simplicity, let's ignore the pre-emphasis and mel-warping that are part of the definition of MFCCs, and look just at the basic definition of the cepstrum. The cepstrum can be thought of as the *spectrum of the log of the spectrum*. This may sound confusing. But let's begin with the easy part: the *log of the spectrum*. That is, the cepstrum begins with a standard magnitude spectrum, such as the one for a vowel shown in Fig. 9.14(a) from Taylor (2008). We then take the log, i.e. replace each amplitude value in the magnitude spectrum with its log, as shown in Fig. 9.14(b).

The next step is to visualize the log spectrum *as if itself were a waveform*. In other words, consider the log spectrum in Fig. 9.14(b). Let's imagine removing the axis labels that tell us that this is a spectrum (frequency on the x-axis) and imagine that we are dealing with just a normal speech signal with time on the x-axis. Now what can we say about the spectrum of this 'pseudo-signal'? Notice that there is a high-frequency repetitive component in this wave: small waves that repeat about 8 times in each 1000 along the x-axis, for a frequency of about 120 Hz. This high-frequency component is caused by the fundamental frequency of the signal, and represents the little peaks in the spectrum at each harmonic of the signal. In addition, there are some lower frequency components in this 'pseudo-signal'; for example the envelope or formant structure has about four large peaks in the window, for a much lower frequency.

Fig. 9.14(c) shows the **cepstrum**: the spectrum that we have been describing of the log spectrum. This cepstrum (the word **cepstrum** is formed by reversing the first letters of **spectrum**) is shown with **samples** along the x-axis. This is because by taking the spectrum of the log spectrum, we have left the frequency domain of the spectrum, and gone back to the time domain. It turns out that the correct unit of a cepstrum is the sample.

Examining this cepstrum, we see that there is indeed a large peak around 120, corresponding to the F0 and representing the glottal pulse. There are other various components at lower values on the x-axis. These represent the vocal tract filter (the position of the tongue and the other articulators). Thus if we are interested in detecting phones, we can make use of just the lower cepstral values. If we are interested in detecting pitch, we can use the higher cepstral values.

For the purposes of MFCC extraction, we generally just take the first 12 cepstral values. These 12 coefficients will represent information solely about the vocal tract filter, cleanly separated from information about the glottal source.

It turns out that cepstral coefficients have the extremely useful property that the variance of the different coefficients tends to be uncorrelated. This is not true for the spectrum, where spectral coefficients at different frequency bands are correlated. The fact that cepstral features are uncorrelated means, as we will see in the next section, that the Gaussian acoustic model (the Gaussian Mixture Model, or GMM) doesn't have to represent the covariance between all the MFCC features, which hugely reduces the number of parameters.

For those who have had signal processing, the cepstrum is more formally defined as the **inverse DFT of the log magnitude of the DFT of a signal**, hence for a windowed frame of speech $x[n]$:

$$c[n] = \sum_{n=0}^{N-1} log \left(\left| \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \right| \right) e^{j\frac{2\pi}{N}kn} \tag{9.15}$$

9.3.6 Deltas and Energy

The extraction of the cepstrum via the Inverse DFT from the previous section results in 12 cepstral coefficients for each frame. We next add a thirteenth feature: the energy

ENERGY

from the frame. Energy correlates with phone identity and so is a useful cue for phone detection (vowels and sibilants have more energy than stops, etc). The **energy** in a frame is the sum over time of the power of the samples in the frame; thus for a signal x in a window from time sample t_1 to time sample t_2, the energy is:

$$Energy = \sum_{t=t_1}^{t_2} x^2[t] \tag{9.16}$$

Another important fact about the speech signal is that it is not constant from frame to frame. This change, such as the slope of a formant at its transitions, or the nature of the change from a stop closure to stop burst, can provide a useful cue for phone identity. For this reason we also add features related to the change in cepstral features over time.

DELTA

VELOCITY

DOUBLE DELTA

ACCELERATION

We do this by adding for each of the 13 features (12 cepstral features plus energy) a **delta** or **velocity** feature, and a **double delta** or **acceleration** feature. Each of the 13 delta features represents the change between frames in the corresponding cepstral/energy feature, while each of the 13 double delta features represents the change between frames in the corresponding delta features.

A simple way to compute deltas would be just to compute the difference between frames; thus the delta value $d(t)$ for a particular cepstral value $c(t)$ at time t can be estimated as:

$$d(t) = \frac{c(t+1) - c(t-1)}{2} \tag{9.17}$$

Instead of this simple estimate, however, it is more common to make more sophisticated estimates of the slope, using a wider context of frames.

9.3.7 Summary: MFCC

After adding energy, and then delta and double-delta features to the 12 cepstral features, we end up with 39 MFCC features:

12	cepstral coefficients
12	delta cepstral coefficients
12	double delta cepstral coefficients
1	energy coefficient
1	delta energy coefficient
1	double delta energy coefficient
39	MFCC features

Again, one of the most useful facts about MFCC features is that the cepstral coefficients tend to be uncorrelated, which will turn out to make our acoustic model much simpler.