1 Recommended books

1.1 **Speech Technology**

1.1.1 John Holmes and Wendy Holmes, *"Speech Synthesis and Recognition, 2nd Edition",* Taylor & Francis, 2001

1.1.2 L. Rabiner and B-H. Juang, "*Fundamentals of Speech Recognition*", Prentice Hall, 1993

1.1.3 L R Rabiner and R W Schafer, "*Digital Processing of Speech Signals*", Prentice-Hall, 1978

1.1.4 J.R. Deller, J.H.L. Hansen, J.G. Proakis, "*Discrete-Time Processing of Speech Signals*", IEEE Press, 1999

1.2 **Pattern Recognition**

1.2.1 Christopher M Bishop, *"Neural Networks for Pattern Recognition",* Oxford, 1995, ISBN 0-19-853864-2

"The pen was an archaic instrument, seldom used even for signatures ..... Actually he was not used to writing by hand. Apart from very short notes, it was usual to dictate everything into the speak-write".

George Orwell, "Nineteen-Eighty-Four", 1949.

2 What is Speech Processing?

2.1 Component Technologies

2.1.1 Speech processing is concerned with all aspects of the interaction between computers, speech and spoken language. It includes a number of 'component technologies':

Automatic Speech Recognition
Spoken Language Understanding
Spoken Dialogue Processing
Paralinguistic Speech Processing
Speech Verification
Speech Synthesis
Spoken Language Generation
Speech Coding

2.2 Automatic speech recognition

2.2.1 Automatic Speech Recognition, or 'ASR', is the technology which underlies the current generation of automatic dictation systems, such as Dragon 'Naturally Speaking', IBM 'Via Voice' and Speech Machines 'CyberTranscriber'. Strictly speaking, ASR is concerned with automatic **transcription** of speech i.e. converting an acoustic speech signal into a verbatim text transcription of that speech. Anything beyond this involves **interpretation** of the speech signal, and this is the realm of **spoken language understanding**.

2.3 Spoken Language Understanding

2.3.1 Spoken Language Understanding, or 'SLU', is concerned with the relationship between an acoustic speech signal and its associated **meaning**. SLU is related to Natural Language Understanding (NLU), but NLU is normally concerned with written, rather than spoken language, and we shall see later that it is an error to regard speech as "acoustic text". ASR is one of the component technologies of SLU, but SLU is much broader, and places a much greater emphasis on language processing, or syntactic processing, and semantics. Although SLU has been the subject of considerable research over the past few decades, it is still in its infancy.

2.3.2 SLU can be concerned purely with passive understanding of a given utterance. When the role of the SLU system becomes active this leads to the concept of **dialogue**.

2.4 Spoken Dialogue Processing

2.4.1 Spoken Dialogue Processing is concerned with interaction between two or more people or machines using speech. It is fundamental to many applications of spoken language processing. For example most information gathering applications, such as automatic railway timetable query, will require some form of interactive dialogue between the user and the computer to establish the exact nature of the query. Although there are examples of laboratory-based spoken dialogue systems, they are typically very rudimentary and only function by allocating much of the 'initiative' in the dialogue to the computer, so that the human role in the dialogue is restricted to 'yes', 'no' or simple phrases.

2.5 Paralinguistic Speech Processing

2.5.1 In this context, 'paralinguistic' refers to properties of a speech signal which go beyond its linguistic content (i.e. which go beyond the words and meaning included in the speech). Paralinguistic speech processing includes **speaker recognition**, **speaker verification**, **speaker gender identification**, **language recognition**,… This is a fascinating area, with potential or real applications in forensic science, access control and other security applications, biometrics, and many other areas of technology.

2.5.2 The concept of paralinguistic speech processing is often broadened to include **spoken topic spotting**, which is normally concerned with determining whether or not a particular section of speech is relevant to a particular topic. Arguably, this is not really paralinguistic processing but 'broad focus' spoken language understanding. It is closely linked with **gisting**, or summarisation of a section of speech. Spoken topic spotting is applicable whenever relevant sections need to be identified in a large corpus of spoken data. It is a 'speech' analogy of **information retrieval** from text databases.

2.6 Speech Verification

2.6.1 Speech verification is related to automatic speech recognition, however, instead of being concerned with recognising what was said, it is concerned with verifying that what was **actually** spoken is the same as what was **expected** to be spoken, or was **pronounced** appropriately. Although this may seem like a fine distinction, speech verification is the technology which underlies a number of important educational applications of speech technology, such as **automatic interactive language tuition** and **automatic interactive reading tuition**.

2.6.2 Some of the tasks listed under paralinguistic speech processing, such as speaker verification, can also be regarded as verification tasks.

2.7 Speech Synthesis

2.7.1 Speech synthesis is one of the key component technologies. It is concerned with automatically translating a symbolic description of speech, such as a sequence of words, into an acoustic speech signal. It typically involves several stages, text-to-phoneme conversion, conversion of a phoneme sequence into a sequence of synthesiser control parameters, and synthesis of an acoustic signal using these parameters.

2.7.2 Note that there is a 'duality' between speech synthesis and speech recognition both are concerned with verbatim translation between symbolic and acoustic representations of speech, and neither is concerned with understanding. The speech synthesis analogy of spoken language understanding is **spoken language generation**.

2.8 Spoken Language Generation

2.8.1 Spoken Language Generation (SLG) is concerned with generating spoken language from meaning, or concept. It is related to Natural Language Generation (NLG), except that NLG is mainly concerned with written language. Like SLU, NLU is much less developed than speech synthesis.

2.8.2 One can argue that, ultimately, speech recognition and speech understanding are inseparable, since the meaning of an utterance will affect its acoustic realisation. In speech synthesis this is even more the case. One of the most commonly cited shortcomings of current speech synthesis is its **prosodic** structure. Prosody refers to the durational structure of a speech signal (the relative lengths of its different parts, and the presence and lengths of any pauses), its amplitude structure (the relative amplitudes of its different parts), and its **intonational** structure (the variations in pitch). Prosody includes the notion of **stress**. The use of prosodic structure is closely linked with meaning. Consider the following example, from (Altmann (1997))[1]

Joe: Hey-did you hear? Sam took Mary out and bought her a pizza!

Mike: You're wrong - Sam didn't buy Mary a pizza

2.8.3 This seems pretty straightforward - but it's not. Did Mike say:

"*Sam didn't buy Mary a PIZZA"* (he bought her something else)

"*Sam didn't buy MARY a pizza"* (he bought it for someone else)

---

[1] Gerry T Altmann, "The Ascent of Babel", Oxford University Press, 1997.

"Sam didn't BUY Mary a pizza" (he made her one)

"*Sam DIDN'T buy Mary a pizza"* (he did something else, or nothing)

"SAM didn't buy Mary a pizza" (someone else did)

2.8.4 Convinced?

2.9 **Note on Terminology**

2.9.1 Unfortunately, speech processing is an area where different sub-cultures have invented their own terminology. In these notes I will try to point out areas where there is potential confusion, but new ones seem to emerge daily! For example, you may here people talk of **DVI** (Direct Voice Input) instead of automatic speech recognition, and **DVO** (Direct Voice Output) instead of synthesis. Some people use **Voice Recognition** synonymously with speech recognition, while others use the same term to mean speaker recognition and others use it to mean speaker-dependent speech recognition (see the later section on speech recognition for the definition of this)! Beware!

3 Human Speech Production

3.1 The Vocal Organs

3.1.1 The main organs in the human body responsible for speech production are the **lungs**, **larynx**, **pharynx**, **nose** and various parts of the **mouth** (figure 1). The energy in the system is produced by muscular force expelling air from the lungs to produce one of three types of sound source. The properties of the resulting sounds are modified according to the resonant properties of the **vocal tract**. This is the collection of vocal organs above and including the larynx.

3.2 Types of Sound Source

3.2.1 There are basically three types of sound source:

3.2.1.1 For **voiced** sounds, for example vowels, the air flow from the lungs is modulated by vibrations of the vocal cords, or vocal folds, in the larynx. The fundamental frequency of this signal lies somewhere between 50Hz (low adult male) and 400Hz (high adult female) or even higher for small children.

3.2.1.2 For **fricative** sounds, the air flow from the lungs is forced through a narrow constriction in the vocal tract to produce **turbulent** noise, or **frication**. For example, constrictions can be made between the tongue and the roof of the mouth, between the lips, or even in the region of the larynx (as in "h" sounds).

3.2.1.3 Finally, **plosive** sounds occur when the vocal tract is temporarily closed at some point. Pressure builds up and is then released, causing a transient excitation of the vocal tract.

3.3 The Vocal Tract Transfer Function

3.3.1 Whichever of the three types of sound is produced is modified by the **resonant** properties of the vocal tract. The main resonant modes of the vocal tract are called **formants**. They are usually referred to as F_1, F_2, F_3,.. etc, starting with the low frequency resonances.

3.3.2 A simple account of vocal tract acoustics considers two cases, corresponding to whether the **soft palate**, or **velum** is raised (closing off the nasal cavity from the rest of the vocal tract) or lowered.

3.3.3 If the velum is **raised** there is no opening between the nose and throat and the vocal tract can be modelled as an unbranched air-filled acoustic tube with a large number of cylindrical sections of different diameters joined together. For the purposes of this course, the resulting system can be thought of as a resonant system with a transfer function with poles but no zeros, i.e. an **all-pole transfer function**. For a more detailed description see Holmes (1988).

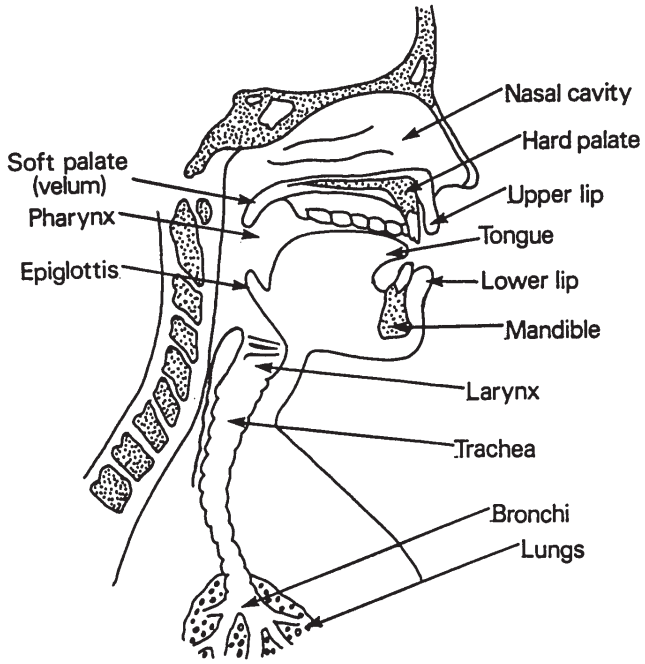


Figure 1: Schematic cross-section of the human head, showing the vocal organs (Holmes, 1988)

3.3.4 If the velum is **raised** there is no opening between the nose and throat and the vocal tract can be modelled as an unbranched air-filled acoustic tube with a large number of cylindrical sections of different diameters joined together. For the purposes of this course, the resulting system can be thought of as a resonant system with a transfer function with poles but no zeros, i.e. an **all-pole transfer function**. For a more detailed description see Holmes (1988).

3.3.5 On the other hand, if the velum is **lowered**, the nasal cavity is included in the vocal tract. For a variety of reasons, this makes the acoustic much more complex. For the purposes of this course, the result is that spectral **zeros**, as well as poles, appear in the vocal tract transfer function. It will be seen later that this has an effect on the suitability of some speech analysis techniques.

3.4 The Source-Filter model

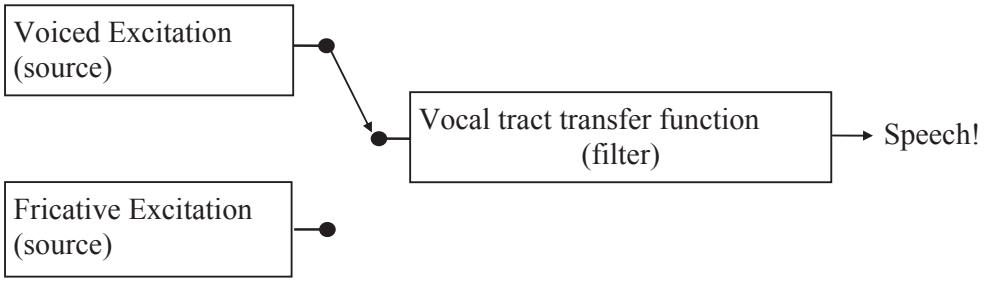


Figure 2: The source-filter model of speech production

3.4.1 The implicit model of speech production which has been described above is the **source-filter model**. This model recurs in many different parts of speech technology, and certainly in recognition, synthesis and coding. The model can be summarised in the simple block diagram shown in figure 2:

3.5 Elementary Speech Analysis

3.5.1 **Speech analysis**, or **speech signal processing**, or **front-end processing**, or **pre-processing** all refer to the initial transformation of a speech waveform into a form which is more **suitable** for **analysis**. In this context, **analysis** might be:

- Human visual inspection - for example by a phonetician, speech scientist, speech therapist or forensic phonetician
- Computer analysis - for example for automatic speech recognition, speaker recognition or paralinguistic processing

3.5.2 and **suitable** might mean:

- Amenable to human visual interpretation
- Requiring a sufficiently small number of bits per second to allow transmission across a communications channel or storage on a particular device.
- Compatible with the assumptions in a particular speech model for speech recognition

3.5.3 Analogous with our understanding of the analysis which is performed in the human peripheral auditory system

3.5.4 In this initial section on speech analysis we shall concentrate on a representation which is particularly suitable for human inspection, amenable to human interpretation and analogous with our understanding of the analysis which is performed in the human peripheral auditory system. This representation is the **speech spectrogram**.

3.5.5 A speech spectrogram (figure 3) is a **time-frequency** representation of a speech signal:

- the horizontal axis represents time,
- the vertical axis represents frequency, and
- the intensity (or colour) of a point in the time-frequency plane represents the relative power, or amplitude, of the speech signal over a particular time in a particular frequency band.

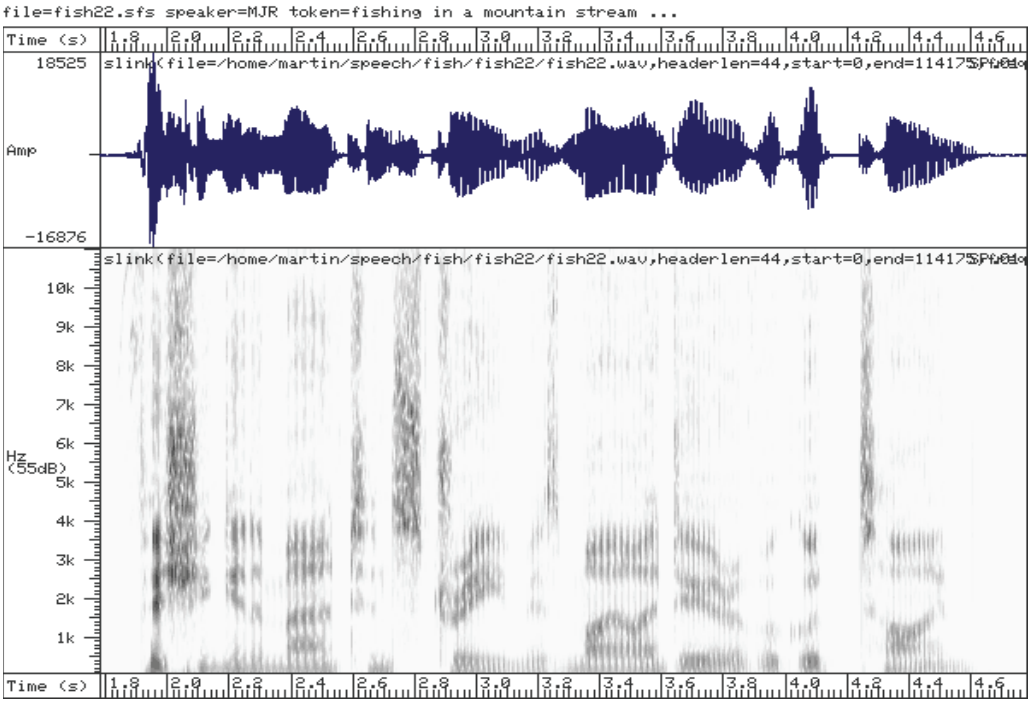


Figure 3: Speech spectrogram of the utterance "Fishing in a mountain stream is my idea of a good time". The speech signal was samples at 22,000 samples per second.

3.5.6 If t is a particular time, then a vertical 'slice' through the spectrogram represents the distribution of power with respect to frequency over a short time interval centred at time t.

3.5.7 Such a description is of interest on two accounts:

3.5.7.1 From the perspective of the source-filter model, it tells us about the shape of the vocal tract at time *t*

3.5.7.2 From the perspective of human speech perception, we know that a similar analysis is performed in the cochlea in the initial stages of human speech perception

3.5.8 The precise details of how such a representation is derived are not covered in this course, though you will probably have encountered most of them in courses on **signal processing** courses. However, starting with an analogue speech waveform, the following stages are involved:

3.6 Analogue-to-digital conversion, or digitisation

3.6.1 The simplest approaches to analogue to digital conversion measure and encode the amplitude of the speech waveform at regular sampling points. This approach is referred to as **Pulse Code Modulation (PCM)** (figure 4).

3.6.2 PCM will come up again in our discussion of speech coding. This method of coding does not exploit the knowledge that the signal in question is speech and is therefore applicable to any signal. For any given type of signal, the bit rates which can be achieved are dictated by general principles from information theory.

3.6.3 In order to faithfully encode a signal with frequency components up to *N*Hz, **Nyquist's theorem** states that a sample rate of 2*N* samples per second is needed. Conversely, prior to applying sampling to a signal at a rate of *2N* samples per second, the signal must be passed through a low-pass filter with a cut off frequency of *N*Hz.

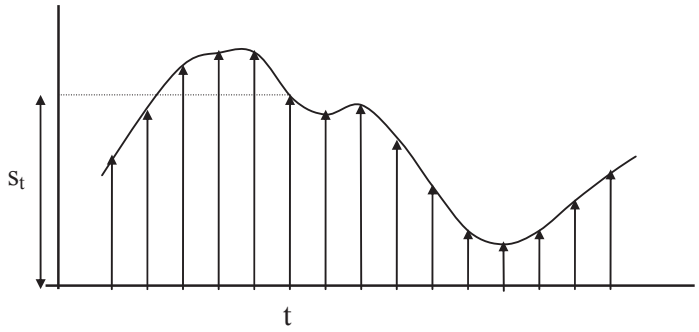


Figure 4: Pulse Code Modulation (PCM). The amplitude of the speech signal is sampled at regular points and stored as a fixed number of bits.

3.6.4 The human ear is sensitive to frequencies up to around 20,000Hz. Therefore, according to Nyquist's theorem, a sample rate of at least 40,000 samples per second is needed to capture high quality audio. For example, the sample rate on audio compact discs is 44,000 samples per second.

3.6.5 In order to encode high quality speech it is sufficient to preserve frequencies up to 10,000Hz. Hence the Nyquist sampling rate for high-quality speech is around 20,000 samples per second. Assuming that each sample is encoded in 16 bits, this results in a bit rate of 320,000 bps.

3.6.6 For speech, this bit rate can be reduced in a number of ways. The bandwidth can be cut to 4,000Hz or less, whilst still preserving intelligibility (the bandwidth of normal, civil, telephone speech is approximately 3,500Hz). Consequently the sampling rate can be reduced to 8,000 samples per second. For example, civil telephony uses PCM with 8 bits per sample (achieved by amplitude compression) and 8,000 Hz sampling rate, giving a standard bit rate of 64,000bps.

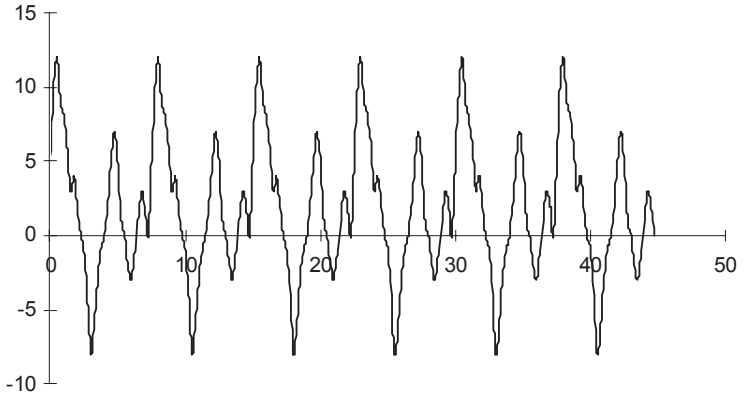


Figure 5(a) : An example analogue speech signal

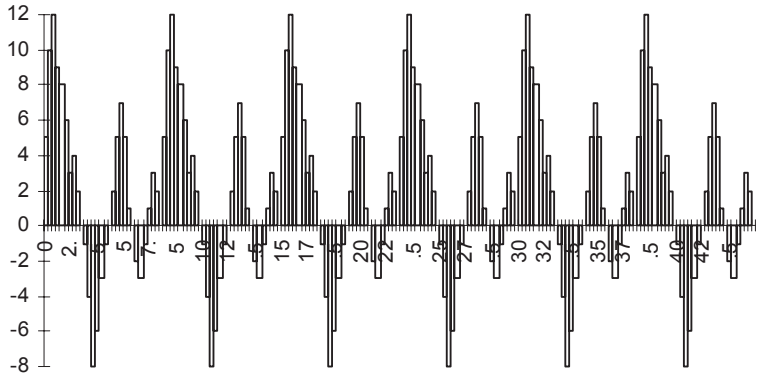


Figure 5(b) : The corresponding digitised signal

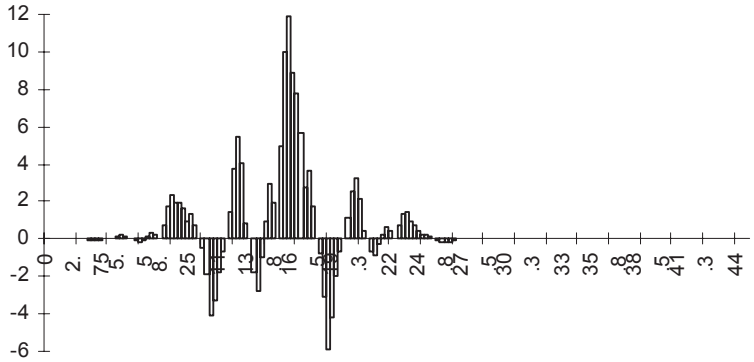


Figure 5(c): Windowed digitised signal

3.6.7 Figure 5(a) shows a synthetic analogue speech signal and figure 5(b) shows a representation of the digitised signal.

3.7 **Windowing**

3.7.1 A point in a spectrogram represents an estimate of the power in a short frequency band over a short interval of time. For a particular time t, it is obtained by applying some form of frequency analysis to a short interval (typically approximately 20ms) of speech waveform. A smooth windowing function is applied to ensure that the signal is zero outside this window and that the 'cutoff' at the edge of the window is not sharp (figure 5(c)).

3.8 **Frequency Analysis**

3.8.1 A Discrete Fourier Transform (DFT) is applied to the windowed digital waveform $\{s(t):t=1,...,N\}$.

3.8.2 Assuming that the window comprises N sample points, this results in an N/2 point complex spectrum $\{S(f): f=1,..,N/2\}$.

3.8.3 The modules is taken to obtain an *N/2* point power spectrum

$\{P(f)=|S(f)|: f=1,...,N/2\}$.

This amounts to ignoring the phase of the speech waveform, and is justified by the observation that human hearing is relatively insensitive to changes in phase.

3.8.4 Finally a logarithm is taken to compress the dynamic range of the values, resulting in the log-power spectrum

$\{LP(f)=log|S(f)|:f=1,...,N/2\}$.

Again this is motivated by results from psycho-acoustics which show that the logarithmic scale is more directly related to a 'perceptual scale'. It also results in a spectrogram which is easier to interpret visually.

3.8.5 From now on the log-power spectrum, computed in this way over a short window centred at time *t*, will be referred to as the short-term (Fourier) spectrum at time *t*.

3.9 **Example**

3.9.1 Suppose that a speech waveform is sampled at 8,000 samples per second. Then a 20ms window will contain 160 points (*N*=160), resulting is an 80 point spectrum. A window of 16ms would give a 128 point window, which might be chosen for application of the Fast Fourier Transform.

3.10 **The Mel scale**

3.10.1 If a human subject is played a pure tone at 1,000Hz and then asked to adjust the frequency of a second tone so that its pitch appears to be half that of the

first tone, then on average they will select a frequency of around 400Hz, rather than 500Hz.

3.10.2 On the other hand, if you ask them to adjust the frequency so that the pitch is doubled, then they will choose a frequency of around 4,000Hz. This demonstrates that it is necessary to distinguish between perceptual notions, such as pitch, and measurable properties, such as frequency. It also indicates that the linear frequency scale may not be the most perceptually relevant frequency scale.

3.10.3 The perceptual frequency scale based on the experiment describe above is called the mel scale. 1,000 mels is taken to be a frequency of 1,000 Hz, but, as we have seen above, 500 mels corresponds to a frequency of around 400 Hz. Figure 6 shows the relationship between measured frequency (measured in Hz) and subjective pitch (measured in mels).

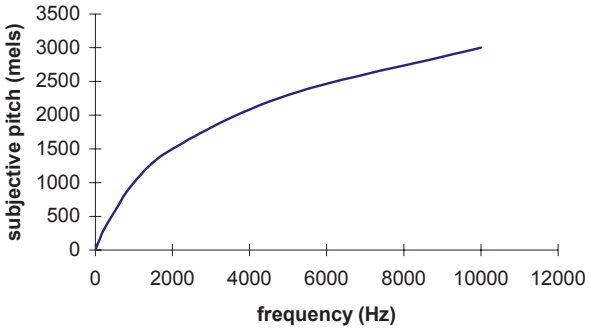


Figure 6: Relationship between frequency (measured in Hz) and subjective pitch (measured in mels)

3.10.4 We shall see later that the mel scale is used in front-end analysis for automatic speech recognition. An engineering approximation to the mel scale is to use a scale which is linear up to about 1,000Hz and logarithmic above 1,000Hz.

3.11 Bandwidth: Wide-band and narrow-band spectrograms

3.11.1 It should be clear from the above that there is a **trade-off** between frequency resolution and time (or temporal) resolution. If the window is **long** then the number of points *N/2* in the frequency analysis is large and so the number of points in the spectrum is large, resulting in relatively **fine frequency resolution**. However, the same long window will clearly result in **poor temporal resolution**. Thus a long window corresponds to a narrow-band frequency analysis. The resulting spectra are referred to as a **narrow-band spectra**.

3.11.2 Conversely, a short time window results in a **coarse resolution**, **broad-band** frequency analysis, but **fine temporal resolution**. The resulting spectra are referred to as **broad-band spectra**.

- **Long** time window ⇒ **narrow** band spectrum
- **Short** time window ⇒ **broad** band spectrum

3.11.3 Spectrograms which are produced using narrow-band and broad-band analysis are referred to as **narrow-band spectrograms** and **broad-band spectrograms**, respectively.

3.11.4 Intuitively we can think of the value of the short-term spectrum at frequency *f* as the output of a bandpass filter centred at *f* with bandwidth determined by the length of the analysis window. For a Hamming window the bandwidth of this filter is approximately $4N/L$, where L is the length in samples of the analysis window.

3.11.5 Both types of spectrogram are useful to phoneticians and speech scientists, as they reveal different aspects of the structure of speech patterns. Figure 7 shows an example of a broad-band spectrogram (bandwidth 200Hz, top spectrogram) and a narrow-band spectrogram (bandwidth 30Hz, bottom spectrogram). The useful range of bandwidths for speech analysis is between 25Hz and 400Hz.

3.11.6 The broadband spectrogram is the type of spectrogram which is most often used to inspect speech signals in speech technology. It has good temporal resolultion, but still shows the important structure in the frequency domain.

3.12 Window Size and Fundamental Frequency

3.12.1 What promised to be the straightforward process of frequency analysis has become quite complicated. Unfortunately its about to become a bit more complicated!

3.12.2 The fundamental frequency for adult males typically lies in the range 50Hz - 200Hz, and between 100Hz and 400Hz (an octave higher than for males) for an adult female speaker. If we choose a window size of 20ms (one fiftieth of a second) for short-term frequency analysis, then for an adult male speaker with a low fundamental frequency of 50Hz, the window size corresponds approximately to the space between larynx pulses. If a pulse lies in the middle of the window (figure 9, window W2), the resulting spectrum will be quite different to the spectrum which would occur if the window lay between larynx pulses (figure 8, window W1). This concern leads to the notion of pitch-synchronous short-term frequency analysis, where the position of the analysis window is synchronised with the larynx signal.

3.12.3 But that's not all. Now consider performing the same analysis on the speech of an adult female speaker with a high fundamental frequency of 400Hz.

The interval between larynx pulses is then 2.5ms. A 20ms analysis window corresponds, on average, to 8 larynx pulses (figure 9). The exact position of the window relative to any particular larynx pulse now becomes less important. However, any frequency analysis will detect the periodic structure due to the excitation signal from the larynx and this will show up as a low frequency component, plus harmonics, in the short-term spectrum. Thus the spectrum will no longer just show the vocal tract filter shape, and any change in fundamental frequency will lead to a change in the detail of the short-term spectrum. This is one reason why many complex speech systems work better for adult male speakers than for female speakers or children!

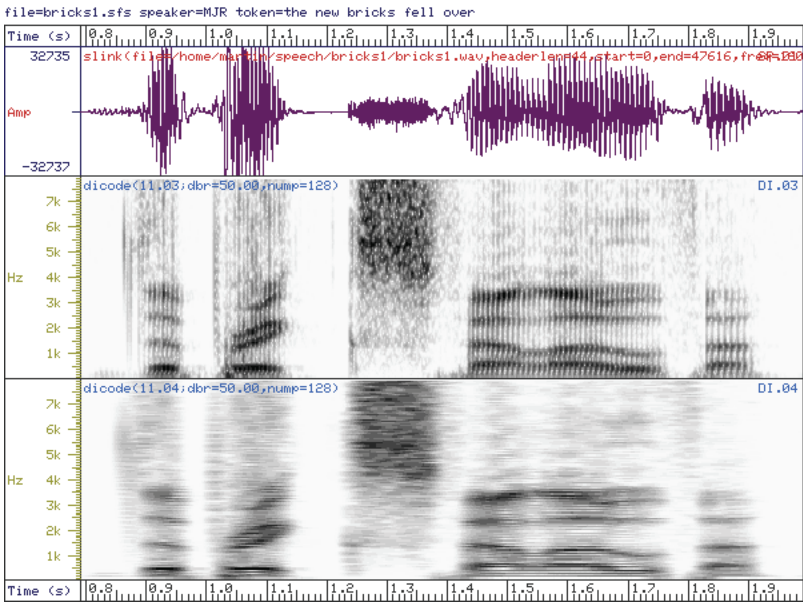


Figure 7: Broadband spectrogram (bandwidth 200Hz) of the sentence "The bricks fell over". The speech was sampled at 16,000 samples per second

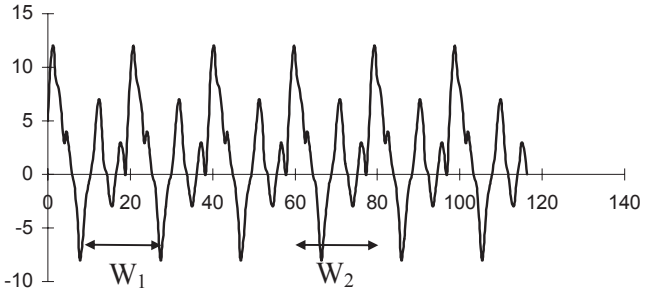


Figure 8: Schematic diagram of a speech waveform in a vowel sound for an adult male speaker with 50Hz fundamental frequency, showing alternative positions of 20ms short-term frequency analysis windows

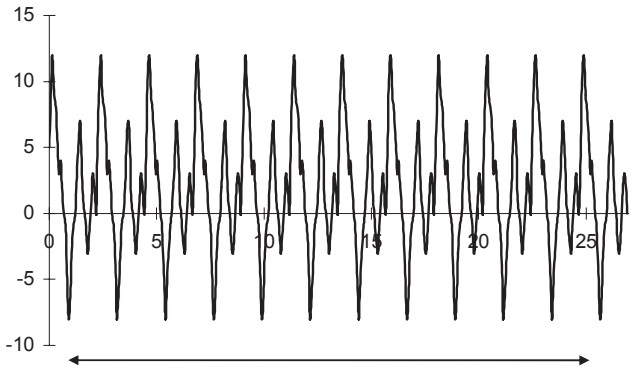


Figure 9: Schematic diagram of a speech waveform in a vowel sound for a female speaker with 400Hz fundamental frequency, showing a 20ms short-term frequency analysis window

3.12.3.1 Figure 10 shows wideband and narrow-band spectrograms of the sentence "The bricks fell over", spoken by an adult male. The pitch was raised artificially during the utterance of the vowel in the word 'bricks'. This change in pitch is clearly visible in the narrow-band spectrogram.

3.13 Estimation of the Fundamental Frequency

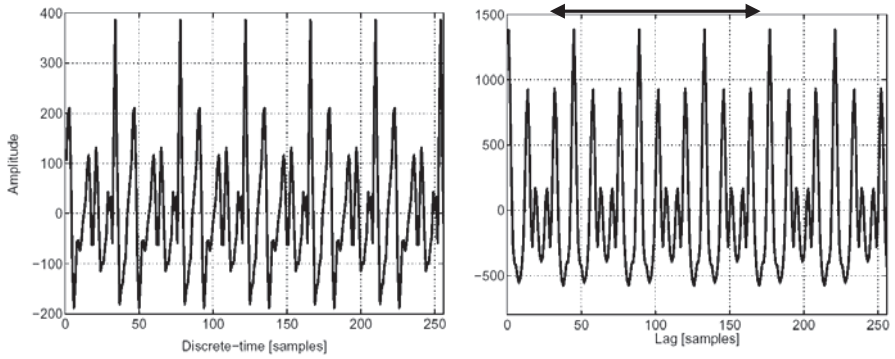
3.13.1 The information about fundamental frequency F0 is required in the analysis of speech sounds (e.g., variations of F0 contribute to prosody, and in tonal languages they help distinguish lexical categories), speech enhancement systems and analysis/synthesis coding of speech. The autocorrelation-based and cepstral-based methods are most widely used for F0 estimation. The former is described below.

3.13.2 The autocorrelation of a discrete signal *s(t)* is defined as:

$$r_t(\tau) = \sum_{j=t}^{t+N-1} s(j)s(j+\tau)$$

where $r_t(\tau)$ is the autocorrelation function (ACF) of lag τ calculated at time index t, and N is the window size. The lag τ is the shift between the signals.

3.13.3 For a periodic signal, the ACF shows peaks at multiples of the period of the signal. The autocorrelation method estimates the F0 by choosing the first lag (within a reasonable range) that gives highest peak in the ACF. Given the lag τ_{est}, the F0 is then estimated as: $F0_{est} = Fs/\tau_{est}$, where Fs is the sampling rate/frequency.



(a) (b)

Figure 3.13.1: Example of a speech waveform (a) and the corresponding autocorrelation function (b)

3.13.4 Example of a speech waveform, sampled at Fs=8kHz, and the corresponding autocorrelation function is depicted in Figure 3.13.1. Considering that the F0 is within the range from 50Hz to 400Hz, the corresponding range in the lag (depicted as horizontal arrow in the figure) is from 160 down to 20 samples, respectively. The highest peak in the autocorrelation function in the Figure 3.13.1(b) is at the lag value 44, giving the F0 estimate of approx. 182Hz.

3.13.5 The estimation of F0 typically suffers from halving and doubling errors. A possible way to reduce these errors is to employ some temporal continuity constraints of F0 estimates.

3.14 **Speech Bandwidth Revisited**

3.14.1 The following figure emphasises the claims about appropriate sampling rates for speech signals which were made earlier. Figure 11 shows a broad-band spectrogram of the sentence from figure 3. In this case the sample rate is 44,100 samples per second (CD quality). In fact, the speech waveform in figure 3 was obtained by downsampling the waveform in figure 11. The resulting spectrogram shows frequencies up to over 22kHz. The figure reinforces the claim that the important structure in speech signals is contained in the frequency range up to 10kHz. It also shows that most of the structure which is characteristic of vowel sounds is contained in the 'telephone bandwidth' up to 4kHz.

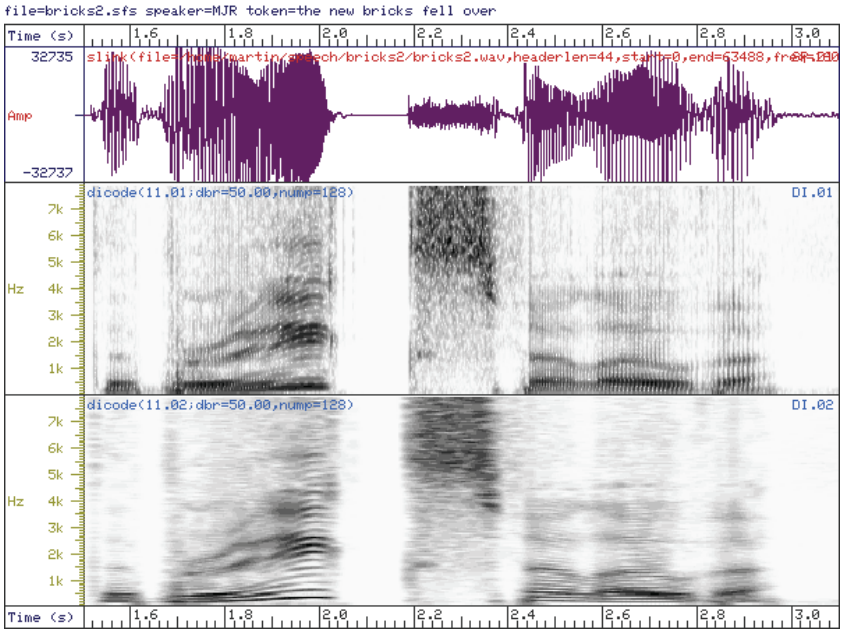


Figure 10: Broadband spectrogram (bandwidth 200Hz) of the sentence "The bricks fell over". The speech was sampled at 16,000 samples per second. The artificial raising of the pitch in the vowel in 'bricks' is clearly visible in the narrow-band spectrogram.

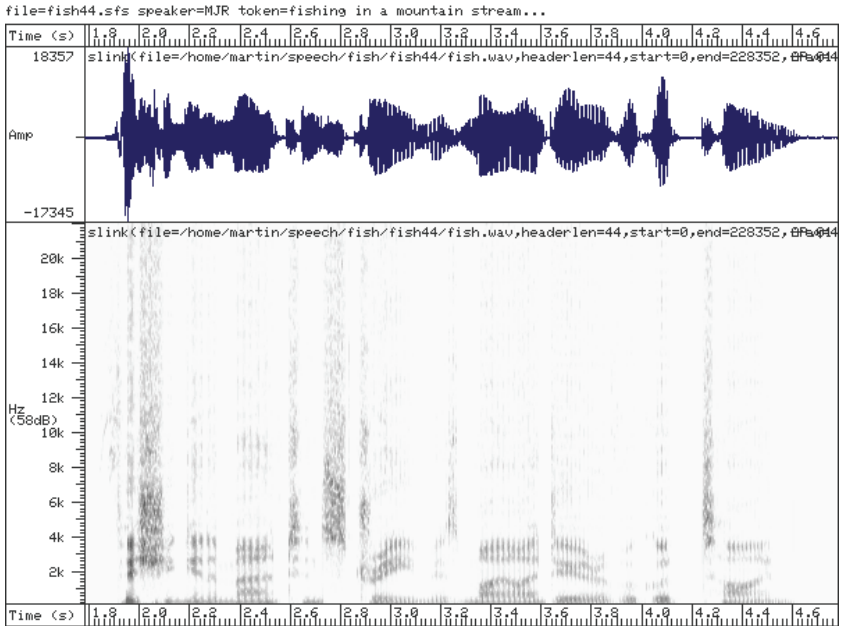


Figure 11: Broadband spectrogram of the sentence "Fishing in a mountain stream is my idea of a good time". The speech was sampled at 44,100 samples per second. The figure shows that most of the important striucture is contained in the frequency range up to 10kHz.

3.15 **Speech is Continuous**

3.15.1 While we are looking at speech spectrograms it is a good time to make the important point that we should not think of speech as 'acoustic text'. Speech is a continuous signal, produced through the (generally) continuous movement of the components of the vocal tract. There are a number of standard phrases which speech scientists use to illustrate this point. Figure 12 shows a broad-band spectrogram of such a phrase – "we were away a year ago". Note that there are no gaps between the words. The apparent gap between 3.32 and 3.4 seconds corresponds to the 'stop' in the 'g' sound in the middle of the word 'ago'.

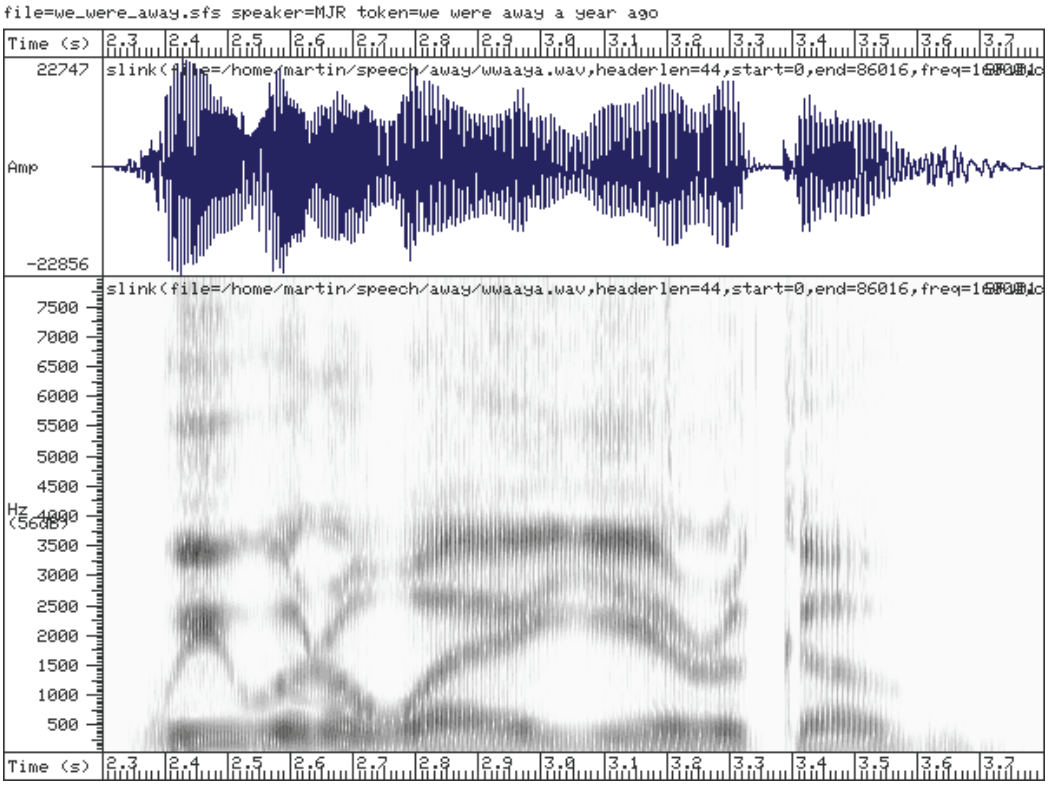


Figure 12: Broad-band speech spectrogram of the phrase "we were away a year ago", spoken by an adult male speaker. The sampling rate was 16,000 samples per second.

3.15.2 The articulators in the vocal tract move relatively slowly. Consequently the evolution of the articulators throughout the pronunciation of a word is influenced by the preceding and following words. This is called **co-articulation**. Similarly, prosodic effects like intonation, rate-of-speaking and amplitude change relatively slowly, and these factors will also be influenced by context. The interpretation of a sentence will be strongly influenced by details of the individual words which are used. Remember the

sentence "*Sam didn't buy Mary a pizza"*. The different realisation of each phoneme due to the context is referred to as **allophones.**

3.15.3 The central regions of many speech sounds are approximately stationary and less susceptible to coarticulation effects.

3.15.4 We have already seen that it is useful to think of the vocal tract as a sequence of cavities, each with its own resonant frequency, which modifies the shape of the spectrum of the excitation signal.

3.15.5 The resonant frequencies of the vocal-tract, i.e., formants, are visible as peaks in the spectrum. Experiments in psycho-acoustics have shown that the frequencies, amplitudes and bandwidths of the first three formants determine the phonetic properties of a speech sound and are of primary importance for speech perception. In general, formants are reasonably well defined for voiced sounds and less well defined for un-voiced sounds.

3.15.6 **Prosody** includes durational structure, the appropriate insertion of pauses, intonation and stress. The importance of this component of a speech signal should not be underestimated. We have already seen that it is possible to alter the meaning or the function of a sentence simply by changing the intonation pattern.

4 Automatic Speech Recognition

4.1 Why is speech recognition difficult?

4.1.1 Intuitively, meaning is represented by sentences, a sentence is a sequences of words, a word is a sequences of phonemes, …

4.1.2 This sequential model of speech is based on our experience of text, but speech is **not** just "*acoustic text"*.

4.1.3 For example, speech is:

- **Continuous :** "We were away a year ago"
- **Variable:** "bread and butter" or "brembudder"
- **Ambiguous:** "The grey tape can fix that leak" or "The great ape can fix that leek" or "The great ape can fix that league" or "The great tape can. Fix that'll eek!"

or, compare "recognise speech" with "wreck a nice beach"!

4.1.4 The high-level phonemic description of speech can also be misleading. Consider the words "league" and "leek". Phonemically, "*league"* is transcribed **/ l i g /** and *"leek"* is transcribed as **/ l i k /**. The difference appears to be in the final consonant, which is voiced (**/g/**) in "league" but unvoiced (**/k/**) in "leek". But in natural fluent speech, the **duration** of the vowel **/i/** may be a more important cue to recognition!

4.1.5 These types of effects just do **not** happen with text.

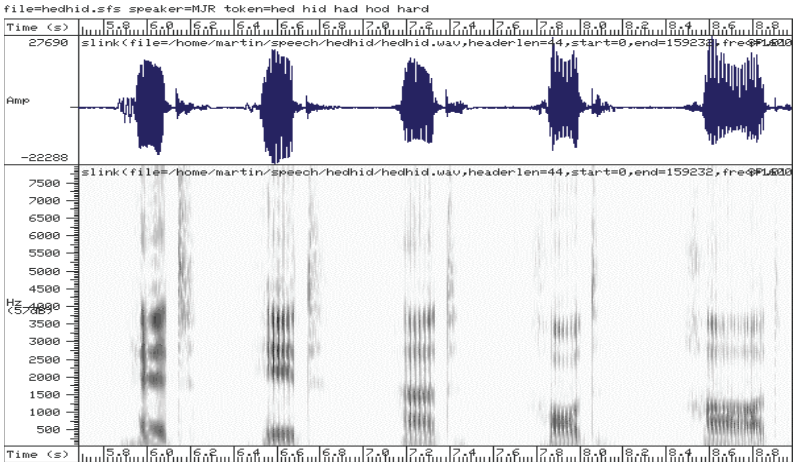


*Figure 25: Speech spectrograms for five English vowels, spoken in the context / **h - d/** - 'hed', hid', 'had', 'hod' and 'hard'. Speech sampled at 16,000 samples per second, adult male speaker*

4.2 **Structure of Vowels**

4.2.1 If you look at speech spectrograms corresponding to different vowels in the same, controlled context, then, for example in the formant positions, you observe clear consistency within vowel classes and differences between classes, (figure 25). This suggests that speech recognition shouldn't be too difficult, provided that we can accurately identify features such as formants! Unfortunately, this simplistic view fails on two accounts:

4.2.2 This apparently simple structure becomes much more complicated in fluent speech (figure 26)

4.2.3 Features such as formants are notoriously difficult to detect automatically!

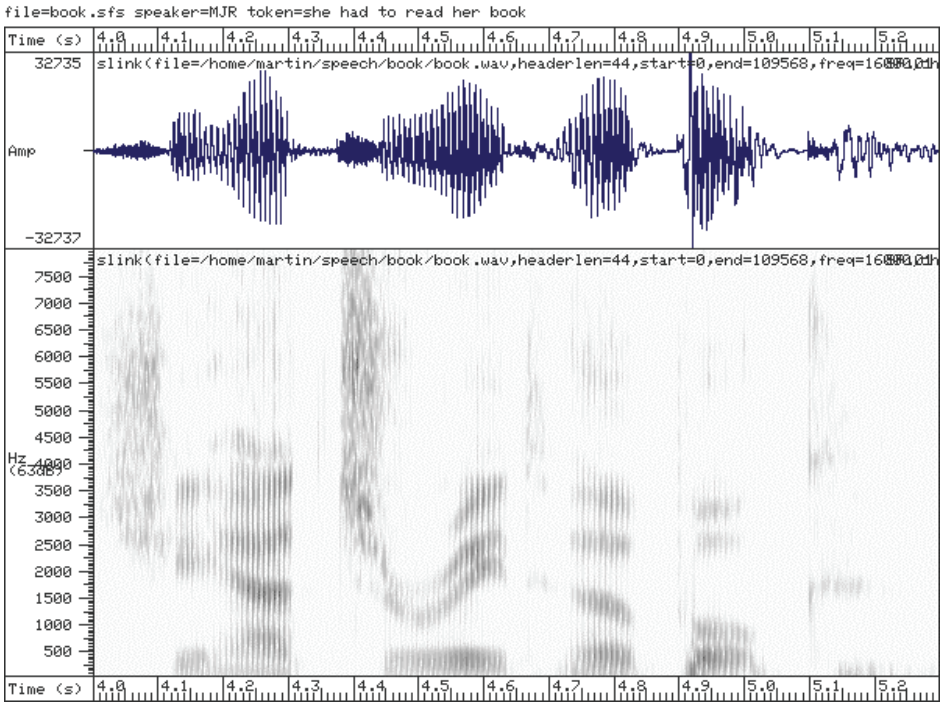


Figure 26: Spectrogram of the phrase "She had to read her book". Sample rate 16,000 samples per second, adult male speaker.

4.2.4 Many approaches to speech recognition have been tried in the past, including:

- Artificial Intelligence (AI)
- Artificial Neural Networks
- …

4.2.5 The use of **Artificial Intelligence (AI)** based methods was widespread in the 1970s. Researchers believed that there was insufficient information in the

acoustic data to recognise speech, and that additional sources of 'knowledge' were necessary. These additional knowledge sources included acoustic-phonetic, lexical (words), syntactic (grammar), semantic and domain-specific knowledge. In fact, some of these systems paid very little attention to the acoustic waveform at all!

4.2.6 There is a famous story about speech controlled system for playing chess. It paid little attention to the actual acoustic signal and relied more heavily on domain knowledge - in this case knowledge about chess. To win at chess the user simply needed to cough! The acoustic signal would then be ambiguous, and hence ignored, and the system would rely almost entirely on its knowledge of chess - and assume that the best move had been requested! If you're interested in the AI approach, look at Newell (1978).

4.2.7 By the end of the 1970s, AI-based systems had been outperformed by systems based on simple pattern matching techniques. In fact, the main lasting influence of these systems is that they pioneered some of the basic techniques in **Integrated Knowledge Based Systems (IKBS)**. Indeed, the **HEARSAY II** system is the original 'blackboard model'

4.2.8 By far the most successful approach to date is based on **statistical modelling**, and in particular **hidden Markov models (HMMs).** This is the basis of all state-of-the-art commercial (and most laboratory) speech recognition systems.

4.2.9 Before we look in detail at hidden Markov models, lets fix some **terminology**.

4.3 Speech Recognition Terminology

4.3.1 The basic problem in speech recognition is **variability**. We need to be able to ignore substantial irrelevant variability, but be sensitive to small significant differences. Early attempts at speech recognition tried to overcome the variability problem by **removing** it.

4.3.2 **Speaker-dependent** speech recognition systems avoid inter-speaker variability by training on, and subsequently recognising, a single speaker

4.3.3 **Multiple-speaker** systems work for a particular **population** of speakers

4.3.4 **Speaker Independent** systems work for **any speaker**, with no implicit or explicit training. Note that speaker-independent speech recognition is probably beyond the capabilities of many human listeners!

4.3.5 **Speaker adaptive** systems automatically adapt to a new speaker. For example, one might begin with a speaker-independent system, and then adapt the system to a particular speaker to obtain a speaker-dependent system. This is the goal of much current research.

4.3.6 Another source of variability is coarticulation between words. **Isolated word** recognition systems overcome this problem by requiring the user to leave gaps between words (you saw this on the Dragon-Dictate video).

4.3.7 **Connected speech recognition** systems recognise isolated phrases or sentences.

4.3.8 **Continuous speech recognition** systems recognise continuous speech.

4.3.9 Another important issue is vocabulary size. **Small vocabulary** systems work with vocabularies of *10-100* words. Medium vocabularies process around *100* to *5,000* words. **Large Vocabulary Continuous Speech Recognition (LVCSR)** systems can cope with 60,000 words, while **unlimited vocabulary** systems have no vocabulary size limitation.

4.3.10 In the next sections we shall see how a state-of-the-art hidden Markov model (HMM) based speech recognition system works.

4.4 Front-end processing

4.4.1 Front-end processing is the first stage in any automatic speech recognition system. Its goal is to convert the speech waveform into a representation which is suitable for recognition. From the perspective of general pattern recognition, front-end analysis is **feature extraction**.

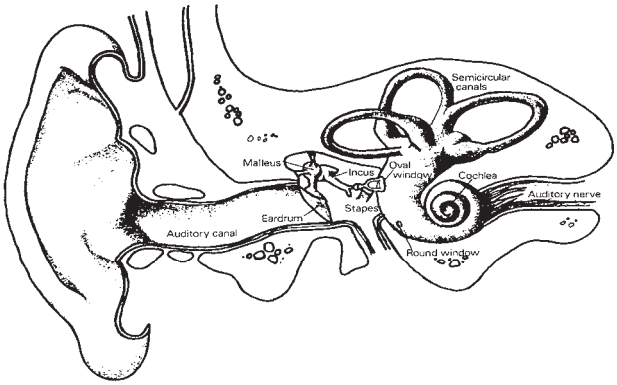


Figure FE1: The human peripheral auditory system. Taken from J N Holmes, "Speech Synthesis and Recognition", Van Nostrand Reinhold (1988)

4.4.2 Most approaches to front-end processing are inspired by knowledge of the human auditory system (figure FE 1).

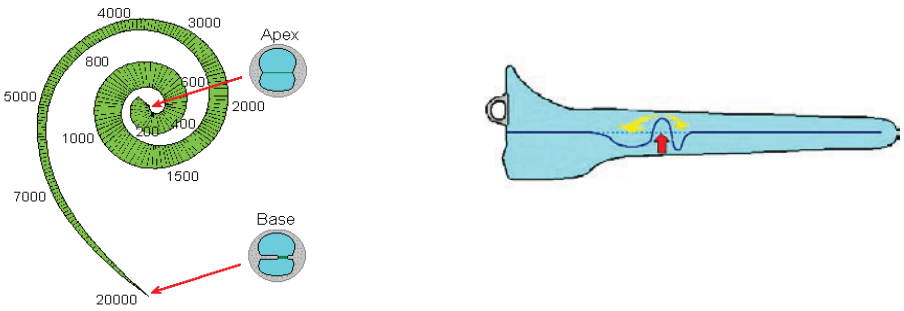
4.4.3 Any acoustic signal, including speech, will cause the eardrum to vibrate. This in turn causes vibration of the oval window, which is part of a spiral shaped organ called the **cochlea**. Running along the length of the cochlea is a membrane, called the **basilar membrane**, which vibrates according to the

frequency content of the signal (figure FE2). The following properties of this analysis are typically exploited in front-end analysis for speech recognition:

- An individual point on the basilar membrane can be modeled as a band-pass filter.
- Frequency is not perceived on a linear scale but on a non-linear mel scale
- Loudness perceived on logarithmic scale
- Phase is of limited significance for speech recognition

4.5 **Frequency analysis**

4.5.1 The front-end analysis process typically begins with low-pass filtering the speech at somewhere between 4kHz and 8kHz. The speech is then sampled at between 8,000 and 16,000 samples per second (depending on the cut off frequency of the low pass filter). Some form of frequency analysis is then performed. For example, the speech is divided into short segments, typically 20-30ms with a 10ms overlap between adjacent segments. A Hamming window is then applied to the segment, followed by a Discrete Fourier Transform



(a) (b)

Figure FE2: Frequency response of the basilar membrane (a) and basilar membrane dynamics (b). (School for advanced studies, Triste, http::/poirot.sissa.it/multidisc/cochlea/utils/basilar.htm)

4.5.2 The underlying assumption here is that the speech spectrum remains approximately constant over an interval of 20-30ms. Since the shape of the spectrum reflects the shape of the vocal tract, this is actually an assumption about vocal tract dynamics (we made the same assumption in the section on speech coding).

4.6 **The log-power spectrum**

4.6.1 In the next stage, the **modulus** is taken of each point in the complex discrete spectrum. Since the complex components of the Fourier coefficients encode the **phase**, this has the effect of removing phase information from the acoustic representation (which is consistent with the belief that phase is not important for recognition). Next the logarithm is applied to each Fourier coefficient, for consistency with psycho-acoustic experiments on perception of loudness and, for expediency, to compress the dynamic range. The result at this stage is that each 20ms section of speech is represented by a **discrete (*M* point) log power spectrum** $s = (s(1),\ldots,s(M))$

4.7 The mel frequency scale

4.7.1 The next stage in our typical front-end processing scheme takes account of the non-linear perceptual frequency scale. The **mel** scale (figure FE3) is a **perceptually** relevant frequency scale based on psycho-acoustic experiments on human hearing. We have already discussed the mel scale, and the fact that perceptual notions, such as **pitch**, do not necessarily correspond directly to measurable quantities, such as **frequency**. From the perspective of speech recognition, the main consequence of this result is that spectral differences are not equally significant at all frequencies. It is clear from figure FE3 that, for example, differences in the location of a peak in the spectrum are more significant at lower frequencies than at higher frequencies. However, the process whereby this effect is achieved also achieves several other goals, which are discussed below.

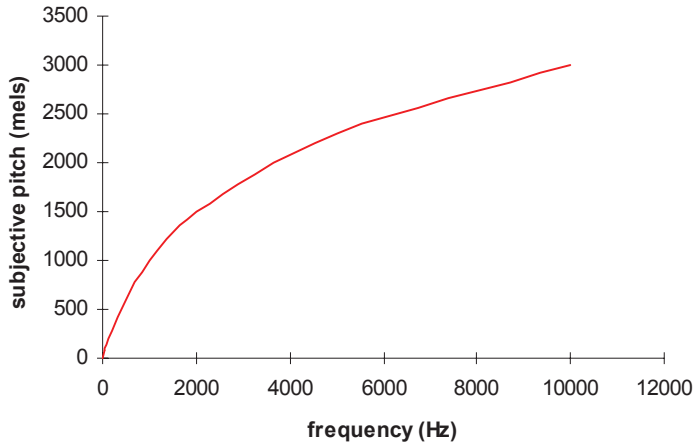


Figure FE3: The mel perceptual frequency scale

4.8 Critical bands

4.8.1 If we think of the frequency response of the basilar membrane at a particular point as a bandpass filter, then the bandwidth of that filter is a **critical band**. Consider the following experiment. A subject is played a tone at a particular frequency *f* together with band-limited white noise, such that the noise band

is centred around the tone. The subject is then asked to adjust the volume of the tone so that it is just audible. In this way it is possible to obtain an estimate of subjective signal-to-noise-ratio. The experiment is repeated for different noise bandwidths. The results are shown schematically in figure FE4.

4.8.2 The figure shows that the perceived SNR is constant for noise bandwidth greater than a bandwidth W_f, but that as the noise bandwidth is reduced below W_f the perceived SNR increases. W_f is called the **critical bandwidth** at f.

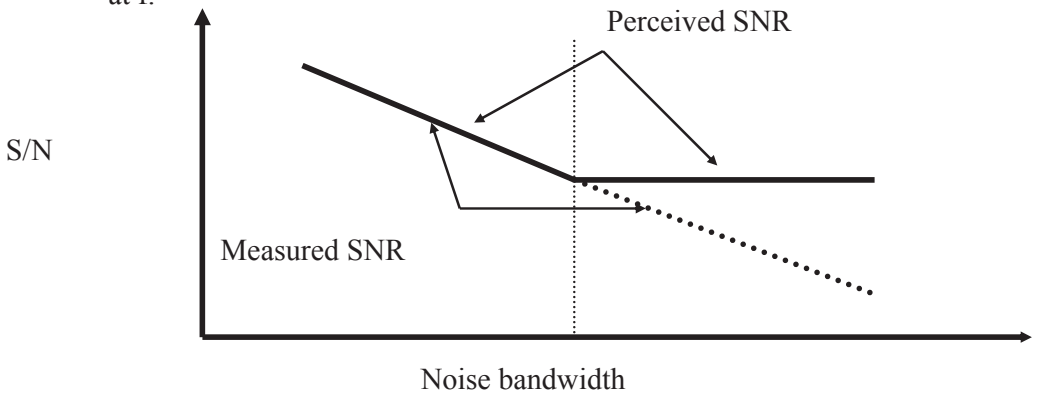


Figure FE4: Schematic graph showing typical results of the 'critical band' estimation experiment described in the text.

4.8.3 In our typical front-end analysis scheme, the effects of mel-scale frequency conversion and critical bandwidths are implemented together. A finite number *N* of frequencies, spread linearly on the mel scale (but non-linearly on the Hertz scale) are chosen. For each frequency f_n, a triangular, critical-band 'filter' is constructed with centre frequency f_n. An example set of triangular filters is illustrated in figure FE5.

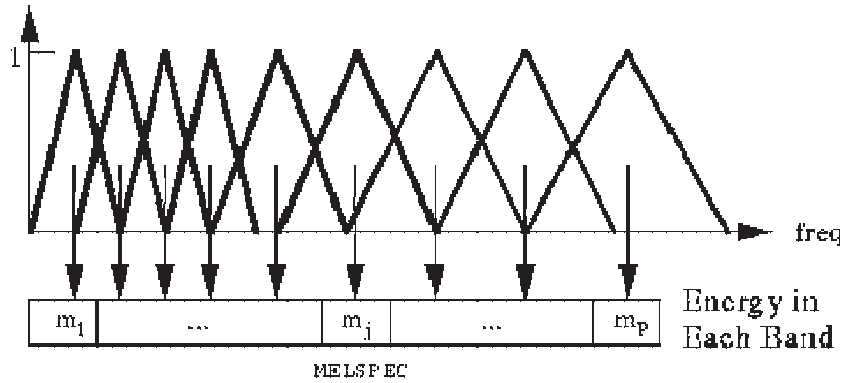


Fig. 5.3 Mel-Scale Filter Bank

Figure FE5: Mel-scale triangular filter bank (from Steve Young, "The HTK Book", Cambridge University Engineering Department)

4.8.4 Suppose the filter centred at f_n is denoted by T_n. The **(*N* point) mel scale log power spectrum** $g = (g(1),....,g(N))$ is defined by

$$g(n) = \sum_{m=1}^{M} s(m) T_n(m)$$

4.9 Smoothing

4.9.1 The conversion of the log power spectrum into the mel scale log power spectrum is a type of smoothing operation. One of the additional benefits of this process is that it removes possible noise in the spectrum due to effects of the excitation signal. These effects were discussed in an earlier section.

4.10 The Cosine Transform

4.10.1 Next, the mel scale log power spectrum is transformed using a cosine transform to obtain the mel frequency cepstrum. The resulting representation is often referred to as a set of mel frequency cepstral coefficients (or sometimes mel frequency cosine coefficients). In either case the relevant acronym is MFCC. The cosine transform has two effects. In principle, as demonstrated earlier, it can be used to remove the effects of the excitation signal on the spectrum. However, this may already have been achieved as part of the mel frequency conversion process described above. A second, and potentially more important effect, is that the cosine transform results in a representation in which much of the correlation between the different components of an individual acoustic vector has been removed. This significantly simplifies the mathematics and computation in speech recognition.

4.10.2 Formally, the mel frequency cepstrum $c=(c(1),\ldots,c(N))$ is obtained from the mel scale log power spectrum $g=(g(1),\ldots,g(N))$ by applying the cosine transform:

$$c(n) = \sqrt{\frac{2}{M}} \sum_{m=1}^{M} g(m) \cos\left(\frac{\pi n}{M}(m - 0.5)\right)$$

4.10.3 Just as the Fourier transform was used to decompose the speech waveform into a sum of weighted sine functions, the cosine transform decomposes the spectrum into a **weighted sum of cosine functions**.

4.10.4 A moments thought will confirm that $c(0)$ is proportional to the average log power spectrum component value. Figure FE6 shows the cosine functions corresponding to cosine coefficients c_0 to c_4.

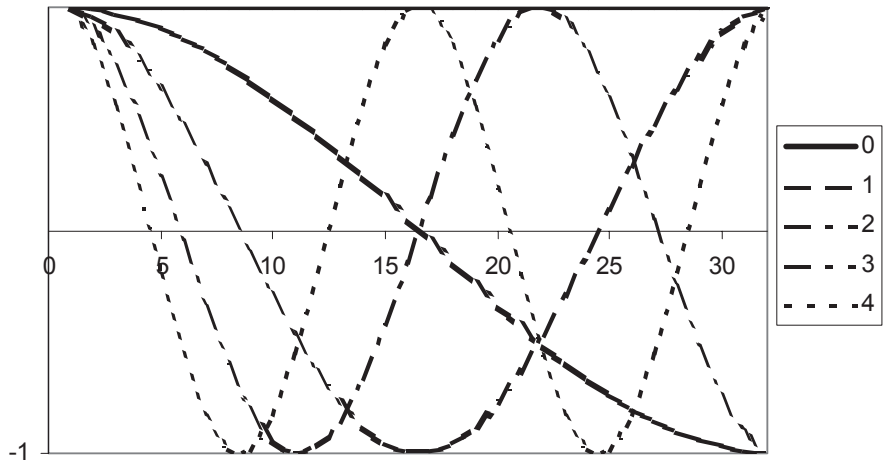


Figure FE6: Cosine functions corresponding to cosine coefficients c_0 to c_4.

4.11 'Delta' and 'Delta^2' parameters

4.11.1 We now have a parameterisation of a speech signal as a sequence of mel frequency cepstral vectors, approximately one vector every 10ms. Although these vectors give instantaneous information about the shape of the vocal tract and the type of speech sound, they contain no information about speech dynamics – the way in which the speech signal is changing.

4.11.2 The simplest remedy is to supplement these **static** parameters with difference and second difference parameters which approximate the velocity and acceleration of each of the cepstral parameters. More precisely, if $c_t = c_t(1),\ldots,c_t(N)$ are the cepstral parameters at time t then we define the **delta** parameters Δc_t at time t by

$$\Delta c_t(n) = c_{t-D}(n) - c_{t+D}(n)$$

and the **delta-delta** parameters $\Delta^2 c_t$ by

$$\Delta^2 c_t(n) = \Delta c_{t-D}(n) - \Delta c_{t+D}(n).$$

4.11.3 The final parameterisation at time t is the $3N + 3$ dimensional vector y_t defined by:

$$y_t = (c_t(0),\ldots,c_t(N), \Delta c_t(0),\ldots, \Delta c_t(N), \Delta^2 c_t(0),\ldots, \Delta^2 c_t(N))$$

4.11.4 A typical value of N is 12.

4.12 Summary of Front-End Speech Signal Processing

4.12.1 In summary, the stages involved in a typical front-end signal processing scheme for automatic speech recognition, starting with a sampled waveform, are as follows:

4.12.1.1 Segment waveform into segments of 20-30ms overlapping by approximately 10ms. Consider the segment centred at time t.

4.12.1.2 Apply a Hamming window to this segment

4.12.1.3 Apply a Discrete Fourier Transform to obtain a set of M complex spectral coefficients

4.12.1.4 Take the modulus and logarithm of each coefficient to get the log power spectrum $(s_t(1),\ldots,s_t(M))$

4.12.1.5 Apply mel frequency averaging to obtain a set of N mel frequency log power spectrum coefficients $g_t=(g_t(1),\ldots,g_t(N))$ $(N < M)$

4.12.1.6 Apply a discrete cosine transform to obtain a set of N **mel frequency cepstral coefficients** $c_t = (c_t(1),\ldots,c_t(N))$. At this point the mel frequency cepstral coefficients can be truncated by discarding the upper coefficients

4.12.1.7 Compute the difference and second difference coefficients $\Delta c_t(n) = c_{t-D}(n) - c_{t+D}(n)$ and $\Delta^2 c_t(n) = \Delta c_{t-D}(n) - \Delta c_{t+D}(n)$

4.12.1.8 Form the concatenated feature vector

$$y_t = (c_t(0),\ldots,c_t(N), \Delta c_t(0),\ldots, \Delta c_t(N), \Delta^2 c_t(0),\ldots, \Delta^2 c_t(N))$$

4.12.2 This is the feature vector which is used to describe the speech signal at time t in recognition

5 Hidden Markov Models

5.1 Mathematical Modelling for Speech Recognition

5.1.1 As in most areas of mathematical modelling, in the case of speech recognition there are two conflicting requirements (figure 27). On the one hand one would like a **faithful model of human speech production/perception**. Against this, we want the model to be **mathematically tractable** and **computationally useful**. Hidden Markov Models (HMMs) are the best compromise at the moment.

Figure 27: Hidden Markov models are a compromise between mathematical tractability and computational usefulness, and faithful modelling of human speech recognition.

5.2 Delayed Decision Making

5.2.1 Before we go on to discuss the details of Hidden Markov Models, it is useful to mention some of the basic principle which underlie the HMM-based approach to speech recognition. One of the most important of these is the principle of **delayed decision making**.

5.2.2 One possible approach to automatic speech recognition might be referred to as sequential **divide and conquer**. According to this scheme we would take an approach to the problem along the following lines:

- classify speech vectors as 'acoustic features'
- classify sequences of acoustic features as phonemes
- classify sequences of phonemes as words
- classify sequences of words ...

5.2.3 Another name for this might be **non-recoverable error propagation!** By formulating the speech recognition problem in this sequential way, and by making hard decision at each stage of the sequence, classification errors will be introduced at each level which cannot be recovered in the next level. It is better to avoid all classification decisions until all sources of information are available, and then to perform classification as a single, integrated process - this is the principle of **delayed decision making**.

5.2.4 Delayed decision making is one of the key reasons for the success of the HMM-based approach to automatic speech recognition.

5.3 The 'HMM Compromise'

5.3.1 Before going into the detail of HMMs, let us look at the assumptions which we are about to make about speech patterns. Broadly speaking, the assumptions are as follows.

5.4 HMM ssumptions:

5.4.1 A spoken utterance is a time-varying sequence which moves through a sequence of **segments**

5.4.2 The underlying structure of these segments is c**onstant** with respect to time

5.4.3 Transitions between these segments are **instantaneous**

5.4.4 The **durations** of the segments vary

5.4.5 All variations between different realizations of the segments are **random**

5.4.6 Assumption 7.4.1 is a reasonable first assumption. It is consistent with a traditional view of speech pattern structure, but at odds with modern theories based on 'non-linear phonology' which think of speech in terms of parallel asynchronous processes. From this modern perspective, assumption 1 might be referred to as a **'beads-on-a-string'** model of speech.

5.4.7 Assumption 7.4.2 is the source of one of the major criticisms of HMM-based approaches to speech pattern modelling and is clearly wrong. Speech is produced by a continuously moving physical system (the human vocal tract) and is, in general, a continuously changing signal. Any inspection of a speech pattern will confirm this. Assumption 7.4.2 is made for mathematical tractability, as is assumption 7.4.3.

5.4.8 Assumption 7.7.4 is OK. If one accepts a model of speech in terms of a sequence of segments, then the durations of these segments will certainly vary. However, we shall see later that the actual model of speech segment duration in a HMM is not ideal.

5.4.9 Assumption 7.4.5 is another major source of speech science's unhappiness with HMMs. It is a direct consequence of the need for mathematical tractability. A standard technique in mathematical modelling is treat variation which cannot be explained directly by the mechanisms inherent in the model as **random**. Hence, for example, in the case of HMMs it is assumed that the segments which constitute a speech pattern are constant. The fact that they are not is treated as random variation around an underlying constant theme.

5.5 Markov Models

5.5.1 One of the factors which distinguishes speech pattern processing from other areas of pattern processing is the need to accommodate the essential **time-varying** nature of speech patterns. Currently the most popular mathematical model of temporal structure in speech is a **Markov model**.

5.5.2 Formally, a (finite state) Markov model M consists of:

- A set of **states** $S = \{\sigma_1,...,\sigma_N\}$
- A **state transition probability** matrix $A = [a_{ij}]_{i,j=1,...,N}$, where
- $a_{ij} = \text{Prob}(x_t = \sigma_j \mid x_{t-1} = \sigma_i)$
- An **initial state** probability vector $\pi = [\pi_1,...,\pi_N]$, where $\pi_i = Prob(x_1=\sigma_i)$

5.5.3 Thus a_{ij} is the probability that the model is 'in' state j at time t given that the model was 'in' state i at time $t-1$. Notice that t is a dummy variable in this definition, and that the probability a_{ij}, and hence the matrix A, does not depend on t. A Markov model which has this property is called a **time-homogeneous** Markov model.

5.5.4 Notice that the matrix A has the property $\sum_{j=1}^{N} a_{ij} = 1$. Such a matrix is called **row stochastic**.

5.5.5 The **Markov property** states that the state of the process at time $t+1$ depends on the state at time t (but is independent of the history of the process before time t).

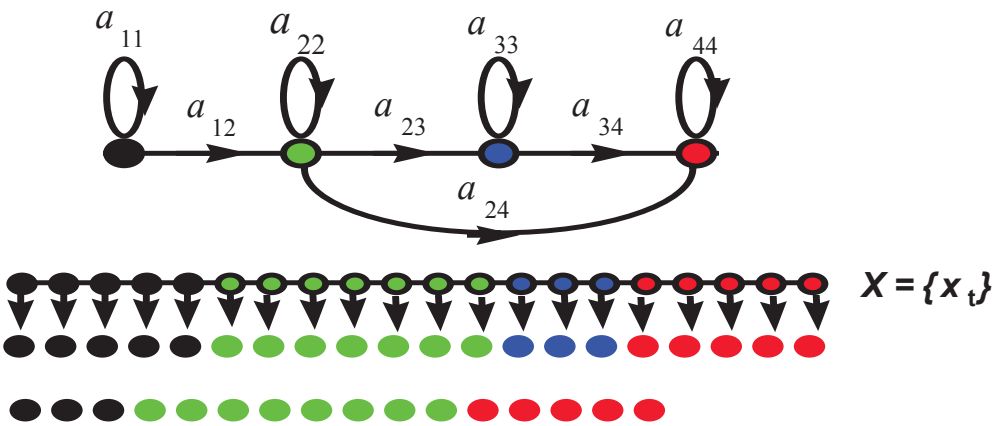


Figure 28: A schematic diagram of a simple Markov model, showing examples of the type of sequence which it can generate.

5.5.6 A finite state Markov model can be conveniently represented as a **state transition network**, as shown in figure 28. The convention when drawing such a diagram is that a transition from state *i* to state *j* is only drawn in the diagram if $a_{ij} > 0$. The figure shows a **left-right** Markov model, in which $a_{ij} = 0$ whenever $j < i$. This is the type of model which is used most commonly, but not exclusively, in speech pattern modelling.

5.5.7 It is sometimes convenient to think of a Markov model as a **generative** model (i.e. as a machine which generates random sequences). Examples of two sequences generated by the model are shown in the figure. The mechanism for generating such sequences is as follows:

5.5.7.1 At time $t=1$ the model is in state $x_1 = \sigma_n$ where n is determined randomly according to the initial state probability vector π. The n^{th} symbol (i.e. the symbol associated with state σ_n) is generated.

5.5.7.2 Suppose that at time $t-1$ the model is in state x_{t-1}. At time t the model moves to state x_t, where x_t is determined randomly according to the probability vector $A_t = [a_{xt,1},...,a_{xtN}]$, and the symbol corresponding to state x_t is generated.

5.5.7.3 And so on.

5.5.8 The sequence of observations generated in this way is called a **Markov Chain**.

5.6 Hidden Markov Models

5.6.1 A Markov model is a useful model of sequences whose elements are drawn from a small, finite, alphabet. However, in general a speech pattern is represented as a sequence $y = y_1,...,y_T$ where each y_t is an element of an infinite, multi-dimensional acoustic vector space. Even if the y_ts are drawn from a large, finite space, the resulting transition probability matrix may be so large that estimation of its parameters is impractical. Hence a more powerful model is required which retains the ability of a Markov model to characterise time-varying sequences. A **Hidden Markov Model (HMM)** is such a model.

5.6.2 Formally, a finite state HMM consists of

- A set of **states** $S = \{\sigma_1,...,\sigma_N\}$
- A state transition probability matrix $A = [a_{ij}]_{i,j=1,...,N}$, where a_{ij} =Prob(σ_j at time t | σ_i at time $t-1$)
- An **initial state probability vector** $\pi = [\pi_1,...,\pi_N]$, where $\pi_i = Prob(x_1=\sigma_i)$

– For each state σ_i, a **probability density function** b_i defined on the set of possible observations Y s.t. $b_i(o) = \text{Prob}(y_t=o \mid x_t=\sigma_i)$

5.6.3 The first three conditions are simply the definition of a Markov model. In the context of a HMM, this is called the **underlying Markov model**. The additional structure which characterises a **hidden** Markov model is property 4. The probability density function (PDF) b_i is called the **state output PDF** for state i (or the $\boldsymbol{i}^{\textbf{th}}$ **state output PDF**).

5.6.4 So, rather than associating a state of the underlying Markov model with a single element of the acoustic vector space Y, it is associated with a distribution over Y. Intuitively, if we think of a state as corresponding to a basic 'sound' within an utterance, then the function of the state output PDF is to model variations in the acoustic realisation of that state.

5.6.5 As with a Markov model, it is sometimes useful to think of a HMM as a generative model. Examples of two sequences generated by a HMM are shown in figure 29. The mechanism for generating such sequences is as follows:

5.6.6 At time $t=1$ the model is in state $x_1 = \sigma_n$ where n is determined randomly according to the initial state probability vector π. An element of the acoustic vector space Y is then generated randomly according to the state output PDF b_n.

5.6.7 Now suppose that at time $t-1$ the model is in state x_{t-1}. At time t the model moves to state x_t, where x_t is determined randomly according to the probability vector $A_t = [a_{xt,1},\ldots,a_{xtN}]$. An element y_t from the acoustic vector space Y is then generated randomly according to the state output PDF b_{xt}.

5.6.8 The process continues.

5.6.9 The essential point is that at any time t, a HMM can generate **any** acoustic vector y_t from the acoustic vector space Y.

5.6.10 Now that we have a formal definition of a HMM it is a good time to revisit the assumptions which such a model makes about the structure of speech patterns. In more detail, they are as follows:

- The **Temporal Independence** assumption - the observation y_t depends on the state x_t but is otherwise independent of the rest of the observation sequence $y = \{y_t\}$
- ... so, the position of the vocal tract at time t is independent of its position at time $t-1$
- The **Piecewise stationarity** assumption - the underlying structure of speech is a sequence of stationary segments with instantaneous transitions between them
- The **Random variability** assumption - variations from this underlying structure are random

5.7 Modelling Duration in a HMM

5.7.1 Duration is clearly a key aspect of speech pattern structure. The model of speech pattern duration in a HMM is based on a model of **state** duration.

5.7.2 Given a state σ_j the probability that a transition from σ_j to itself occurs between times t and $t+1$ is given by a_{jj}. Thus, if the model is in state σ_j at time t, the probability of remaining in state σ_j for another time period is a_{jj}. Notice that this probability is independent of the time that the process has already spent in state σ_j. Thus the probability that the process remains in state σ_j for precisely D time units is the probability of $D-1$ successive transitions from state σ_j to itself, followed by a transition to any other state. If we denote this probability by $p_j(D)$ then:

$$p_j(D) = a_{jj}^{(D-1)}(1-a_{jj}).$$

5.7.3 Such distribution is called a **geometric distribution** and is shown in figure 30.

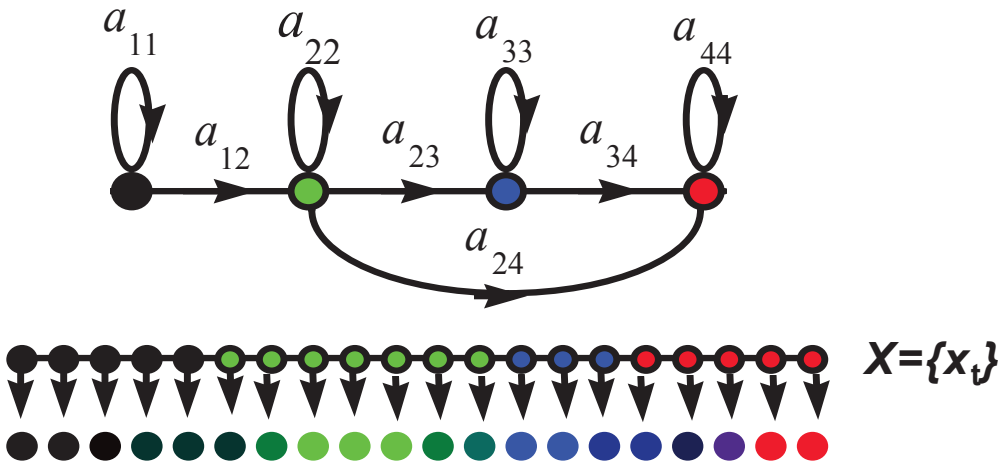


Figure 29: A schematic diagram of a simple hidden Markov model (HMM), showing an example of the type of sequence which it can generate.

5.7.4 Clearly such a distribution is an inappropriate model of speech segment duration. However, it is a direct consequence of the Markov assumption.

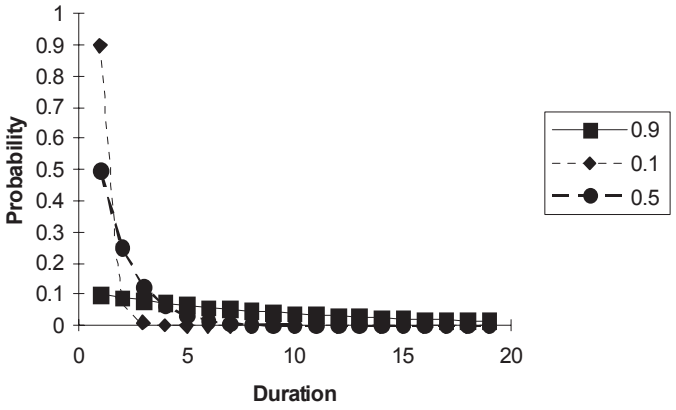


Figure 30: Geometric state duration PDFs for a_{ij} = 0.9, 0.1 and 0.5

5.8 Types of Hidden Markov Model

5.8.1 In this section we shall look at the common types of HMM which are currently in use in speech processing. Figure 31 shows a 'first level' taxonomy which splits general HMMs into three groups.

5.8.2 The left-hand box contains extensions of HMMs in which the state output PDFs are defined for sequences of acoustic feature vectors, rather than individual feature vectors as in a conventional HMM. Thus the left-hand box includes **hidden semi-Markov models**, in which an underlying semi-Markov model is used in order to obtain an improved model of state duration, and **segmental hidden Markov models** in which sequences of acoustic vectors, or segments, are modelled as homogeneous units as a means to achieve an improved model of speech dynamics. These models are the subject of current research, and their detailed description is beyond the scope of this course.

5.8.3 The right-hand box contains systems which attempt to combine the best features of HMMs and **Artificial Neural Networks (ANNs)**. These systems originate from the mid 1980s, when ANNs were first applied to speech pattern processing. The motivation for using ANNs is that their training procedures are concerned with **discrimination**, rather than **modelling**. Early experiments were indeed able to demonstrate that ANNs were able to focus onto the subtle differences which distinguish certain types of speech sound. However, these early systems treated whole-word spectrograms as static patterns and were unable to deal effectively with the time-varying nature of speech signals. The need to deal with temporal variability led researchers to attempt to combine the discriminative advantages of ANNs with the

temporal processing advantages of Markov models. This resulted in the development of a number of different types of **hybrid HMM/ANN** systems. Perhaps the most notable ANN based approach to speech pattern processing is the **ABBOTT** system, developed by Tony Robinson at Cambridge University Engineering Department.

5.8.4 The middle box in figure 27 contains the mainstream conventional HMMs, and these are the subject of the remainder of this section.

5.9 Types of Conventional HMM

5.9.1 Figure 32 shows a taxonomy of conventional HMMs.

5.9.2 The main distinction at this level is between **discrete HMMs** and **continuous HMMs**. It is important to realise straight away that the words **discrete** and **continuous** do not refer the treatment of time (which is discrete in both cases) or to the state space (which is finite, and hence also discrete, in both cases). Instead, the terms discrete and continuous refer to the nature of the observation space *Y*, and hence to the nature of the state output PDFs which are needed to define probabilities over this space.

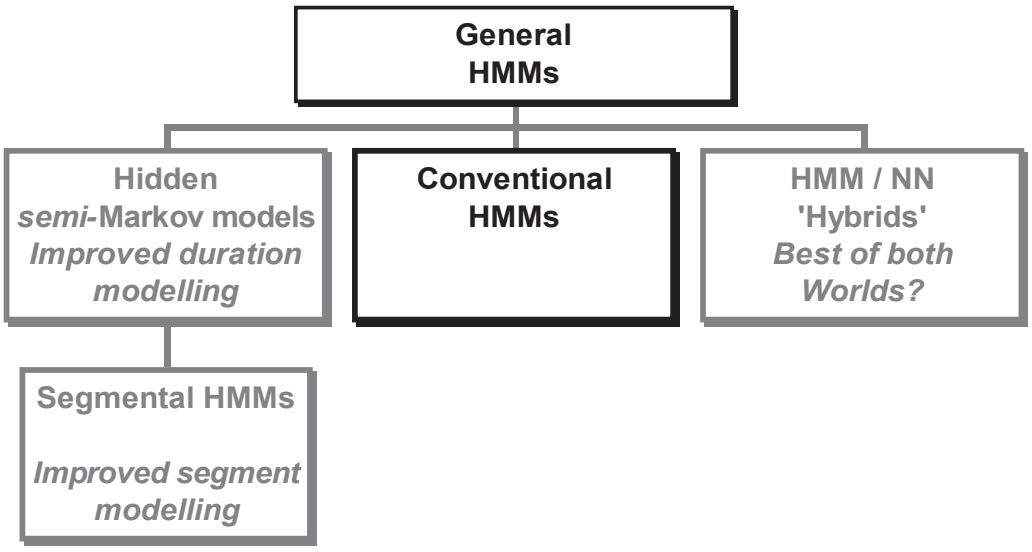


Figure 31: A taxonomy of types of HMM

5.10 Discrete HMMs

5.10.1 In the case of a **discrete HMM** the observation space *Y* consists of a **finite** number of elements $e_1,\ldots,e_M$. In this case each state output PDF b_i is simply required to specify the probability

$$b_{jm} = b_j(e_m) = \text{Prob}(y_t = e_m \mid x_t = \sigma_j)$$

for each element e_m. Thus the state output probabilities are simply defined by an $N \times M$ row stochastic state output probability matrix

$B = [b_{jm}]_{j=1,\ldots,N;\ m=1,\ldots,M}$.

5.10.2 Clearly the main obstacle to using discrete HMMs is the requirement that the acoustic observation space Y is finite. This is normally achieved through the application of **Vector Quantisation (VQ)** to the original acoustic feature vector space (remember this from speech coding). In VQ an acoustic feature vector is replaced (quantised) by substituting it by the closest element from a finite **codebook** of vectors which has been carefully crafted to span that part of the acoustic vector space which is inhabited by feature vectors which actually arise in speech patterns. This involves some form of distance measure between the acoustic vector and the set of codebook entries.

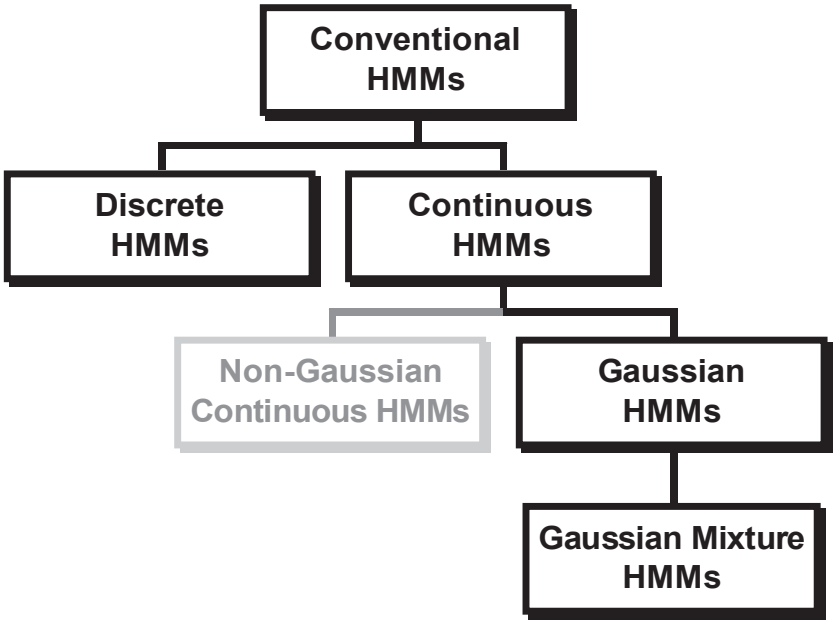


Figure 32: A taxonomy of conventional HMMs

5.10.3 The codebook is normally derived using some form of **clustering** process which attempts to define a set of cluster centroids so that, for example, the sum of the distances of acoustic vectors in the training set from the nearest VQ centroid is minimised. This sum is often referred to as the VQ codebook **distortion**.

5.10.4 Discrete HMMs were 'popularised' in Levinson's 1983 Bell System Technical Journal paper, though they had already been in use for some time by 1983. They were frequently used in speech recognition systems in the 1980s. Their main advantage is computational, since the process of calculating the state output probabilities is reduced to looking-up values in a table (we shall see in the next sections that other approaches are more

computationally demanding). It was also argued that the non-parametric form of the state output PDF meant that arbitrary shaped distributions could be employed and there was no requirement for the assumption, for example, that the state output PDF was Gaussian. However, I have always thought that this argument is flawed, since the construction of the VQ codebook involves some form of distance function, (for example a Euclidean distance) and the assumptions which are implicit in the use of this type of distance function are similar to those which are explicit in the use of a Gaussian distribution. A further argument against conventional VQ is that it **violates the principle of delayed decision making**, since the VQ process makes hard, irreversible labelling decisions.

5.11 Continuous HMMs

5.11.1 In the case of a **continous HMM** the observation space Y is considered to be **infinite**, multi-dimensional and **continuous**. Therefore, for each state σ_j it is necessary to be able to compute the probability $b_j(y)$ for each acoustic vector y in a continuous space. Unlike in the discrete case, these probabilities cannot simply be stored in a list. Instead it is assumed that the state output PDF b_j takes sort kind of parametric form.

5.11.2 The simplest assumption is that b_j is **Gaussian**, with mean m_j and covariance matrix C_j.

5.11.3 In other words, for any acoustic vector $y \in Y$,

$$b_j(y) = \frac{1}{\sqrt{(2\pi)^d |C_j|}} \exp -\frac{1}{2}\left((y - m_j)^T C_j^{-1}(y - m_j)\right)$$

5.11.4 Figure 33 shows a 1-dimensional Gaussian PDF with mean $m_j=0$ and variance $C_j=20$.

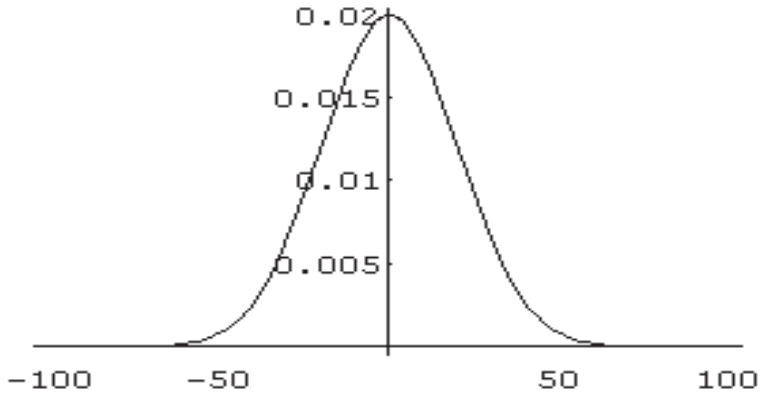


Figure 33: 1-dimensional Gaussian PDF with mean 0 and variance 20

5.11.5 Such a PDF is simple, but it is not sufficiently flexible to accurately model the variation which occurs between different acoustic vectors which

correspond to a state. For example, in a speaker-independent system a multi-modal PDF may be required, with different modes corresponding to different speaker sub-populations. Some researchers would argue, however, that the requirement for a distribution which is more complex that a Gaussian indicates an inadequacy in the underlying model. I guess that basically I agree with this standpoint.

5.11.6 However, in the context of the HMM framework, improved performance can certainly be achieved by employing a more complex form of PDF. The most popular form is the **multiple-component Gaussian mixture**. A *K*-component Gaussian mixture is just a linear combination of *K* Gaussian PDFs. In other words, the state output PDF *bj* has the form:

$$b_j(y) = \sum_{k=1}^{K} w_k b_j^k(y), \text{ where } \sum_{k=1}^{K} w_k = 1$$

and each b_j^k is Gaussian with mean m_j^k and covariance matrix C_j^k.

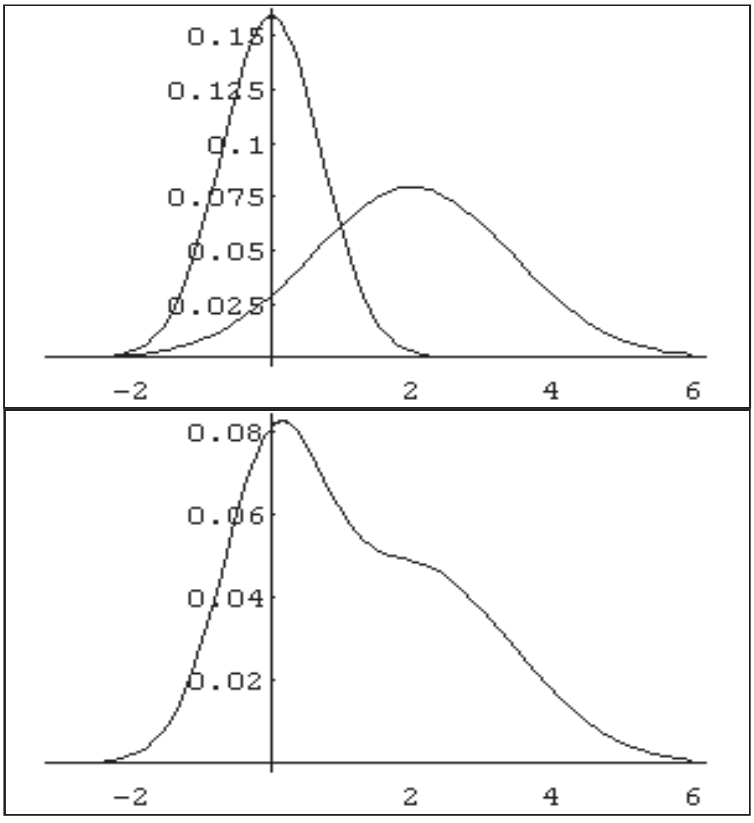


Figure 34: Two 1-dimensional Gaussian PDFs, with means 0 and 2 and variances 1 and 2 respectively (top) and the 2-component Gaussian mixture obtained by combining them with a particular choise of mixture weights (bottom)

5.11.7 In principle, any PDF can be approximated arbitrarily closely by a Gaussian mixture PDF with a sufficiently large number of mixture components. However, the number of parameters in a Gaussian mixture PDF is proportional to the number of mixture components. Hence more components require more training material. Consequently significant research effort has been directed towards the development of methods for robust training of Gaussian mixture state output PDFs with limited training data. This will be discussed in more detail later when we address training algorithms.

5.11.8 It will also be seen later that the choice of the particular parametric form of b_j is influenced by a number of considerations, including the appropriateness of the distribution for modelling variability in acoustic vectors, computational requirements, and the availability of mathematically proven model parameter estimation algorithms

5.12 **Word and Subword Systems**

5.12.1 Word-level HMMs

5.12.2 Early systems, developed in the 1980s, mostly used **word level** HMMs. In other words a single HMM was used to model a complete word. The **advantages** of the word-level approach are as follows:

- It is relatively straightforward
- It provides an explicit model of word-dependent variability between the acoustic realisation of sub-word units in different contexts.

5.12.3 Against these advantages are the following **disadvantages** of the word-level approach:

- Many examples of each word are required for training the word level models
- There is a lack of flexibility. If the engineer designing the interface decides that it is necessary to include an extra word in the vocabulary, then many recordings of that exact word must be collected so that a model can be trained.
- Word level models fail to exploit regularities in spoken language. For example, consider the words ‘five’ and ‘nine’ in a digit recognition system. The phonemic transcriptions of these words are **/ f aI f /** and **/n aI n/** respectively. However, a word level system cannot exploit the fact that the same phoneme **/ aI /** occurs in both transcriptions.

5.12.4 For these reasons, word-level systems are typically restricted to well-defined, demanding, small vocabulary applications.

5.13 Sub-word HMMs

5.13.1 The motivation for the **sub-word** approach is that, in principle, if we construct an acoustic model for each of a complete set of **sub-word units**, then by concatenating these models it is possible to obtain an acoustic model for **any word** in the language **without further training**.

5.13.2 In addition, a sub-word approach should be able to exploit the regularities in the language in a way which is not possible with word level models. For example, in the example cited above, both the words *'five'* and *'nine'* would contribute towards the estimation of the parameters of a model for the phoneme / **aI** /.

5.13.3 The most common type of sub-word HMM is a **phoneme-level** HMM. The advantages of phonemes in this context are that they are:

5.13.4 A **complete** and **compact** set of sub-word units. Completeness refers to the fact that any word in a language can be transcribed as a sequence of phonemes of that language. Compactness refers to the fact that they are relatively few in number - approximately fifty phonemes are required to describe any word in the English language.

5.13.5 Well studied. Hence there is the potential for exploitation of knowledge from the speech sciences. For example, there exist descriptions of pronunciation variation due to accent in phonemic terms.

5.13.6 The key practical advantage is the availability of extensive phoneme-based **pronunciation dictionaries**, which enable idealised pronunciations of most words in a language to be transcribed easily as a sequence of phonemes.

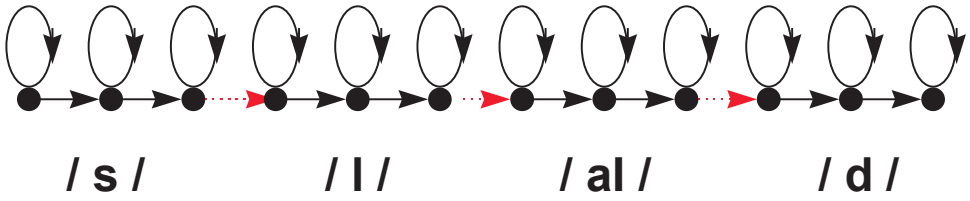


*Figure 35: Schematic diagram showing how HMMs representing the phonemes **/s/, /l/, /aI/** and **/d/** can be concatenated to obtain an acoustic model of the word 'slide' - / **s l aI d /.***

5.13.7 Figure 35 illustrates the principle of concatenating phoneme-level HMMs to obtain a word-level HMM in the case of the word "slide".

5.13.8 The main disadvantage of phonemes is that they are defined in terms of the contrastive properties of speech sounds within a language - **not** their consistency with HMM assumptions! They provide a relatively high-level, symbolic, descriptive framework in speech science. However, it is often

difficult to transform knowledge at this level into something which is **computationally useful**. For example, different instantiations of the same phoneme in different words may result in acoustic patterns which are significantly different. However, because these differences are not **linguistically** important the sounds all constitute the same phoneme.

5.14 Context Sensitivity

5.14.1 To illustrate this point, consider the **/S/** ("sh") sound in a word like 'bookshop' (transcribed as **/b u k S Q p /**) and compare it with the same sound in a word like 'dish' (transcribed as **/d I S/**). In 'bookshop' the lip rounding which is required to produce the **/u/** vowel may easily persist and affect the acoustic realisation of the **/S/**. By contrast, in 'dish' the lips are likely to be still in the wide, stretched configuration which is required for **/I/** when the **/S/** is produced. Thus the acoustic patterns corresponding to the same phoneme **/S/** may be different for these two different realisations.

5.14.2 One can interpret this problem as being one of **context-sensitivity.** The acoustic realisation of a phoneme depends on the sequence of phonemes which precede and follow it in a particular context.

5.14.3 This is a real problem in phoneme-level HMM based approaches to speech recognition. The standard solution is to use **context-sensitive models**, i.e. to build different models for a given phoneme in different contexts. In practice, the most common forms of context-sensitive phoneme-level models are **biphone** and **triphone** HMMs.

5.14.4 **Biphone** HMMs are models of the acoustic realisation of a given phoneme conditioned on the previous (or following) phoneme. Hence in the word "bookshop" (**/b u k S Q p/**) the phoneme **/S/** would be modelled by the biphone **(S:_Q)**, interpreted as **/S/** in the context of a following **/Q/** (or, as the biphone **(S:k_)**, interpreted as **/S/** in the context of a preceding **/k/**). For convenience, biphones of the type **(S:_Q)** will be referred to as **forward-looking biphones**, and biphones of the type **(S:k_)** will be referred to as **backward-looking biphones**.

5.14.5 **Triphone** HMMs are models of the acoustic realisation of a given phoneme conditioned on the immediately **preceding and following** phonemes. Thus in the word "bookshop" the phoneme **/S/** would be represented as the triphone **(S:k_Q)**, interpreted as **/S/** preceded by **/k/** and followed by **/Q/**.

5.14.6 Obviously the use of context sensitive models can result in a huge increase in the number of model parameters.

5.14.7 The basis of biphone and triphone modelling is the assumption that the most significant contextual effects are induced by the immediately neighbouring phonemes. Although this is often the case, it is not generally true. For example, in the example given above the use of the triphone **(S:k_Q)** will

not capture the influence of the preceding **/u/** vowel on the acoustic realisation of the **/S/**.

5.15 Vocabulary Independence

5.15.1 One of the motivations for using phoneme-level HMMs is to enable acoustic models of new words to be constructed directly from their phonemic transcriptions, without any requirement for further training. Unfortunately this ability is compromised by the use of context sensitive models. For, if the transcription of a new word includes a phoneme in a context which did not occur (at all, or sufficiently often) in the training data, which HMM should be used to represent that phoneme in this new context?

5.15.2 Various solutions to this problem have been proposed. These include methods based on **Phoneme Decision Trees (PDTs)** and and **Context Adaptive Phoneme** models **(CAPs)**. Nevertheless, vocabulary-independent performance remains poorer than vocabulary-dependent performance.

5.16 HMM Theory and Practice

5.16.1 One of the major advantage of HMMs is the availability of a **'toolkit'** of powerful, well-founded mathematical methods for HMM manipulation.

5.16.2 The **Baum-Welch** algorithm can be used to train the parameters of a set of HMMs with respect to a set of training data.

5.16.3 **Viterbi Decoding** can be used to classify an unknown speech pattern in terms of the sequence of HMMs which is most likely to have produced it.

5.16.4 This section looks in more detail at the mathematical foundations of HMMs.

5.16.5 The speech recognition problem can be expressed as follows:

5.16.6 Given a sequence of observations $y = \{y_1,...,y_T\}$ we want to find the sequence of words $W = \{w_1,...,w_L\}$ such that the probability $P(W|y)$ of the word sequence W given the acoustic evidence y is maximised.

5.16.7 If $M = \{M_1,...,M_K\}$ is the sequence of HMMs which represents W, then

$$P(W \mid y) = P(M \mid y)$$

5.16.8 Computation of this probability is made possible using **Bayes Theorem**:

$$P(W \mid y) = \frac{P(y \mid W)P(W)}{P(y)}$$

5.16.9 In the context of automatic speech recognition there is an intuitive interpretation of each of the terms in the numerator of the right-hand-side of this expression:

- $P(y|W) = P(y|M)$ is the **acoustic model probability** - the probability that the sequence y was generated by the HMM M.

- $P(W)$ is the probability of the word sequence W. This is computed using a probabilistic language model and is referred to as the **language model probability**.

5.17 The Acoustic Model Probability

5.17.1 The next part of these notes will concentrate of the acoustic model probability $P(y|W)=P(y|M)$. Two issues must be addressed:

5.17.2 The first issue is HMM **training** - how do we construct a HMM M to represent a word sequence W?

5.17.3 The second issue is **recognition** - given a set of HMMs which represent the basic speech units in a system, how does one recognise an unknown utterance?

5.17.4 Before looking at either of these specific issue we must consider the basic ideas and techniques which underlie all HMM computations.

5.18 HMM Computations

5.18.1 The fundamental notion in HMM theory is that of a **state sequence**. In fact, the key to understanding most HMM mathematics lies in understanding the role of state sequences.

5.18.2 A state sequence x of length T for a N-state HMM M is a sequence $x=\{x_1,...,x_T\}$, such that for each t, $x_t = \sigma_j$ for some j. We saw earlier that such a sequence x is a **Markov chain**, generated randomly by the underlying Markov process for M according to the parameters of that process.

5.18.3 The HMM M can only generate a sequence of acoustic vectors $y = \{y_1,...,y_T\}$ of length T via such a Markov chain x of length T, i.e. the acoustic vector y_t is generated by the state x_t. In some descriptions of HMMs, and in particular those produced by the IBM Speech Recognition Group, the sequence y is properly referred to as a **probabilistic function of the Markov chain** x.

5.18.4 To understand the relationship between state-sequences and acoustic vector sequences it is useful (if not essential!) to introduce the notion of the **state-time trellis**.

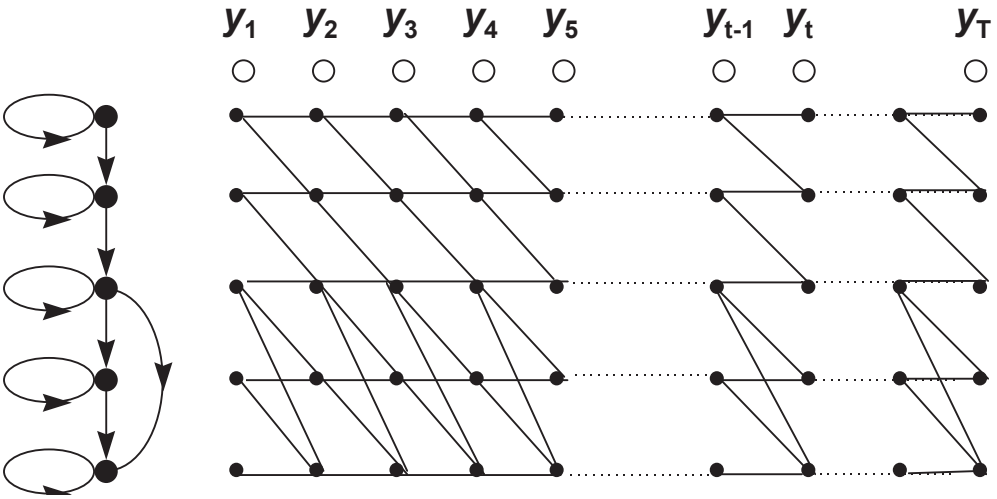


Figure 36: Construction of a simple state-time trellis

5.18.5 A simple state-time trellis is shown in figure 36. The nodes of the trellis correspond to pairs *(j,t)* where *j* corresponds to the j^{th} state of the HMM and *t* corresponds to time. Thus the node *(j,t)* of the trellis corresponds to the hypothesis that the acoustic vector y_t is generated by the state σ_j.

5.18.6 The **connections**, or transtions, in the trellis are determined by the **topology** of the Markov model. A connection is drawn between nodes *(j,t)* and *(k,t+1)*, for all *t=1,...,T-1*, if and only if the transition probability a_{jk} is non-zero. This connection represents the occurrence of a transition between from state *j* to state *k* between times *t* and *t+1*. Take a few moments to understand the relationship between the transition network for the underlying Markov model on the left of figure 32 and the structure of the state-time trellis.

5.18.7 For simplicity let us assume that $\pi_j = \begin{cases} 0 & j \neq 1 \\ 1 & j = 1 \end{cases}$ and that the final observation y_T is generated by the final state σ_N. Then **any state sequence *x* of length *T* (i.e. any state sequence which could have generated *y*) corresponds to a path through the state-time trellis, starting at node *(1,1)* and ending at node *(N,T)***.

5.18.8 Given such a sequence *x* it is straightforward to see that the joint probability *P(y,x|M)* of *y* and *x* conditioned on *M* (i.e. the probability the *M* generates *y* via the state sequence *x*) is given by

$$P(y,x \mid M) = P(y \mid x,M)P(x \mid M) = \prod_{t=1}^{T} b_{x_t}(y_t)\pi_{x_1}\prod_{t=2}^{T} a_{x_{t-1}x_t} = \pi_{x_1} b_{x_1}(y_1)\prod_{t=2}^{T} a_{x_{t-1}x_t} b_{x_t}(y_t)$$

5.18.9 The probability *P(y|M)* that *M* generates the acoustic vector sequence *y* is then given by

$$P(y \mid M) = \sum_{x} P(y, x \mid M)$$

where the sum is over all state sequences x of length T.

5.19 The Forward Probabilities

5.19.1 Unfortunately, direct computation of $P(y|M)$ using this formula is impractical except for very small values on N and T, because of the huge number of possible state sequences. Fortunately, direct computation is unnecessary as there exists a computationally efficient method for evaluating $P(y|M)$. This method is based on the notion of **forward-probabilities**, or **α-probabilities**.

5.19.2 Consider the probability $\alpha_t(i) = \text{Prob}(y_1,..., y_t, x_t = \sigma_i \mid M)$, which is the probability that the HMM M generates the partial acoustic vector sequence $y=y_1,...,y_t$ and that the final observation in this sequence, y_t, is generated by state σ_i.

5.19.3 This expression can be decomposed according to the state which is occupied at time $t-1$, i.e.

$$\alpha_t(i) = \sum_{j=1}^{N} P(y_1,..., y_t, x_{t-1} = \sigma_j, x_t = \sigma_i \mid M)$$

5.19.4 From the basic relationship between joint and conditional probabilities (i.e. $P(a,b)=P(a|b)P(b)$) it follows that

$$\begin{aligned}
&P(y_1,..., y_t, x_{t-1} = \sigma_j, x_t = \sigma_i \mid M) \\
&= P(y_t, x_t = \sigma_i \mid y_1,..., y_{t-1}, x_{t-1} = \sigma_j, M) P(y_1,..., y_{t-1}, x_{t-1} = \sigma_j \mid M) \\
&= P(y_t, x_t = \sigma_i \mid x_{t-1} = \sigma_j, M) \alpha_{t-1}(j) \\
&= P(y_t \mid x_t = \sigma_i, x_{t-1} = \sigma_j, M) P(x_t = \sigma_I \mid x_{t-1} = \sigma_j, M) \alpha_{t-1}(j) \\
&= P(y_t \mid x_t = \sigma_i, M) P(x_t = \sigma_i \mid x_{t-1} = \sigma_j, M) \alpha_{t-1}(j) \\
&= b_i(y_t) a_{ji} \alpha_{t-1}(j)
\end{aligned}$$

5.19.5 This leads directly to the recursive expression for the forward probabilities, sometimes referred to as the **forward recursion**, **forward pass**, or **α-pass**:

$$\alpha_t(i) = \sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji} b_i(y_t)$$

5.19.6 This recursion demonstrates that in order to compute the probability that the partial sequence $y=y_1,...,y_t$ is generated by model M and state σ_i is occupied at time t, it is simply necessary to propagate forward the corresponding

quantities for time *t-1*. Figure 37 shows an interpretation of $\alpha_i(y_t)$ in terms of the state-time trellis.

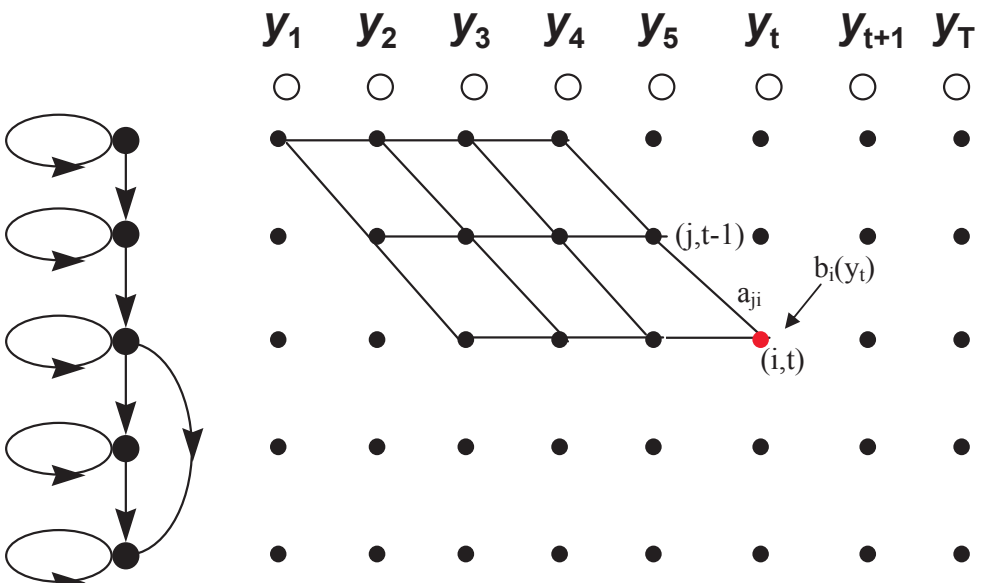


Figure 37: Frame-state trellis interpretation of the calculation of the forward probabilities.

5.19.7 An alternative interpretation of $\alpha_i(y_t)$ is $\alpha_i(y_t) = \sum P(y_1,...,y_t, x \mid M)$,

where the sum is over all state sequences x of length t which are in **state σ_i at time *t***. I.e. the sum is over all possible routes through the sub-trellis shown in figure 37.

5.20 Viterbi Decoding

5.20.1 In principle, the calculations of the previous section enable us to apply **maximum likelihood classification** to recognise a sequence of acoustic vectors y, corresponding to a spoken word, as being an instantiation of the word W, where:

$W = \text{argmax}_V P(y|V)$.

5.20.2 In reality, of course, speech is not like that. Words do not occur as isolated acoustic patterns and there are no explicit acoustic cues to word boundaries in continuous speech (unless we force the speaker to leave gaps between words). The sequence y will typically correspond not to a single word, but to a phrase or sequence of words. What is needed is a method which is able to find the sequence of words whose corresponding HMM sequence is most likely to have generated y. For any reasonably sized vocabulary and grammar, the set of possible word sequences will be huge, and possibly infinite.

5.20.3 If *M* corresponds to an integrated model of the whole language, then *P(y|M)* will be the probability the sequence *y* corresponds to some sentence of the language - interesting but not particularly useful in the present context!

5.20.4 Fortunately there is a computationally efficient algorithm for computing optimal sequences, corresponding to the **best** interpretation of the utterance as a sequence of words. These sequences can be sequences of words, but it is simpler to start with the idea of an **optimal state sequence**.

5.20.5 Returning to the notation of the previous section, given a sequence of acoustic vectors $y = y_1,..., y_T$ and a HMM *M*, we say that a state sequence $\hat{x} = \hat{x}_1,..., \hat{x}_T$ is an **optimal state sequence (for *y* relative to *M*)** if:

$$\hat{x} = \arg\max_x P(y, x \mid M)$$

and we write:

$$\hat{P}(y \mid M) = \max_x P(y, x \mid M) = P(y, \hat{x} \mid M) .$$

5.20.6 In terms of figure 32, the optimal state sequence corresponds to the path through the state-time trellis for which the product of state output probabilities and state transition probabilities is maximal.

5.20.7 The optimal state sequence $\hat{x}$ and the corresponding probability $\hat{P}(y \mid M)$ can be computed via a computationally efficient algorithm which is analogous to the forward probability calculation.

5.20.8 Write $\hat{\alpha}_t(i) = \max_{x_1,...,x_t} P(y_1,..., y_t, x_1,..., x_t, x_t = \sigma_i \mid M)$, then by an argument which is analogous to the corresponding argument for the forward probabilities, it can be shown that:

$$\hat{\alpha}_t(i) = \max_j \left(\hat{\alpha}_{t-1}(j) a_{ji} b_i(y_t) \right)$$

5.20.9 Using this equation, the probability $\hat{\alpha}_t(i)$ can be computed for each point *(i,t)* in the state-time trellis, starting at the top left-hand corner and working top-to-bottom, left-to-right. Clearly $\hat{P}(y \mid M) = \hat{\alpha}_T(N)$. Moreover, if at each point *(i,t)* in the state-time trellis a record is kept of the state *j* which corresponds to the maximum at time *t-1*, then by **tracing-back** along these local records, starting at *(N,T)* in the state-time trellis, the optimal state sequence $\hat{x}$ can be recovered.

5.20.10 The recursive equation for $\hat{\alpha}_t(i)$ is usually referred to as the **Viterbi algorithm** and the whole process of computing the probability $\hat{P}(y \mid M)$ and the optimal state sequence $\hat{x}$ in this way is referred to as **Viterbi decoding**.

5.20.11 More generally, Viterbi decoding is one example of the application of the principle of **Dynamic Programming** in speech pattern processing. From this perspective, the recursive equation for $\hat{\alpha}_t(i)$ is an affirmation that $\hat{\alpha}_t(i)$ obeys the **principle of optimality**, which is the basic prerequisite for the application of dynamic programming.

5.21 The Bridle "One-Pass" Algorithm

5.21.1 In this section we shall see that Viterbi decoding can be extended from the problem of finding optimal state sequences to the more general problem of finding optimal sequences of HMMs. In speech recognition terms, this means that we shall be able to find the sequence of HMMs which is most likely to have given rise to a sequence of acoustic vectors. In other words we can do simple (acoustic) **connected speech recognition**.

5.21.2 Note that the phrase **connected** speech recognition has been carefully chosen. It refers to the process of finding the sequence of words (or phonemes, or other symbols) which provides the most likely explanation of a **finite** sequence of acoustic vectors, starting at time *t=1* and ending at time *t=T*. In practice, we may not know when an utterance started, and we have little idea of when it is going to end. Waiting until the speech ends (whatever that means) is not a viable option if our aim is real-time speech recognition. This more difficult problem is referred to as **continuous speech recognition**, and its solution will be the subject of the next section. For now we shall focus on connected speech recognition.

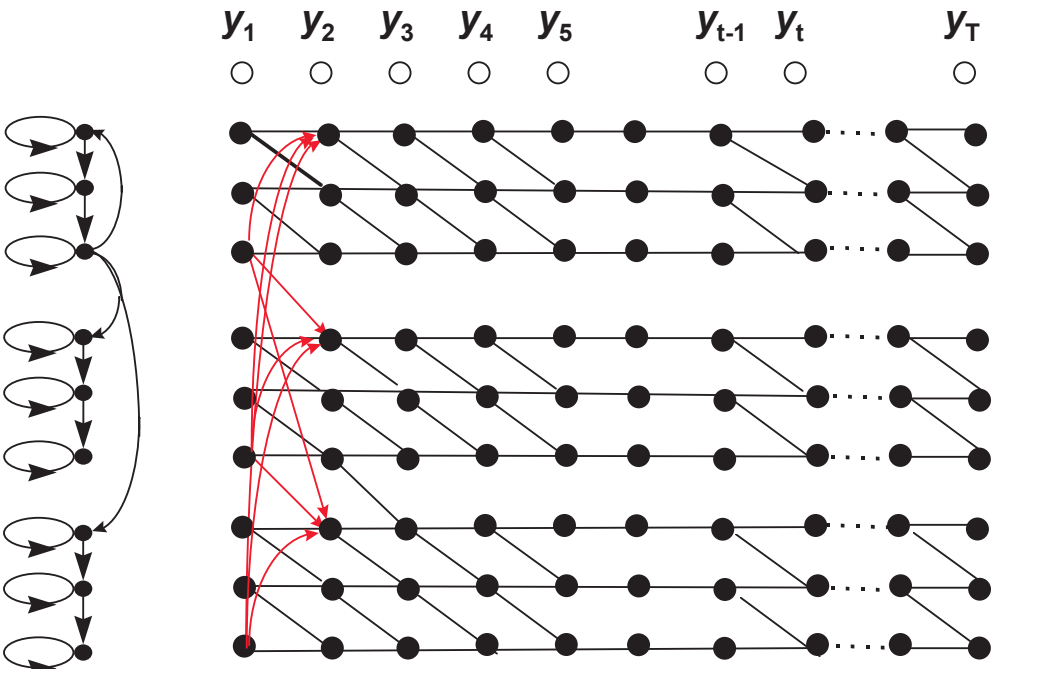


Figure 38: State-time trellis for connected word recognition, showing expanded state transition set in HMMs and in the first column of the state-time trellis. In practice, this first column of the trellis would be copied for all columns

5.21.3 Figure 38 illustrates the principle which underlies connected word recognition. Additional transitions are included between the final state of each HMM and the first state of it and all other HMMs, corresponding to the fact that each word can be followed by any other word. This additional structure is reflected in the state-time trellis, the expanded first column of which is shown in the figure.

5.21.4 Viterbi decoding can now be applied to the expanded trellis. If the i_N^{th} row of the trellis corresponds to the final state of the i^{th} HMM, then the probability $\hat{\alpha}_T(i_N)$ is the joint probability of the acoustic vector sequence y and the best state sequence x which ends in the final state of the i^{th} HMM. If $\hat{i}_N = \arg\max_i\left(\alpha_T(i_N)\right)$, then the optimal state sequence ends in the final state of the $\hat{i}^{\text{th}}$ HMM. In other words the best explanation of the data ends in the $\hat{i}^{\text{th}}$ word (or phoneme). By **tracing back** according to the pointers which were created during the forward pass of the Viterbi algorithm, the optimal state sequence, and hence the optimal explanation of the data, can be recovered.

5.21.5 In other words, Viterbi decoding can be extended quite simply to compute the best explanation of a finite sequence of acoustic vectors in terms of the outputs of the **best sequence** of HMMs from a given set of HMMs.

5.21.6 Notice that one by-product of this analysis is an estimate of the start-points and end-points of each of the words (or phonemes) in this best explanation. It is critical to realise that **these endpoints emerge as a consequence of the recognition process**, and are not a prerequisite to it. This is another illustration of the principle of **delayed** (or **deferred**) **decision making**.

5.21.7 In fact, in some applications the estimation of these endpoints is an end in itself. If the actual sequence of words (or phonemes) is known, then a concatenated HMM, comprising the sequence of appropriate HMMs, can be constructed. Viterbi decoding can then be used to estimate the word or phoneme start and end times. This use of Viterbi decoding is referred to as **forced alignment**.

5.22 Partial Traceback

5.22.1 In practice, even if time *t=1* is taken to be the time at which the speech recogniser was switched on, the endpoint *T* will not be known. Even if we could detect 'silence' (which is a non-trivial task in the presence of background noise) it will not be clear in advance whether the pause is at the end of a sentence, or in the middle of a sentence, or even in the middle of a

word. And what is a sentence, anyway? Besides, any prior segmentation of the speech signal is contrary to the principle of delayed decision making and therefore a likely source of irrecoverable errors.

5.22.2 However, if we do not have an endpoint *T* we will just go on computing the probabilities $\hat{\alpha}_t(i)$, for all states *i*, for ever-increasing values of *t* until the back-pointers occupy all of the available memory. Moreover, since the 'end' of the utterance has not been reached, no trace-back will have occurred and so no output will have been produced! This problem is overcome using the technique of **partial traceback**.

5.22.3 Let *t* be fixed. For each state σ_j there will be a pointer indicating the state which was occupied at time *t-1* by the optimal state sequence up to the point *(j,t)* in the state-time trellis. If we trace-back along this path we will, eventually, reach a word boundary. We can then pick up the sequence of word-link records and trace the history of this path as far back as we wish. Now suppose that there exists a time $t_0 < t$ such that **all of the active paths at time *t*, when traced-back to t_0, converge to exactly the same explanation of the data**. For example, all paths may agree that the data at time t_0 is the end of a period of silence. No matter what happens beyond time *t* all future paths will also converge at t_0, because all paths beyond time *t* will reduce to one of the known optimal paths at time *t*. Therefore, whatever happens beyond time *t* cannot influence the explanation of the data up to time t_0. Hence the current best explanation of the data up to time t_0 will **always** be the best explanation of the data up to time t_0 and can be output, and the memory used to store the word-link records up to time t_0 can be freed. This is the principle of **partial traceback**.

5.22.4 In a continuous speech recognition system, partial traceback is performed at regular intervals, determined by a parameter called the **traceback frequency**.

5.22.5 Of course, it may happen that there is no point of convergence in the past. In this case a pragmatic solution is required - for example the current best explanation of the data could be output to enable memory to be freed.

5.22.6 It is important to not that partial traceback does not compromise optimality. If a point of convergence in the past can be found, then the resulting explanation of the data up to that point of convergence will be optimal.

5.23 Beam-Pruning - "Saving the Batteries"

5.23.1 **Beam pruning** is a practical technique for reducing the computational load in Viterbi decoding.

5.23.2 Let time *t* be fixed. Suppose that, at time *t-1,* some state σ^{t-1} corresponding to a state σ_j in a model *M* has the property that

$$\hat{\alpha}_{t-1}(\sigma^{t-1}) = \max_j \hat{\alpha}_{t-1}(j)$$

where the maximum is taken over all models M and all states σ_j. In other words, the best state sequence ending in state σ^{t-1} at time *t-1* provides the best explanation of the data up to time *t-1*. The probability associated with this sequence provides a criterion against which the probabilities of the best state sequences ending in other states at time *t-1* can be assessed. For example, if $\hat{\alpha}_{t-1}(\sigma^{t-1}) >> \hat{\alpha}_{t-1}(j)$ then one can argue that the best state sequence ending in state *j* at time *t-1* is unlikely to extend to be the globally optimal state sequence at some time in the future. Hence, one might decide to discontinue, or **prune-out** this state sequence. Beam pruning utilises a threshold *B*, called the **beam-width**, such that all states *j* with $|\hat{\alpha}_{t-1}(\sigma^{t-1}) - \hat{\alpha}_{t-1}(j)| > B$ are pruned-out at time *t-1*. This leads to a computational saving at time *t*, because any state which only admits transitions from states which have been pruned-out at time *t-1* **need not be considered in Viterbi decoding at time *t*.**

5.23.3 Figure 35 illustrates the principle of beam-pruning.

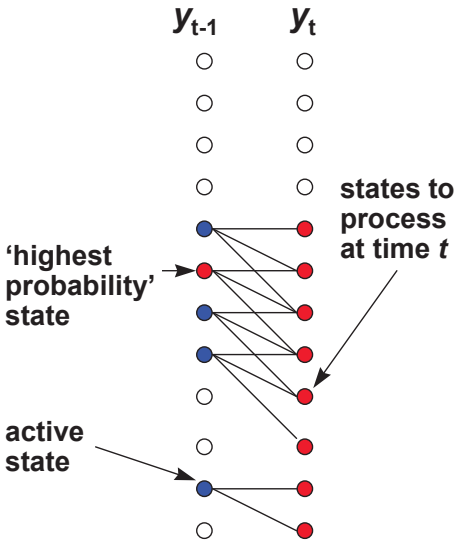


Figure 39: Beam-pruning at time t in Viterbi decoding.

5.23.4 This concludes the discussion of speech recognition using HMMs.

5.23.5 However, all of this is academic unless we have a way of producing an appropriate set of word-level or phoneme-level HMMs.

5.24 HMM Training - The Baum-Welch Algorithm

5.24.1 We shall now look at the problem of HMM training. First note that an acoustic model of a word sequence can be obtained by concatenating the

appropriate word-level models, and that word-level models can be obtained by concatenating sub-word models. In other words, the precise unit which is considered is not important.

5.24.2 The **Baum-Welch algorithm** is the basis of the standard approach to HMM training. It is also referred to as the **Forward-Backward algorithm**, for reasons which will shortly become clear. The basis of the Baum-Welch algorithm is **Baum's Theorem**.

5.24.3 Suppose that M_0 is an HMM and y is a sequence of acoustic vectors representing the spoken utterance corresponding to M_0. Then **Baum's Theorem** shows how a new model set M_1 can be defined such that $P(y|M_1) \geq P(y|M_0)$.

5.24.4 The assumptions that there is a single training utterance y and that y corresponds to an instantiation of the utterance corresponding to M_0 are both unnecessary. The theorem can be extended to cover the case of multiple training utterances and arbitrary HMM sets, provided that each training utterance can be expressed as a sequence of basic utterances, each of which corresponds to an HMM.

5.24.5 Basically, the **Baum-Welch algorithm** involves repeated application of Baum's theorem. An initial estimate M_0 is made, and Baum's theorem is applied to M_0 to obtain M_1. Baum's theorem is then applied to M_1 to produce M_2 such that $P(y|M_2) \geq P(y|M_1)$, and so on. In this way a sequence of models $M_0, M_1, M_2, \ldots, M_n$ is produced, each satisfying $P(y|M_n) \geq P(y|M_{n-1})$. When the difference $|P(y|M_n) - P(y|M_{n-1})|$ is less than some pre-determined threshold ε (and, possibly, remains less that ε over some pre-determined time interval) then the Baum-Welch algorithm is considered to have converged.

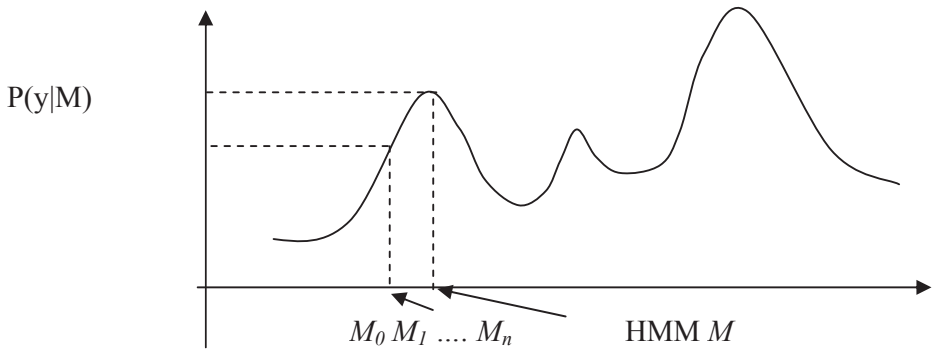


Figure 40: Hill climbing behaviour of the Baum-Welch algorithm

5.24.6 The Baum-Welch algorithm is a **hill climbing** algorithm, which will converge to a **local maximum** of the function $P(y|M)$ (see figure 40). The identity of this maximum will depend on the choice of M_0.

5.24.7 Baum's theorem (and hence the Baum-Welch algorithm) is only valid for particular classes of state output PDF, however, this class includes discrete, Gaussian, and multiple component Gaussian mixture PDFs.

5.24.8 Baum-Welch is a **supervised** training scheme. This means that it requires labelled training data. The basic units of labelled training data need not be the same as the HMMs - e.g. phoneme-level HMMs can be trained using speech labelled at the sentence level. However, it must be possible to translate each training set label into a sequence of model labels. This is normally done using a pronunciation dictionary.

5.25 Derivation of Baum's Theorem (based on Liporace, 1982)

5.25.1 Given a sequence of acoustic vectors $y=y_1, y_2, ..., y_T$, and an HMM M_0, we want to construct a HMM M_1 such that $P(y|M_1) \geq P(y|M_0)$.

5.25.2 Define the **auxiliary** function Q by:

$$Q(M_0, M_1) = \sum_X P(y, x \mid M_0) \log P(y, x \mid M_1)$$

5.25.3 It is easy to show that if $Q(M_0, M_1) \geq Q(M_0, M_0)$ then $P(y \mid M_1) \geq P(y \mid M_0)$ Since direct analysis of $P(y|M_0)$ is not productive, Liporace's derivation of Baum's theorem focuses onto the auxiliary function Q. It follows from the above inequalities that in order to find a model M_1 such that

$P(y|M_1) \geq P(y|M_0)$,

5.25.4 it is certainly sufficient to find M_1 such that

$Q(M_0,M_1) \geq Q(M_0,M_0)$.

5.25.5 This will certainly be the case if $Q(M_0,M)$, thought of as a function of M, has a unique maximum and M_1 is chosen to be that maximum.

5.25.6 It turns out that the function $M \rightarrow Q(M_0,M)$ does have a unique critical point, which is a maximum (this requires some sophisticated mathematics). Computing the partial derivatives of $Q(M_0,M)$ with respect to the parameters of M, setting the resulting expressions to 0, and solving the equations gives a new model M_1 whose parameters are expressed in terms of the parameters of M_0 and the training data y. This requires little more than A-level mathematics. These formulae for the parameters of M_1 are called the **Baum-Welch re-estimation formulae**.

5.25.7 Figure 41 illustrates the role of the auxiliary function in this derivation.

5.26 The Baum-Welch Re-estimation formulae

5.26.1 Suppose that M_0 is an N state HMM with single Gaussian state output PDFs. Suppose further that the i^{th} state output PDF is defined by its mean m_i and covariance matrix C_i. Then the re-estimated HMM M_1 is specified by the following re-estimation formulae:

$$\overline{m}_i^d = \frac{\sum_{t=1}^{T} \sum_{x \in S_{i,t}} P(y,x \mid M_0) y_t^d}{\sum_{t=1}^{T} \sum_{x \in S_{i,t}} P(y,x \mid M_0)}$$

where $S_{i,t} = \{x : x_t = \sigma_i\}$ is the set of all state sequences x which include state σ_i at time t.

$$\overline{c}_i^{d,e} = \frac{\sum_{t=1}^{T} \sum_{x \in S_{i,t}} P(y,x \mid M_0)(y_t^d - m_i^d)(y_t^e - m_i^e)}{\sum_{t=1}^{T} \sum_{x \in S_{i,t}} P(y,x \mid M_0)}$$

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \sum_{x \in S_{i,j,t}} P(y,x \mid M_0)}{\sum_{t=1}^{T-1} \sum_{x \in S_{i,t}} P(y,x \mid M_0)}$$

where $S_{i,j,t} = \{x : x_t = \sigma_i, x_{t+1} = \sigma_j\}$

$$\overline{\pi}_i = \frac{\sum_{x \in S_{i,1}} P(y,x \mid M_0)}{P(y \mid M_0)}, \text{ where } S_{i,1} = \{x : x_1 = \sigma_i\}$$

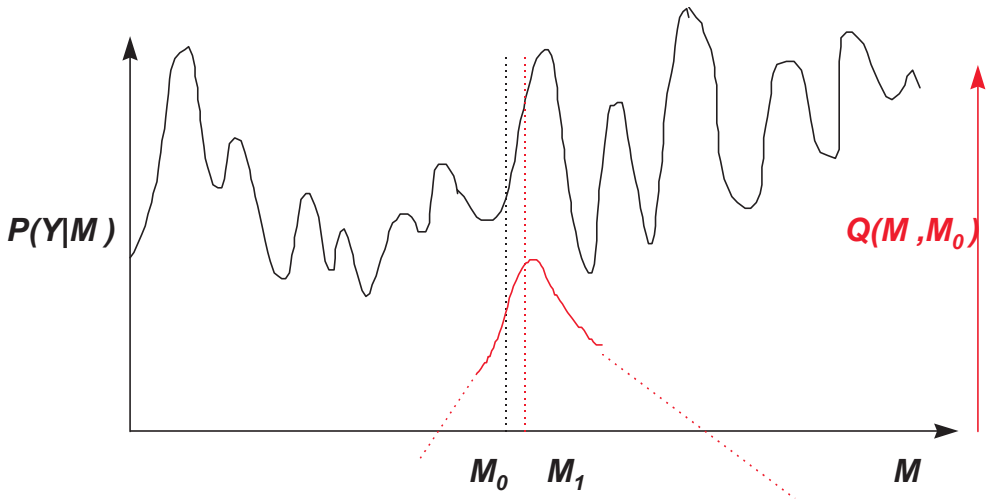


Figure 41: The role of the auxiliary function Q in the derivation of the Baum-Welch re-estimation formulae.

5.26.2 In these equations, the subscripts d and e indicate the d^{th} and e^{th} components of a vector and the subscript j indicates that a parameter is associated with the j^{th} state. The 'bar' indicates the re-estimated value of a parameter, i.e. a parameter of the re-estimated model M_1 rather than the original model M_0.

5.27 Interpretation of the Baum-Welch Formulae

5.27.1 It has to be admitted that the Baum-Welch re-estimation formulae presented in the previous section are rather opaque! The purpose of the current section is to provide an intuitive understanding. In doing this we shall also go some way towards deriving a computationally efficient algorithm for computation of the re-estimation formulae. As in the previous section, we shall concentrate on the case of Gaussian HMMs.

5.27.2 The first step in understanding the Baum-Welch re-estimation formulae is to ask why we need them in the first place! Suppose we have a sequence of acoustic vectors $y=y_1,...,y_T$ and an N-state left-right Gaussian HMM M. Suppose that y is generated through a state sequence $x=x_1,...,x_T$, and let t_j denote the time corresponding to the first occurrence of state σ_j in x. Then state σ_j corresponds exactly to the sub-sequence $y_{t_j},...,y_{t_{j+1}-1}$ and an obvious candidate for the re-estimate of the mean m_j of the Gaussian distribution corresponding to state σ_j is the average $\overline{m}_j = \frac{1}{t_{j+1}-t_j}\sum_{t=t_j}^{t_{j+1}-1} y_t$

5.27.3 Of course, the snag is that because the model M is **hidden** we don't know which state sequence x generated y, so the procedure described above cannot be applied directly. However, for any state sequence x we can compute the joint probability $P(y,x|M)$, and this can be taken as a measure of how **probable** it is that M generated y via x.

5.27.4 Now let's look again at the numerator in the reestimation formula for the mean.

$$\sum_{t=1}^{T}\sum_{x\in S_{i,t}} P(y,x\,|\,M_0)\,y_t^d$$

5.27.5 By changing the order of summation this can be written as

$$\sum_{t=1}^{T}\sum_{x\in S_{i,t}} P(y,x\,|\,M_0)\,y_t^d = \sum_{x\in S_i}\sum_{t=t_i}^{t_{j+1}-1} P(y,x\,|\,M_0)\,y_t^d = \sum_{x\in S_i} P(y,x\,|\,M_0)\sum_{t=t_i}^{t_{i+1}-1} y_t^d$$

5.27.6 Here S_j is the set of state sequences of length T which include state σ_j. Thus the numerator of the Baum-Welch re-estimation formula for the mean of the Gaussian PDF associated with a state is simply a weighted sum of the state-sequence specific averages which were introduced at the start of this section. Moreover, the weights can be interpreted as the probability that the state sequence in question was responsible for the generation of y.

5.27.7 It is important to note that phrases such as "*the probability that the state sequence in question was responsible for the generation of y*" refer to probabilities relative to the model M.

5.27.8 For example, if *M* is a word-level model and it is a "good" model in the sense that the means corresponding to its sequence of states correspond to a phonetically sensible representation of the word, then a "probable" state sequence is likely to be one which segments a training utterance in a "phonetically plausible" manner. In this case the re-estimation formula will give more weight to averages which correspond to "phonetically plausible" segmentations of the training data, and the reestimated HMM is likely to be also phonetically plausible.

5.27.9 By contrast, if the initial model is not phonetically plausible, then the most probable state sequences may also be implausible and the reestimated model, even though it is guaranteed to be more likely to have generated the training data may be even less phonetically plausible. In this way an intuitive understanding can be built up of the way in which the choice of the initial model influences which local optimum will be reached by the Baum-Welch reestimation process.

5.28 **Computation of the Baum-Welch Formulae**

5.28.1 Consider the numerator of the Baum-Welch re-estimation formula for the mean m_j of the state σ_j:

$$\sum_{t=1}^{T} \sum_{x \in S_{i,t}} P(y, x \mid M_0) y_t^d$$

5.28.2 For fixed *t* (i.e. focussing on a single summand) this corresponds to a sum over all state sequences which pass through the point *(i,t)* in the state-time trellis, as illustrated in figure 42. The relationship between the top left-hand corner of this diagram and the corresponding diagram for the forward probabilities is obvious and motivates the following decomposition of such a summand:

$$\sum_{x \in S_{i,t}} P(y, x \mid M) = P(y, x_t = \sigma_i \mid M)$$
$$= P(y_1, ..., y_T, x_t = \sigma_i \mid M)$$

$$= P(y_1, ..., y_t, x_t = \sigma_i \mid M) P(y_{t+1}, ..., y_T \mid y_1, ..., y_t, x_t = \sigma_i, M)$$
$$= P(y_1, ..., y_t, x_t = \sigma_i \mid M) P(y_{t+1}, ..., y_T \mid x_t = \sigma_i, M)$$
$$= \alpha_t(i) P(y_{t+1}, ..., y_T \mid x_t = \sigma_i, M)$$

5.28.3 Given this expression, it is natural to introduce the notion of **the backward, or β-, probabilities**, which are normally denoted by $\beta_t(j)$ and defined by:

$$\beta_t(j) = Prob(y_{t+1}, ..., y_T \mid x_t = \sigma_j, M)$$

5.28.4 In this notation,

$$\sum_{x \in S_{i,t}} P(y, x \mid M) = \alpha_t(i)\beta_t(i)$$

5.28.5 The results of this product are often referred to as the **gamma probabilities**, i.e. $\gamma_t(i)=\alpha_t(i)\beta_t(i)$.

5.28.6 As in the case of the forward probabilities, it is possible to derive a computational efficient and intuitively plausible recursive algorithm for the computation of the backward probabilities:

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) b_j(y_{t+1}) a_{ij}$$

5.28.7 The use of the forward and backward recursions in the computation of the terms in the reestimation formulae in the Baum-Welch algorithm accounts for this algorithms other common name - the **Forward-Backward algorithm**.

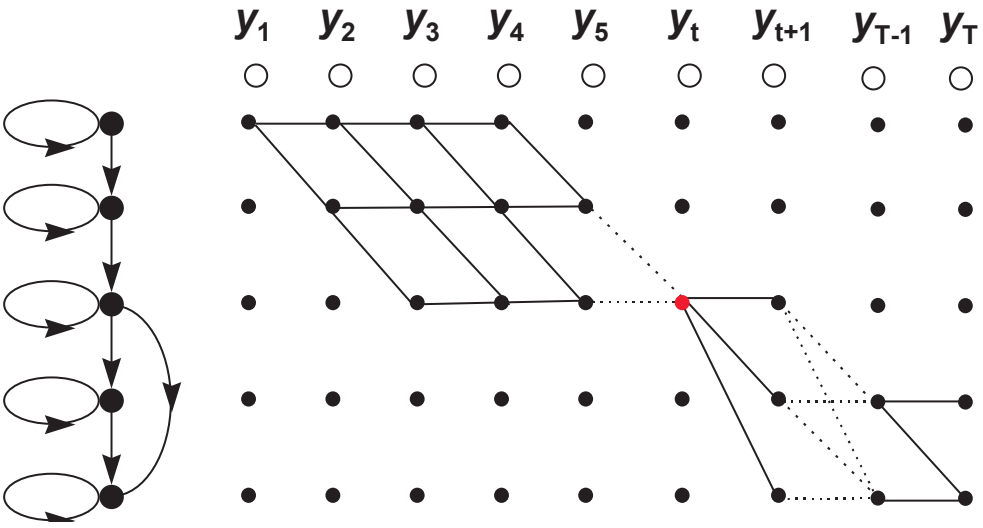


Figure 42: Interpretation of a single summand for fixed t in the numerator of the Baum-Welch re-estimation formula for the mean of a Gaussian PDF associated with a HMM state.

5.28.8 To conclude, we now have the following expressions for the Baum-Welch reestimation formulae for the parameters of an *N*-state Gaussian HMM in terms of the forward and backward probabilities:

$$\overline{m}_i^d = \frac{\sum_{t=1}^{T} \alpha_t(i)\beta_t(i) y_t^d}{\sum_{t=1}^{T} \alpha_t(i)\beta_t(i)}$$

$$\overline{c}_i^{d,e} = \frac{\sum_{t=1}^{T} \alpha_t(i)\beta_t(i)(y_t^d - m_i^d)(y_t^e - m_i^e)}{\sum_{t=1}^{T} \alpha_t(i)\beta_t(i)}$$

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i)a_{ij}b_j(y_{t+1})\beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)}$$

$$\overline{\pi}_i = \frac{\alpha_1(i)\beta_1(i)}{\sum_{j=1}^{N} \alpha_1(j)\beta_1(j)}$$

5.28.9 Finally, notice that for any *t=1,...,T,*

$$\sum_{j=1}^{N} \alpha_t(j)\beta_t(j) = \sum_{j=1}^{N} P(y, x_t = \sigma_j \mid M) = P(y \mid M)$$

5.28.10 That concludes the discussion of hidden Markov models.

6 Language Modelling

6.1 Role of the language model probability

6.1.1 Recall that given acoustic measurements y corresponding to an unknown utterance, want to find the word sequence W such that $P(W \mid y)$ is maximised

6.1.2 By Bayes' Theorem, $P(W \mid y) = \frac{P(y \mid W)P(W)}{P(y)}$

6.1.3 $P(W)$ is the probability that the word sequence W is in the application language and is called the **language model probability**

6.1.4 The function of the Language Model, or Grammar, is to compute the probability P(W) that the sequence of words W 'belongs to' the language. But, what type of Language model is appropriate for Automatic Speech Recognition?

6.1.5 Basically there are two types of candidate:

- Rule-based language model
- Probabilistic language model

6.2 Rule-based language models

6.2.1 The 'conventional' language models used in **linguistics** and **natural language processing** are **rule-based.** A rule-based language model consists of:

- A set of **non-terminal** units (e.g. sentence, noun-phrase, verb-phrase,…)
- A set of **terminal units** (e.g. words)

6.2.2 A set of **rules** which define how non-terminal units can be expanded into sequences of non-terminal and terminal units

6.2.3 This corresponds to the formal notion of 'grammar' taught in schools. Let S denote the non-terminal root node corresponding to 'sentence'. A sequence of words is **grammatical**, or **in the language** if it can be derived from S by the application of a sequence of rules. An example, for the sentence "The cat devoured the tiny mouse" (Finch (1998)) is shown in figure 43.

6.2.4 This approach to language modelling has advantages and disadvantages:

6.2.5 The main **advantages** are:

- that they can model complex structure, e.g. non-local dependencies

– that significant expertise and knowledge already exists

– that much effort has already been devoted to the construction of large language models of this type

6.2.6 The principle **disadvantages** are

– that they are normally applied to **written** language,

– that a deterministic model of this type may not be able to characterise **spoken** language, which is much more variable,

– that they cannot handle uncertainty

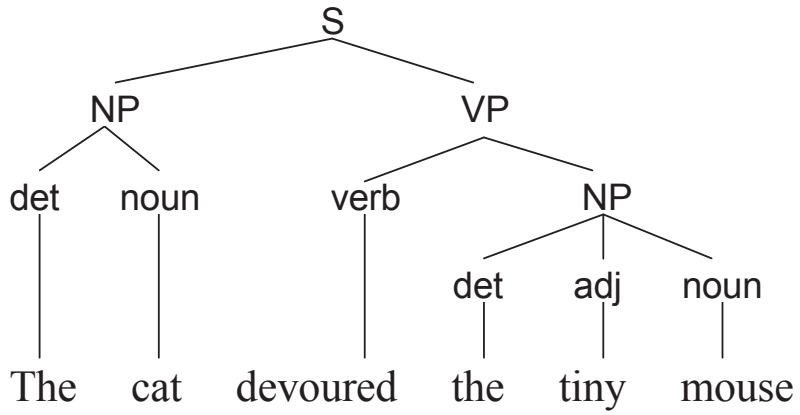


Figure 43: Conventional rule-based analysis of the sentence "the cat devoured the tiny mouse (Finch (1998))

6.2.7 With a rule based language model, a sequence of words *W* is either in the language (**grammatical**) or outside the language (**not grammatical**). With a statistical language model, a sequence of words *W* is in the language (**grammatical**) with **probability *P(W)*** .

6.2.8 The most common statistical language model is known as the ***N*-gram model.**

6.3 N-gram Language Models

6.3.1 Let $W = W_1, W_2, ..., W_K$ be a sequence of words. Then, in general:

$$P(W) = P(W_1)P(W_2|W_1)...P(W_n|W_{n-1}, ..., W_1)...P(W_K|W_{K-1}, ..., W_1)$$

6.3.2 In an *N*-gram language model, we assume:

$$P(W_k|W_{k-1}, W_{k-2}, ..., W_1) = P(W_k|W_{k-1}, ...W_{k-N+1})$$

I.e. the probability of the *kth* word in the sequence depends only on the identities of the **previous *N-1* words**. The most commonly used *N*-gram models are *2*-gram (**bigram**) and *3*-gram (**trigram**) models.

6.3.3 In a **Bigram** Language Model,

$$P(W_k|W_{k-1},W_{k-2},...,W_1) = P(W_k|W_{k-1})$$

6.3.4 In a **Trigram** Language Model,

$$P(W_k|W_{k-1},W_{k-2},...,W_1) = P(W_k|W_{k-1},W_{k-2})$$

6.3.5 These probabilities can be **estimated from data**.

6.3.6 For example, given a training text, an **estimate** of the **bigram probability** $P(W_2|W_1)$ is given by

$$P(W_2|W_1) = N(W_1,W_2)/N(W_1)$$

where

- $N(W_1,W_2)$ is the number of times the word pair W_1,W_2 occurs in the training text
- $N(W_1)$ is the number of times the word W_1 occurs in the training text

6.3.7 Consider the training text:

"John sat on the old chair. John read the old book. John was interesting. The book was interesting"

6.3.8 Suppose that this text is used to train a bigram grammar.

6.3.9 *'the'* occurs *3* times in the text, while the bigrams *'the old'* and *'the book'* occur twice and once respectively.

Hence

$P('old'|'the')$=2/3, and

$P('book'|'the')$=1/3.

6.3.10 Similarly, if the symbol # denotes start of sentence (and $ denotes end of sentence, then

$P('john'|\#)$=3/4, and

P('the'|#)=1/4

P($|chair)=1

6.3.11 The probability of the sentence *S "John sat on the old chair"* is given by:

P(S)= P(john|#)P(sat|john)P(on|sat)P(the|on)P(old|the)P(chair|old)P($|chair)

= 3/4 x 1/3 x 1 x 1 x 2/3 x ½ x 1

= 1/12

6.3.12 In practice, most systems use a **trigram** language model

6.3.13 In reality, there is never enough text to estimate trigram probabilities in this simple way. For example, in the early 1980s IBM experimented with trigram language models for a 1,000 word vocabulary application. They used 1.5 million words for training, and 300,000 words to test the models. 23% of the trigrams in the **test** corpus were **absent from the training** corpus.

6.3.14 Hence much more sophisticated training procedures are needed.

6.4 Advantages and disadvantages of N-gram language models are:

6.4.1 Advantages:

Can be trained automatically from data

Probabilistic model

Consistent with acoustic model

Mathematically sound algorithms

6.4.2 Disadvantages:

Large amounts of **training data** needed

Difficult to incorporate **human knowledge**

Cannot model long term dependency: "She walked, hands in pockets, quickly across the bridge"

6.4.3 On balance, nearly all practical speech recognition systems use some form of statistical, *N*-gram language model

6.5 **Summary: An HMM-based speech recognition system**

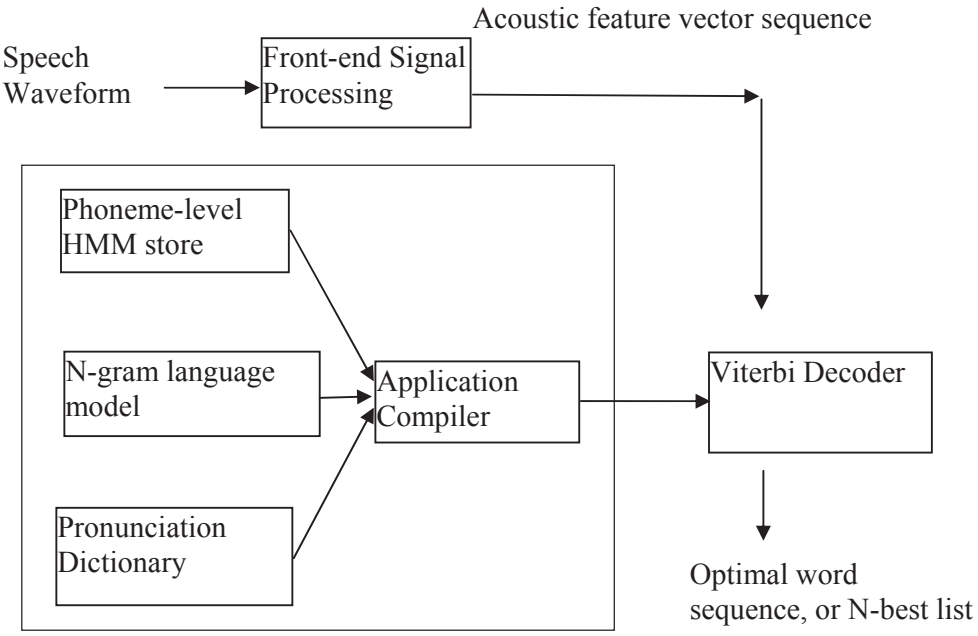


Figure 44: Schematic diagram of a typical phoneme-HMM-based large vocabulary continuous speech recognition system

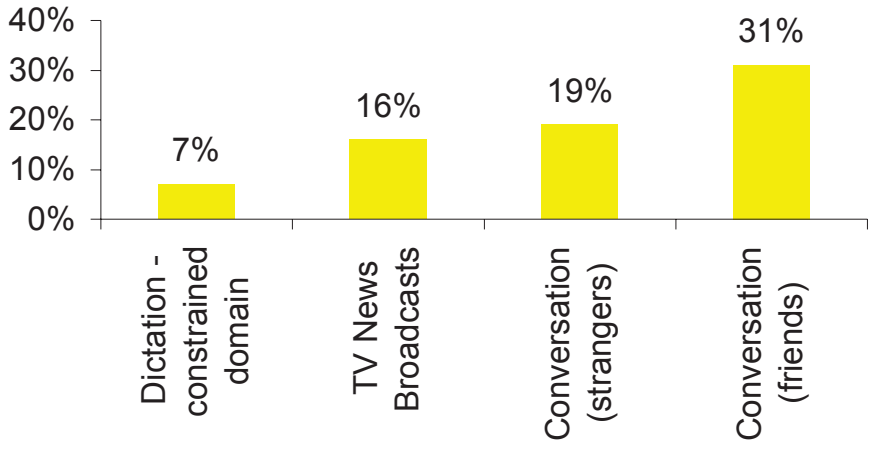


Figure 45: Typical performance of current laboratory-based large vocabulary continuous speech recognition systems.

7 Speaker Recognition

7.1 What is speaker recognition?

7.1.1 Speaker recognition, as its name suggests, is concerned with determining **who** is speaking, rather than what has been said. It splits into three sub-topics:

7.1.2 **Speaker verification**. This is concerned with verifying that a given speaker is who they claim to be.

7.1.3 **Speaker identification**. This is concerned with detecting a particular speaker from a potentially unlimited population

7.1.4 **Speaker recognition**: This is concerned with recognising which of a population of speakers responsible for a given utterance.

7.1.5 Other related problems in speech technology include **word spotting** (detecting that a particular word has been spoken), **language recognition** (detecting which language is being spoken from a set of known languages), **pronunciation verification** (checking that a subject has spoken a given word acceptably).

7.1.6 This course will focus on speaker verification. The underlying principals are also applicable in the other areas.

7.2 Speaker Verification

7.2.1 It is convenient to divide speaker verification into two cases:

7.2.2 **Text dependent** Speaker Verification – where the subject's speech corresponds to a known text. This is typical of applications such as identity verification for access restriction. For example, if speaker verification were used as a biometric for access control to a bank account the user might be expected to use a key phrase, analogous to a 'PIN'. In such an application the subject is cooperative.

7.2.3 **Text independent** speaker verification – where there are no constraints on what the subject might say. This is typical of applications where the subject is not cooperative, may not want to be recognised, and cannot be guaranteed to speak a particular phrase. Applications include forensic applications and security applications.

7.2.4 The first reason for distinguishing between the text-dependent and – independent cases is that text-dependent speaker verification is easier. Clearly, the question 'does this sound like the claimed subject speaking a given utterance' is easier than 'does this sound like the claimed subject'. Secondly, from a technical perspective, word- and phone-level acoustic

modelling techniques (and in particular HMMs) are readily applicable to text-dependent speaker verification.

7.2.5 A variation of text-dependent speaker verification is **text-prompted** speaker verification. This is motivated by the concern that modern high-quality recording and speech synthesis technology might be used to foil a text-dependent system. In text-prompted verification the user is prompted to speak a randomly selected phrase on contact with the system.

7.3 The acoustic model – text-independent verification

7.3.1 In principle it is possible to use phone-level HMM-based techniques for text-independent recognition. One can envisage using phone-level HMM based techniques to simultaneously recognise the speech and the speaker (and some laboratories with expertise in LVCSR, e.g. Dragon Systems, have attempted to do this). However, most text-independent speaker verification systems are based on a Gaussian Mixture Model (GMM).

7.3.2 In such a system, the distribution of acoustic feature vectors for a given speaker is modelled by a Gaussian mixture pdf. Recall that such a pdf has the form

$$b(x) = \sum_{m=1}^{M} w_m b_m(x), \text{ where } 0 \le w_m \le 1, \sum_{m=1}^{M} w_m = 1$$

and each b_m is a multivariate Gaussian pdf with mean μ_m and (co)variance matrix σ_m.

7.3.3 The number of mixture components is typically large (>1024) and adaptation techniques might be used to train a speaker-dependent model using a relatively small quantity of training data and a speaker-independent model.

7.4 The acoustic model – text-dependent verification

7.4.1 In this case it is possible to exploit knowledge of the utterance and construct an appropriate phrase-level model by concatenation of the relevant phone- or word-level models.

7.5 The basic problem and intuitive solution

7.5.1 The basic problem is as follows: We have acoustic data y which, it is claimed, corresponds to a speaker S. We also have a statistical acoustic model M for S. How do we decide whether or not to agree that the data really was spoken by S?

7.5.2 An intuitively obvious solution might be to compute the probability $P(y|M)$, and accept the utterance if $P(y|M)$ is greater than some threshold T. This

threshold will have been decided empirically in advance. Unfortunately this simple approach does not work.

7.5.3 An intuitive explanation for this is that the absolute value of the probability $P(y|M)$ will depend on a range of extraneous factors which have little to do with the identity of the speaker. These include changes in microphone or acoustic environment, variations in the subject's speech due to health and other factors, etc. Ideally the threshold T would need to adapt to accommodate these factors.

7.5.4 More formally, the problem arises because we are trying to make the classification decision based on the *class conditional* probability $P(y|M)$, whereas we should be using the *posterior* probability $P(M|y)$. Of course, these two probabilities can be related using Bayes' Theorem:

$$P(M \mid y) = \frac{P(y \mid M)P(M)}{P(y)}$$

7.5.5 The probability $P(M)$ is the *prior* probability that the utterance was spoken by the authorised subject rather than an impostor, and can be estimated. But how can we calculate the probability $P(y)$?

7.6 The World Model

7.6.1 The probability $P(y)$ is the probability of the acoustic data y, independent of who spoke, and the standard approach to estimating $P(y)$ is to use a **World Model (WM)** (also known as a **General Speaker Model**, or a **Babble Model**, or a **Garbage Model**). The world model is simple a statistical acoustic model of the same type as the speaker-dependent model (i.e. a GMM or a set of HMMs) but trained on a large population of speakers.

7.6.2 Once a World Model has been introduced to estimate $P(y)$, classification can be based on the *posterior probability* $P(\mathrm{M}|\mathrm{y})$. I.e. the subject is authorised if $P(\mathrm{M}|\mathrm{y}) > T$ for some appropriately chosen threshold T (more on this later).

7.6.3 Returning to the earlier intuitive explanation for the inadequacy of $P(y|M)$, note that:

7.6.4 $P(M \mid y) > T$ if and only if $\frac{P(y \mid M)P(M)}{P(y)} > T$ if and only if $P(y \mid M) > \frac{T \times P(y)}{P(M)}$ (*)

7.6.5 In other words, comparing the posterior probability $P(\mathrm{M}|\mathrm{y})$ to a fixed threshold can be thought of as comparing the class conditional probability $P(y|M)$ to an **adaptive** threshold.

7.7 Implementation

7.7.1 Implementation of a real-time speaker verification system based on these principles is straightforward for a GMM system. Given a section of speech, the probabilities of the corresponding acoustic data are calculated based on the speaker-dependent model and the world model. The classification decision is then based on (*).

7.7.2 For a system based on phone-level HMMs, the decoder is realised using the Viterbi decoding techniques described in the section on Automatic Speech Recognition. Basically, Viterbi decoding is applied to a network of the kind depicted below:

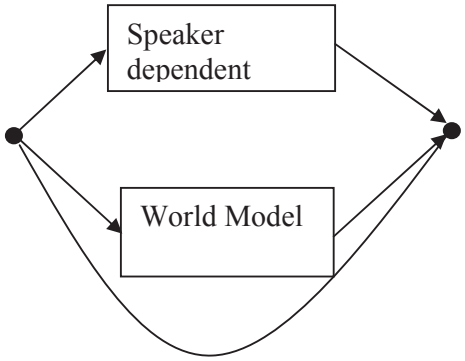


Figure 46: Syntax network for HMM-based speaker verification

7.8 Cohorts

7.8.1 In speech recognition the key problem is to distinguish between similar sounding words. In speaker recognition the corresponding problem is to distinguish between similar sounding speakers. One approach to this problem is the use of speaker **cohorts**. For a given subject, a cohort is a set of speakers who sound most similar to the target speaker. These speakers are typically selected from the database used to train the world model. The cohort can be used to train an additional 'world' model whose purpose is to match speech from impostors who sound similar to the authorised subject. How do you think this might be done?

7.9 Scoring Speaker-Verification Systems

7.9.1 Two measures are of interest when scoring a speaker-verification rate:

7.9.2 The **false acceptance rate** –the percentage of times that an impostor is wrongly accepted.

7.9.3 The **false rejection rate** – the percentage of times that the legitimate subject is rejected.

7.9.4 The tradeoff between false acceptance and false rejection is controlled by the threshold T. If $T=0$ then clearly all utterances will be accepted. The false

acceptance rate will be 100% and the false reject rate will be 0%. Conversely, if *T* is very large, then the false acceptance rate will tend to 0% but the false rejection rate will tend to 100%. Percentage false acceptance and rejection are typically plotted on the same graph (figure 47), or combined into an ROC (Received Operating Characteristic) curve (figure 48).

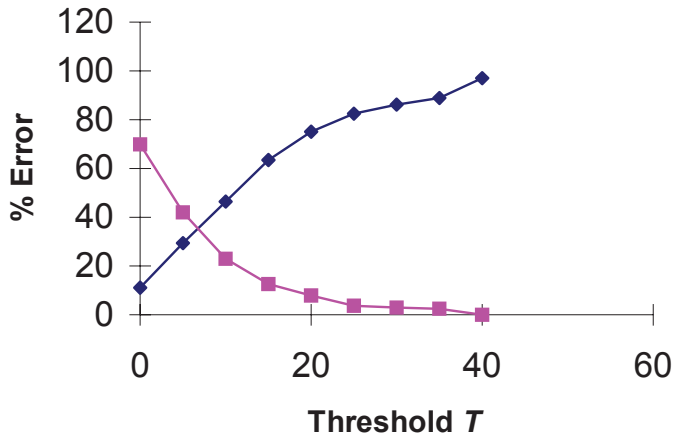


Figure 47: Example plots of false acceptance and false rejection rates for a speaker verification system.

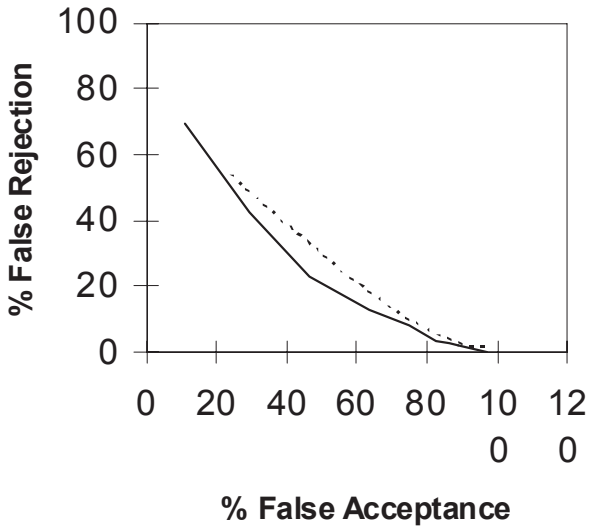


Figure 48: Example ROC curve for a speaker verification system.

8 References

8.1 **Gerry T Altmann (1997), *"The ascent of Babel"*, Oxford University Press**

8.2 **F J Charpentier and M G Stella, "Diphone synthesis using an overlap-add technique for speech waveforms concatenation", Proc. ICASSP, pp 2015-2018, Tokyo, 1986.**

8.3 **Peter B Denes and Elliot N Pinson (1993), *"The speech chain - the physics and biology of spoken language",* Second Edition, W H Freeman and Co, New York**

8.4 **M Edgington, A Lowry, P Jackson, A P Breen and S Minnis (1996a), " Overview of current text-to-speech techniques: part 1 - text and linguistic analysis", BT Technology Journal, Vol. 14, No. 1, pp 68 - 83.**

8.5 **M Edgington, A Lowry, P Jackson, A P Breen and S Minnis (1996b), " Overview of current text-to-speech techniques: part 2 - prosody and speech generation", BT Technology Journal, Vol. 14, No. 1, pp 84 - 99.**

8.6 **Geoffrey Finch (1998), "How to study linguistics", MacMillan.**

8.7 **D B Fry (1979), "The Physics of Speech", Cambridge Textbooks in Linguistics**

8.8 **J N Holmes, I G Mattingly and J N Shearme (1964), "Speech synthesis by rule", Language and Speech, 7, pp. 127-143.**

8.9 **J N Holmes (1973), "The influence of glottal waveform on the naturalness of speech from a parallel formant synthesizer", IEE Transactions on Audio and Electroacoustics, 21, pp. 298-305.**

8.10 **J N Holmes (1988), "Speech synthesis and recognition", Van Nostrand Reinhold (UK). (Currently out of print, new edition expected later this year)**

8.11 **J N Holmes and W J Holmes (2001), "Speech synthesis and recognition: 2nd edition", Taylor & Francis**

8.12 **W J Holmes (1989), "Copy synthesis of female speech using the JSRU parallel formant synthesiser", Proc. EUROSPEECH'89, Paris, pp 513-516.**

8.13 **A J Hunnicutt and D Klatt (1987), "From text to speech: the MITalk system", Cambridge University Press.**

8.14 **S E Levinson (1983), "An introduction to the theory of ", Bell System Technical Journal**

8.15 **A Newell, (1986), "Speech Understanding Systems",**

APPENDIX

A Basic Probability Theory for Speech Recognition

8.16 The sample space

A.1.1 In probability theory, the set of all possible outcomes, or observations, is called the sample space and will be denoted by Ω. Each possible outcome is a sample. An event $A \subseteq \Omega$ is a set of samples[2].

A.1.2 For speech recognition, or most other pattern recognition problems, we are concerned with just one type of sample space: Ω is **discrete** and **finite**. The members of Ω depend on the particular problem that we are trying to solve. If we are doing **sentence** recognition, then Ω is the set of all sentences. If we are doing **word** recognition, then Ω is the set of all words. If we are doing **phoneme** recognition, then Ω is the set of all phonemes. In general, Ω is the set of classes that we are trying to recognise.

A.1.3 To simplify notation and subsequent discussions, and to remind you that we are talking about pattern recognition rather than general statistics, I'm going to call a member of Ω a **class**.

8.17 Probability measures

A.1.4 A **probability measure** (or **probability function**) is a function P defined on the set of all events which satisfies the following conditions:

A.1.4.1 $0 \le P(A) \le 1$

A.1.4.2 $P(\Omega) = 1$

A.1.4.3 If A and B are mutually exclusive events (i.e. $A \cap B = \varnothing$), then $P(A \cup B) = P(A) + P(B)$. More generally, if A and B are not mutually exclusive, $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.

A.1.4.4 You can think of $P(A)$ as the probability that a sample lies in the set A.

A.1.5 The fact that our sample space Ω is discrete and finite makes life a lot easier. For each class w in Ω, the set $\{w\}$ consisting of the single point w, is a measurable set. In other words we can write $P(\{w\})$ and talk about **"the probability of *w*"**. In this case it is more usual to write $P(w)$ rather than $P(\{w\})$.

---

[2] In fact, only some subsets are events. The sets A for which $P(A)$ is defined are called the **measurable sets**. In general, not all subsets of the sample space are measurable, but we don't need to worry about this, because in a few moments we are going to restrict to a particularly simple case.

8.18 The *prior* probability of a class

A.1.6 The probability $P(w)$ is called the **prior probability** (or ***a priori* probability**) of the class w.

A.1.7 The reason for this name is straightforward. In general we will be given some sort of measurement and asked to determine which class the measurement is in. The prior probability of a class is the probability of the class **before** any such measurements have been made (hence 'prior'). For example, in speech recognition the prior probability of a word is the probability of that word being spoken before any measurements of any acoustic signals have been made.

8.19 Random variables

A.1.8 As mentioned above, in pattern recognition the problem is to determine the class which has given rise to a particular set of **measurements**. In speech recognition, a class might be a word w, and the corresponding measurement is the electrical signal s which comes from a microphone.

A.1.9 If every time anyone spoke the word w the exact same signal s was produced from the microphone, speech recognition would be a very simple problem indeed and we could all go home and grow vegetables. Unfortunately this is not the case, and it is very rarely the case in any physical pattern recognition problem. In other words the function, f say, which maps classes to physical measurements, $f(w)=s$, is **not well-defined**, because for fixed w there are lots and lots of possible values of s. The mathematical notion which was invented to address this problem is the concept of a **random variable**.

A.1.10 By saying that f is a random variable we are acknowledging the fact that $f(w)$ can take on many different values, and we are assuming that these values are determined by a probability measure.

8.20 Feature extraction

A.1.11 Deducing the identity of a word directly from the electrical signal which comes out of the microphone is not a very fruitful approach. Typically we process this signal to convert it into a form which is more suitable for recognition. Typically we try to suppress properties of the signal which are not important for recognition, and emphasise those which are. This process is normally called **feature extraction**.

A.1.12 Typically one, scalar value is not enough to classify a measurement. Instead we need to extract a whole set of numbers from the raw signal. These numbers are normally represented as the components of a **vector**, called a **feature vector**.

A.1.13 So, in pattern recognition, random variables normally associate classes with vectors, rather than with scalars. In other words we will need to be able to deal with **vector valued random variables.**

8.21 Statistical speech recognition

A.1.14 Lets take a moments break here to examine where we have got to. The basic problem is, as we have already seen, that two utterances of the same word, even if they are by the same speaker, will never give rise to exactly the same sequence of feature vectors. This variability is what makes speech recognition (or any other serious pattern recognition problem, for that matter) difficult.

A.1.15 Now it is absolute nonsense to assert that this variability is really random. Each instantiation of a particular word is determined by a set of neurological processes, which ultimately control the dynamics of the vocal tract, lungs, and the rest of the speech production system. However, at the moment understanding these processes is completely beyond us (and everyone else, for that matter) and so we make two huge **simplifying assumptions**.

A.1.16 **First Huge Simplifying Assumption**: we are going to assume that the process which associates a word with a sequence of acoustic vectors is a sequence of **random variables**. In other words, although there is a notional acoustic 'target' for the word, its actual realisation as a sequence of acoustic vectors it is governed entirely by probabilities (in fact we are going even further and assuming that the choice of the sequence of words which is spoken is also governed entirely by probabilities).

A.1.17 **Second Huge Simplifying Assumption**: although a spoken utterance is realised as a sequence $y=y_1,...,y_t,...,y_T$ of acoustic vectors, we are going to assume that, initially at least, we can treat these vectors **individually**.

A.1.18 So, as a first step, we need to revise basic probability theory as it applies to random variables which take their values in N dimensional vector spaces. We will start with $N=1$ and the simplest form of random variable that we are going to deal with – one whose values are determined by a Gaussian (or normal) 'distribution'.

8.22 Probability density functions

A.1.19 Suppose we have a class w, an associated random variable f which takes scalar values, and a scalar measurement x. You probably think that we would like to talk about the probability $P(f(w)=x)$, that is, the probability that the measurement x occurs in response to the class w.

A.1.20 Unfortunately (or fortunately, depending on whether you are inclined towards simplicity or elegance) the scalar x typically belongs to the continuous set of real numbers. In this case $P(f(w)=x)=0$. Intuitively this is because the set $\{x\}$

is an infinitesimally small subset of the real line, and the probability that $f(w)$ takes on exactly the value x is zero[3].

A.1.21 If we want to talk about probabilities for random variables which take real number values, then we have to talk about quantities like $P(a \leq f(w) \leq b)$, the 'probability that $f(w)$ lies between a and b' rather than $P(f(w)=x)$, 'the probability that $f(w)$ equals x'. To do this we need the notion of a **probability density function** or **PDF**.

A.1.22 In the case of random variables which take real scalar values, a probability density function is just a function p defined on the real line such that:

A.1.22.1 $0 \leq p(x) \leq 1$ for all real numbers x , and

A.1.22.2 $\int_{-\infty}^{\infty} p(x)dx = 1$

A.1.23 In this case, if the random variable f is governed by the PDF p then

$$P(a \leq f(x) \leq b) = \int_{a}^{b} p(x)dx$$

A.1.24 If a random variable f is governed by a PDF p, then the **expected value** of f, denoted by $E[f]$, is given by: $E[f] = \int_{x} xp(x)dx$

A.1.25 The most common probability density function is the **Gaussian** or **normal** PDF.

8.23 The Gaussian Distribution

A.1.26 A **Gaussian** PDF, with **mean** μ and **variance** σ^2 is the PDF p defined by

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma} \right)^2 \right]$$

and we write $p(x)=N_{(\mu,\sigma 2)}(x)$

A.1.27 In the case where $\mu=0$ and $\sigma=1$, p is referred to as the **standard Gaussian probability density function (PDF)** or **standard Gaussian density**. In this case

---

[3] More formally it is because $\{w\}$ is what mathematicians call a "set of measure zero"

$$p(x) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2} x^2 \right]$$

and we write p=N(0,1)

A.1.28 It is straightforward to show that if f is governed by a standard Gaussian PDF then the random variable g defined by $g(x)=\mu+\sigma f(x=x)$ is governed by a Gaussian PDF with mean μ and variance σ^2.

A.1.29 Figure P1 shows the density function p for a random variable $f = N_{(0,4)}$ which is governed by a Gaussian PDF with mean 0 and variance 4. For two real number a and b, the probability $P(a \leq f(w) \leq b)$ is just the area under this curve between a and b.

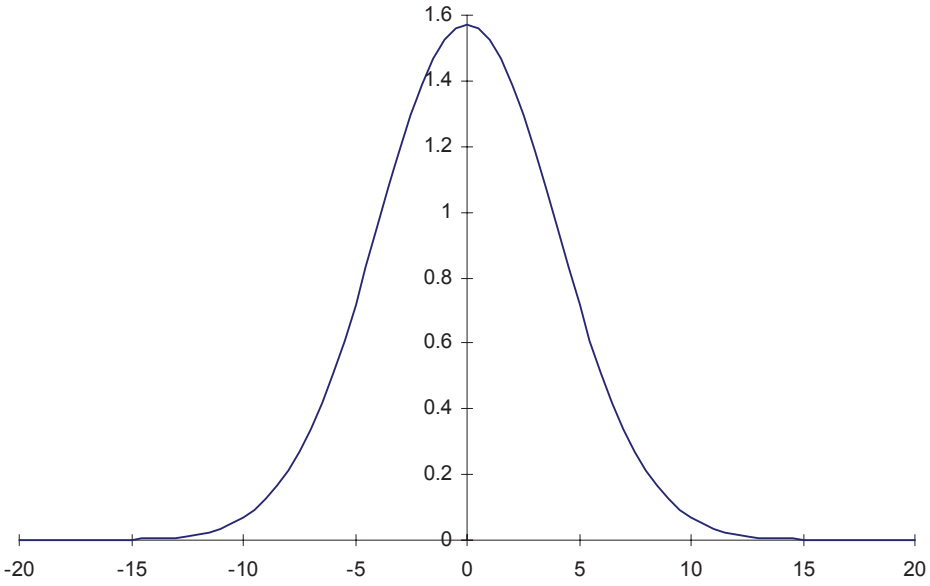


Figure P1: Gaussian density for a $N_{(0,4)}$ random variable

8.24 The Multivariate Gaussian Distribution

A.1.30 To exend the notion of a Gaussian distribution to N dimesional space, we need to begin by considering N scalar valued random variables (one for each dimension) $f_1,...,f_N$, each conforming to the standard Gaussian distribution. If these PDFs are mutually independent, then their **joint** density is just the product of the individual densities for the N separate dimensions and is given by:

$$p(x_1,...,x_N) = \prod_{n=1}^{N} p(x_n) = \frac{1}{(2\pi)^{N/2}} \exp\left\{ -\frac{1}{2} \sum_{n=1}^{N} x_n^2 \right\}$$

A.1.31 Now suppose that $f=(f_1,\ldots,f_N)$ is a vector valued random variable taking values in N-dimensional real space. Let A be an $N\times N$ matrix (which you should think of as a transformation on N dimensional space).

A.1.32 The vector valued random variable g defined by $g(w)=\mu+Af(w)$, where μ is an N dimensional vector, has density

$$p(y)=\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp\left\{-\frac{1}{2}(y-\mu)^{*}\Sigma^{-1}(y-\mu)\right\}$$

A.1.33 This is the **multivariate** Gaussian density. The matrix Σ is called the covariance matrix of Y and is equal to A^*A.

A.1.34 A vector valued random variable which conforms to a Gaussian PDF with mean vector μ and covariance matrix Σ is denoted by $N_{(\mu,\Sigma)}$

8.25 Covariance and covariance matrices

A.1.35 Let's suppose that N=2 and that the 2-dimensional Gaussian PDF p occurs as the product of two standard 1-dimensional Gaussian PDFs:

$$p(x_1,x_2)=p(x_1)p(x_2)=\frac{1}{2\pi}\exp\left\{-\frac{1}{2}\left(x_1^2+x_2^2\right)\right\}$$

A.1.36 Figure P2 is a schematic plot of values taken by a random variable f governed by this PDF

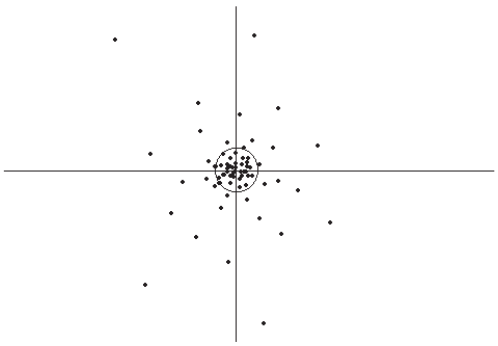


Figure P2: Schematic diagram of points in 2 dimensional space produced by the product of two standard 1 dimensional Gaussian PDFs

A.1.37 Figure P3 is a schematic plot of values taken by a random variable g defined by $g(w) = \mu + f(w)$

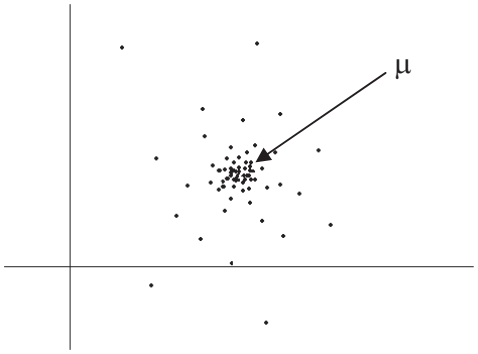


Figure P3: Schematic plot of values taken by the random variable g

A.1.38 Figure P4 is a schematic plot of values taken by a random variable g defined by $g(w) = \mu + Af(w)$, where A is the 2×2 matrix $\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

A.1.39 Note that in figure P4 the 2 dimensional PDF is still a product of 2 1-dimensional PDFs. The change is that while the PDF in the 'x' direction is still a standard Gaussian PDF, the PDF in the 'y' direction is a Gaussian PDF with variance 4 (because this is the second diagonal element of A^*A).

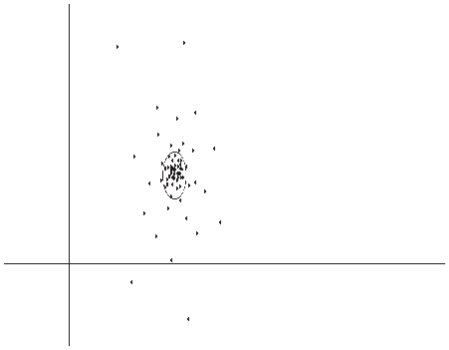


Figure P4: Schematic plot of values taken by the 2-dimensional random variable $g(w) = \mu + Af(w)$, *where A is the 2 × 2 matrix* $\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

A.1.40 Finally, let $g(w) = \mu + Af(w)$ where A is the matrix $A = \begin{bmatrix} \cos\theta & 2\sin\theta \\ -\sin\theta & 2\cos\theta \end{bmatrix}$ where $\theta = \frac{\pi}{4}$. Then A corresponds to scaling by 2 in the 'y' coordinate followed by a clockwise rotation through 45°. A schematic plot of values taken by $g(w)$ is shown in figure P5.

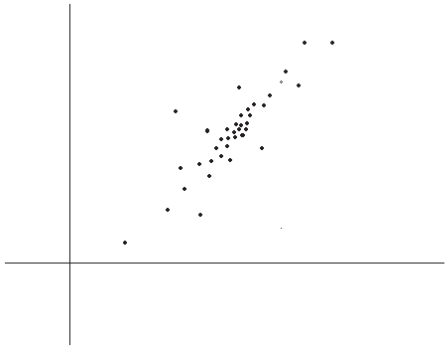


Figure P5: Schematic plot of values taken by the 2-dimensional random variable $g(w) = \mu + Af(w)$, *where A is the* 2×2 *matrix* $A = \begin{bmatrix} \cos\theta & 2\sin\theta \\ -\sin\theta & 2\cos\theta \end{bmatrix}$ *where* $\theta = \pi/4$

A.1.41 In this case the covariance matrix becomes

$$\Sigma = A^* A = \begin{bmatrix} \cos^2\theta + 4\sin^2\theta & 3\sin\theta\cos\theta \\ 3\cos\theta\sin\theta & \sin^2\theta + 4\cos^2\theta \end{bmatrix}.$$

A.1.42 As before the diagonal elements of the covariance matrix are the variances in the x and y directions. The non-zero off-diagonal elements are the **co-variance** between the *x* and *y* components of the values taken by the random variable. These off-diagonal elements are zero if and only if the *x* and *y* components are independent.

A.1.43 This is evident from figures P2, P3, P4 and P5. In the first three figures the *x* and *y* components are independent. For example, in figure P4 imagine walking along the *y* axis. As you move away from the *y* component of the mean, the 'density' of data points will change, but the underlying distribution of *x* values will remain unaltered. This contrasts with figure P5, where as the *y* component increases the average value of the *x* component increases.

A.1.44 If the non-diagonal entries of the covariance matrix were negative, then the average *x* value would decrease as the *y* value increased.

8.26 Gaussian Mixture Distributions

A.1.45 The final form of PDF which is important for speech pattern processing we will look at is the **Gaussian mixture PDF**.

A.1.46 Gaussian distributions are often used in pattern recognition problems. They are mathematically simple, their parameters are easy to estimate, and in many cases they provide an acceptable model of the actual variation which is observed. However, there are many cases where the simple uni-modal shape of a Gaussian density is not appropriate. This may be because the actual

distribution of the data is multi-modal, or the actual distribution of the data is uni-modal, but does not conform to the simple Gaussian 'bell-shape'

A.1.47 For example, consider the problem of building a pattern recognition system to recognise a person from measurements of their speech. One could attempt to build a Gaussian model of these measurements, in which case the mean of the distribution for a particular individual might reflect some general physiological property, such as the dimensions of their vocal tract.

A.1.48 However, 'speech' is made up of several sub-classes, corresponding to the basic sounds of the language. Hence, to a first approximation, it is likely that the actual distribution of measurements will be multi-modal, with different modes corresponding to different basic sounds.

A.1.49 A Gaussian mixture density is a multi-modal extension of The Gaussian density. An ***M*-component Gaussian mixture density** p has the form:

$$p(w) = \int_{m=1}^{M} w_m p_m(w)$$

where $0 \le w_m \le 1, \sum_{m=1}^{M} w_m = 1$ and $p_m = N_{(\mu_m, \Sigma_m)}$.

A.1.50 Figure P6 shows and example of a 3 component, 1 dimensional Gaussian mixture PDF.

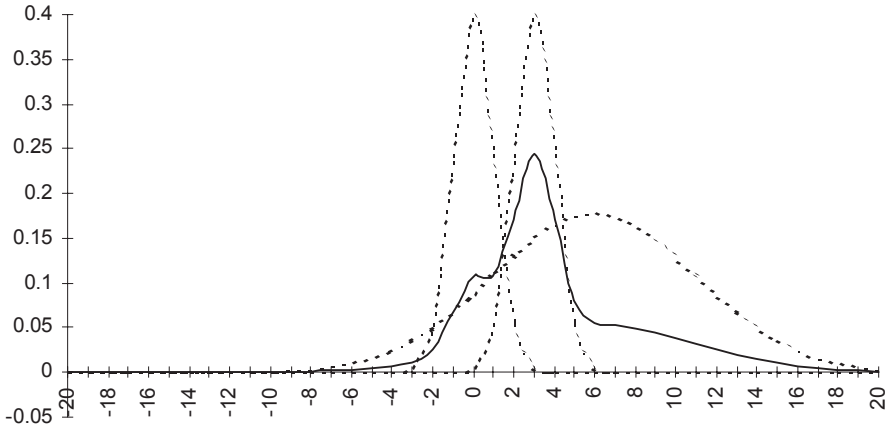


Figure P6: 3-component Gaussian mixture density (solid line), with component parameters μ_1=0, σ_1=1, w_1=0.2; μ_2=3, σ_2=1, w_2=0.5; μ_3=6, σ_3=5, w_3=0.3 (broken lines)

8.27 **Class conditional PDFs**

A.1.51 Suppose that w is an element of the sample space (i.e. w is a class). In a pattern recognition problem we are specifically interested in knowing the

distribution of measurements which occur for an individual class w. Consequently, rather than being interested in the probability density function which describes the values which our random variable takes over the whole sample space, we are specifically interested in the PDF which describes the values which it takes for w.

A.1.52 If our random variable is f we are specifically interested in probabilities such as $P(a \leq f(w) \leq b)$, **given the identity of the class *w***.

A.1.53 First let's simplify notation before it gets more complicated. The random variable f represents the process whereby a symbol, w, gets converted into a measurement $f(w)$. In other words f is fixed. Consequently we drop it from the notation and write:

$P(a \leq x \leq b)$ instead of $P(a \leq f(w) \leq b)$

A.1.54 The **conditional probability** that the measurement x lies between a and b, given that x is an instantiation of the class w, is denoted by:

$P(a \leq x \leq b|w)$

8.27.1 In patter recognition, $P(a \leq x \leq b|w)$ is referred to as the **class conditional probability**.

A.1.55 The corresponding PDF is then denoted by $p(x|w)$ and is called the **class conditional PDF for the class *w***. In other words, the class conditional PDF for the class w describes the distribution of measurements which arise as a consequence of the class w.

8.28 The posterior probability

A.1.56 Our final probability is the posterior probability of the class w. This probability is key for pattern recognition. It is the probability of the class w **given that the measurement *x* has been observed**. The posterior probability of the class w is denoted by $P(w|x)$.

A.1.57 Note that this really is a probability function, and not a PDF.

8.29 Bringing it all together: Bayes' Theorem

A.1.58 While calculations of the class conditional density $p(x|w)$ and the prior probability $P(w)$ are both easy, in principle, direct calculation of the posterior probability is not. Luckily we don't need to calculate it directly, because we can use **Bayes' Theorem**:

$$P(w \mid x) = \frac{p(x \mid w)P(w)}{p(x)}$$

A.1.59 This is the particular version of Bayes' Theorem that we need for statistical pattern recognition. Notice that it combines probabilities (denoted with an upper case '*P*') and densities (denoted with a lower case '*p*').

A.1.60 The probability $p(x)$ which is the denominator on the right-hand side of this equation needs a bit of attention. If we have a finite number of classes, $w_1, \ldots, w_C$ and the goal is to decide which class has given rise to the measurement x then we could use Bayes' Theorem to calculate $P(w_c|x)$ for each c and then assign x to the class w_c for which $P(w_c|x)$ is biggest (more on this later). In this case, since $p(x)$ is independent of w_c, we can ignore it and just maximise the numerator

$$P(w_c \mid x) \propto p(x \mid w_c)P(w_c)$$

8.29.1 This is the solution to the **classification** or **recognition problem**. In speech recognition (or isolated word recognition, anyway) the classification problem is "Which of a finite number of known words was spoken?"

A.1.61 However, we may be interested in a different type of problem, which we will call a **verification** problem.

A.1.62 The corresponding **verification** problem is "was the word *w* spoken?".

A.1.63 The difference is that in the classification problem we are only interested in finding which class w_c has the biggest probability. We're not bothered about the absolute value of that probability. $P(w_c|x)$ might be really small, but as long as it's the biggest that's all that matters. In the verification problem we want to know if the absolute value of the probability $P(w_c|x)$ is big enough.

A.1.64 To determine this we need to calculate the numerator **and** the denominator in Bayes' Theorem.

8.30 Calculation of *p*(*x*)

A.1.65 Calculation of $p(x)$ can be achieved as follows. The density $p(x)$ determines the probability of particular subsets of measurements irrespective of any particular class. However, x must be an instantiation of one of the classes, and all of the classes are mutually exclusive. Hence a basic rules for calculating probabilities applies and we can write:

$$p(x) = \sum_{c=1}^{C} p(x \mid w_c)P(w_c)$$

A.1.66 Bayes' Theorem then becomes:

$$P(w_c \mid x) = \frac{p(x \mid w_c)P(w_c)}{\sum_{c=1}^{C} p(x \mid w_c)P(w_c)}$$

8.31 **Why is Bayes' Theorem important for Pattern Recognition?**

A.1.67 Suppose that you have a friend who regularly goes fishing in a lake *L*. By keeping a record of his or her catches over a period of time it would be possible to build a statistical model which could be used to compute the probability density $p(c|L)$ of a given catch *c* from the lake *L*. The catch *c* might be a vector detailing the number of each species which were caught, or the total weight of each species caught. Of course, we would probably opt to ignore variations due to season, weather etc, and we might assume that the probability of catching one fish of one species is independent of all other catches.

A.1.68 This is interesting, and it might even be fun to do. A certain amount of pleasure and smugness might be got from remarking that you were "not at all surprised" by a particular catch. However, this is tangential to the use of statistics in pattern recognition. The corresponding **pattern recognition** problem would involve several lakes, $L_1,...,L_N$, and the aim would be to identify which lake had been fished, given the catch. How would we do that?

A.1.69 Our first attempt might be to build *N* models, corresponding to $L_1,...,L_N$, and to compute $p(c|L_n)$ for each *n*. Our guess would be that the catch was made at the lake L_n for which $p(c|L_n)$ was greatest.

A.1.70 But what if our friend had a preference for fishing one particular lake? The fact that $p(c|L_n)$ is maximal would be compromised by knowledge that our friend seldom, if ever, fishes L_n. We need to include the **prior** probability $P(L_n)$.

A.1.71 We are really interested in $P(L_n|c)$ and not $p(c|L_n)$.

A.1.72 In pattern recognition, one of the most common uses of Bayes' Theorem is in the context of classifying such a measurement *c* into one of a finite number of classes $L_1,...,L_N$. In this case the relevant form of Bayes' Theorem is:

$$P(L_n \mid c) = \frac{p(c \mid L_n)P(L_n)}{p(c)} = \frac{p(c \mid L_n)P(L_n)}{\sum_{n=1}^{N} p(c \mid L_n)P(L_n)}$$

A.1.73 Notice that the numerator of the right-hand term includes the prior probability $P(L_n)$. Also, for fixed *c* the denominator is constant and can be ignored if the goal is to find L_n which maximises $P(L_n|c)$.

8.32 **Parameter Estimation**

A.1.74 In order to apply the ideas that have been developed above, we need to find ways to estimate the various probability density functions and probability functions which appear on the right-hand side of Bayes' Theorem. In fact, only two quantities need attention, the class conditional density $p(x|w)$ and the prior probability $P(w)$.

A.1.75 For the class conditional density we are going to use a Gaussian mixture PDF, which is determined by its **parameters** (i.e. its mean vectors and covariance matrices, and its mixture weights). Let's denote this set of parameters by φ. Once the parameter set φ is fixed, then we are going to define:

$$p(x \mid w) = p(x \mid \varphi)$$

A.1.76 How do we choose the 'best set of parameters' φ?

A.1.77 Given a set of example measurements $x_1,\ldots,x_S$ corresponding to the class w, we could assume that the x_ss are independent and try to find the set of parameters φ which maximises the function

$$p(x_1,\ldots,x_s \mid \varphi) = \prod_{s=1}^{S} p(x_s \mid \varphi)$$

A.1.78 The quantity $L(\varphi) = p(x_s|\varphi)$, thought of as a function of the parameter set φ, is called the **likelihood of** φ, and the procedure of choosing the parameter set φ which maximises the likelihood function $L(\varphi)$ is called **maximum likelihood estimation of the parameter set** φ.

A.1.79 For example, suppose that we are given a set of scalar measurements $x_1,\ldots,x_S$ corresponding to a class w and we want to represent w as a unimodal 1 dimensional Gaussian PDF. The parameters of such a PDF are its mean μ and variance σ^2. Then it is easy to show that the maximum likelihood estimates of μ and σ^2 are given by:

$$\mu = \frac{1}{S}\sum_{s=1}^{S} x_s \quad \text{and} \quad \sigma^2 = \frac{1}{S}\sum_{s=1}^{S}(x_s - \mu)^2$$

A.1.80 In other words, the maximum likelihood estimates of μ and $\sigma 2$ are the sample mean and sample variance respectively.

A.1.81 A corresponding result holds for multivariate Gaussian PDFs.

A.1.82 However, for Gaussian mixture PDFs the problem is more complex. There is no direct analytical expression for the maximum likelihood estimates of the parameters of a Gaussian mixture PDF in terms of a set of training data. Instead some form of iterative procedure must be used. The most common

solution is to use a general technique called the **Expectation-Maximisation**, or **E-M algorithm**.

8.33 Dangers of insufficient training data

A.1.83 A major practical problem in maximum likelihood parameter estimation is **under training**. This is most easily illustrated by an example.

A.1.84 Suppose that some measurement process applied to a class w gives rise to measurements which are uniformly distributed over the interval [0,1]. However, unfortunately we do not know that this is the true distribution and so we try to model the distribution using a Gaussian mixture PDF. So, we obtain a training set comprising N samples $x_1,\ldots,x_N$.

A.1.85 Consider the problem of trying to fit an N component Gaussian mixture PDF to this data. The maximum likelihood estimate of the parameters of the mixture PDF will set the N means equal to the N samples, and make the N variances as small as possible (figure P7(a) shows a simple example for N=4).

A.1.86 In fact, in this case the maximum likelihood estimate of the parameters of a unimodal, conventional Gaussian PDF would provide a better fit to the true PDF (figure P7(b)).

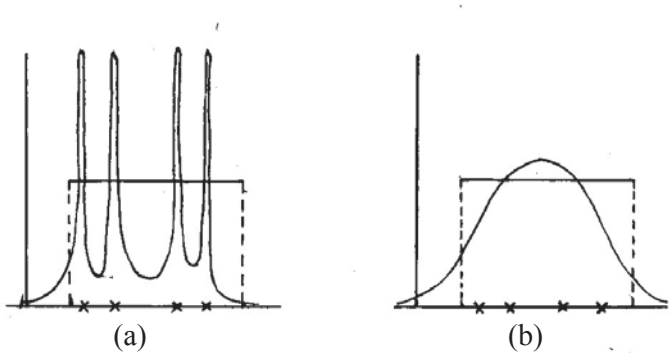


Figure P7: Illustration of maximum likelihood fit of 4 component (a) and 1 component Gaussian mixture PDF to a training set of just 4 points

A.1.87 The problem here is that the PDF in P7(a) does not **generalise to unseen data**. If a new measurement is taken, that new measurement will correspond to a very small value of the PDF in P7(a) and is unlikely to be classified properly

A.1.88 The basic issues here are as follows:

A.1.88.1 Given a finite training set X, and a maximum likelihood estimate M of the parameters of a statistical model, $p(X|M)$ will increase, in general, as the number of parameters in M increases.

A.1.88.2 As the number of parameters increases, the statistical model begin to characterise detail in the training set which is not present in unseen data. The model begins to "remember the training set".

A.1.88.3 As the number of parameters increases, the performance of the resulting classifier will improve at first, as the increased number of parameters allows better modelling of those properties which are shared between the training data and unseen data, but will then start to degrade as the number of parameters increases and the model focuses on specific detail in the training set (figure P8).

A.1.89 This phenomenon is referred to as **under-training**, because it could be overcome by the use of more training data.

A.1.90 The solution which is normally adopted is that, during the development of a pattern recognition system, all of the available data is divided into three sets: the **training set**, the **evaluation set** and the **test set**.

- For each fixed number of parameters, the maximum likelihood estimate of the parameter set is made using the training set
- Classification experiments are then run on the evaluation set, and the number of parameters which gives the best performance is chosen for the final system
- This system is then evaluated using the test set

A.1.91 In this way the test set result is in no way compromised, since the test set is not used to set the parameters or to determine how many parameters there should be.

8.34 Bayes' Decision Theory

A.1.92 So far the rule that a measurement x should be assigned to the class w which maximises $P(w|x)$ has been justified intuitively. However, its formalisation is part of Bayes' Decision theory.

A.1.93 If we have a measurement x and we know the posterior probabilities $P(w_n|x)$ for each $n=1,...,N$, then it is natural to assign y to class w_n if

$$P(w_n|x) > P(w_k|x), \quad k=1,..,N, \; k \neq n.$$

A.1.94 This decision rule partitions the space of measurements X into N regions $X_1,...,X_N$ such that if $x \in \Omega_n$ then x is assigned to class w_n.

A.1.95 We have already seen that Bayes theorem can be used to express the posterior probabilities in terms of the prior probabilities and the class conditional probabilities, namely:

$$P(w_n \mid x) = \frac{p(x \mid w_n)P(w_n)}{p(x)}$$

hence the above decision rule based on posterior probabilities can be re-written in terms of class conditional and prior probabilities. The measurement y is assigned to class w_n if

$$p(x \mid w_n)P(w_n) \geq p(x \mid w_k)P(w_k), k = 1,..,N$$

A.1.96 This is know as **Bayes' rule for minimum error**.

8.35 The likelihood ratio

A.1.97 For two classes this rule may be re-written as

$$l(x) = \frac{p(x \mid w_1)}{p(x \mid w_2)} > \frac{P(w_2)}{P(w_1)}$$

A.1.98 The function $l(y)$ is called the **likelihood ratio**.

A.1.99 Figure P8 shows the functions $p(x|w_n)P(w_n)$ for $n=1,2$ in the case where w_1 is Gaussian distributed $N_{(0,1)}$, w_2 is $N_{(6,5)}$, $P(w_1)=0.6$ and $P(w_2)=0.4$.

A.1.100 Figure P9 shows the corresponding likelihood function together with the threshold $\frac{P(w_2)}{P(w_1)}$

A.1.101 In both cases it can be seen that the decision rule leads to a disjoint region for class w_2.

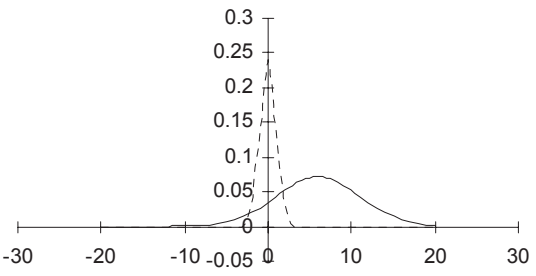


Figure P8: Plots of $p(x|w_n)P(w_n)$, n=1,2.

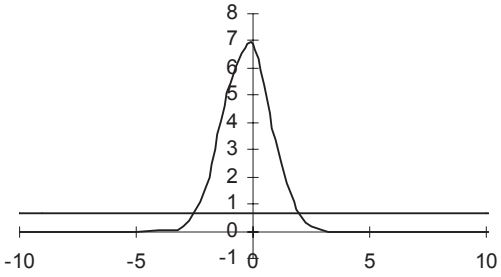


Figure P9: Plot of the likelihood ratio l(y) and threshold $P(w_2)/P(w_1)$

A.1.102 To see that these decision rules minimise the error, note that

$$P(error) = \sum_{n=1}^{N} P(error \mid w_n)P(w_n)$$

But

$$P(error \mid w_n) = \int_{x \notin \Omega_n} p(x \mid w_n)dx$$

So, the probability of an error can be written as follows:

$$P(error) = \sum_{n=1}^{N} \int_{x \notin \Omega_n} p(x \mid w_n)P(w_n)dx$$

$$= \sum_{n=1}^{N} P(w_n)\left(1 - \int_{\Omega_n} p(x \mid w_n)dx\right)$$

$$= 1 - \sum_{n=1}^{N} P(w_n) \int_{\Omega_n} p(x \mid w_n)dx$$

A.1.103 Hence minimising the probability of error is equivalent to maximising the probability of correct classification:

A.1.104 Therefore we need to choose the regions Ωn to maximise the integral

$$\sum_{n=1}^{N} P(w_n) \int_{\Omega_n} p(x \mid w_n)dx$$

Of course, this happens if Ωn is the region where p(x|wn)P(Cn) is a maximum over n, which corresponds to Bayes' rule for minimum error.

The probability of correct classification *c* is

$$c = \int_x \max_n P(w_n)\, p(x \mid w_n) dx$$

and the Bayes error is

$$e_B = 1 - \int_x \max_n P(w_n)\, p(x \mid w_n) dx$$

END