TOWARDS THE USE OF SUB-BAND PROCESSING IN AUTOMATIC SPEAKER RECOGNITION

Robert Andrew Finan

A thesis submitted in partial fulfilment of the
requirements of the University of Abertay Dundee
for the degree of Doctor of Philosophy

July 1998

I certify that this thesis is the true and accurate version of the thesis approved by the examiners.

Signed .....[redacted]..... Date 20-December-98

[redacted]irector of Studies)

ABSTRACT

Automatic speaker recognition uses a person's voice as a means of identifying them. It has many practical applications, particularly in the area of security systems. The thesis investigates the application of neural networks (both classifying and predicting) to the problem, as well as offering a new alternative to the current wide-band approach to speaker recognition.

As part of the work, a new method of ranking a speaker's impostors is presented. It involves creating a vector quantisation (VQ) model for each potential impostor and testing this impostor model against the genuine speaker's VQ model. The ranking of these *model vs. model* scores is indicative of the final ranking of the impostors, as determined by the test results. Previous methods of ranking impostors required testing the genuine speaker model with every impostor's training utterances, which requires considerably more computation than this new method which only needs one test per impostor.

Impostor ranking has potential applications in score normalisation for models which don't use discriminative training. Such models include predictive neural networks (PNN) and vector quantisation, which both produce a score which is the distance of the test utterance from the model. These scores may be normalised by comparing them to a distance from an anti-speaker model. Using the *model vs. model* impostor ranking, a cohort of impostors may be selected for each speaker to represent the anti-speaker model. A normalisation method, based on cohorts of these ranked impostors, markedly improved the verification error rates of distance models for both text-dependent and text-independent conditions.

Further reductions in error rates may be achieved through the use of information complementary to the linear prediction cepstrum coefficients (LPCC). When the scores from a recogniser based on the LP residual are used in conjunction with those from an LPCC recogniser they lead to a drop in the identification error rate.

This concept of combining results from recognisers which focus on complementary areas of the speech signal is further developed in an approach known as sub-band processing. The sub-band processing implemented for this work is new to the field of automatic speaker recognition. It focuses on different regions of the speech signal by splitting it into 16 sub-bands based on the mel-scale, each with its own dedicated recogniser. The individual sub-bands emphasise the spectral properties of the band-limited frequency ranges, and this is reflected in the make-up of the cepstral coefficients for each sub-band. This provides a more detailed model of the speaker than the wide-band approach, which uses a single model to cover all frequencies. The final score for a test utterance is determined by combining the scores from the different sub-bands. The sub-band processing approach made significant improvements on the error rates of the wide-band system.

Acknowledgements

I would like to thank the following people for their contribution to this work or their preservation of my sanity during the course of it:

- Andy Sapeluk for acquiring the funding without which there would have been no PhD, for his encouragement when all appeared lost (which seemed to be quite often) and for his all-round commitment throughout this work.
- Dr. Bob Damper for helping me with the writing of several papers, as well as for his encouragement of my research.
- Dr. Malcolm Hannah for many enlightening discussions on automatic speaker recognition and the rôle of spiders in the bath, not to mention some masterful banjo playing.
- My wife Julia for her support and understanding through what has often been a trying time.
- My parents for having made it all possible in the first place!
- And not forgetting... Murphy's Irish Stout (*"when all looks black as the hour of night, a pint of plain is your only man!"*), The Bland Brothers (*mar ná beidh a leithéidí arís ann!*), Harry and Hazel, Mike (all things technical), Gernot for everything in Wien, Jim (evolution and other science fiction), the Philosophy Group and Scotland for great hillwalking!

Go raibh míle maith agaibh go léir!

Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Why automate speaker recognition?

For over a quarter of a century, there has been considerable interest in the use of speech as a means of recognising a person or confirming their identity (Furui, 1994). If anything, this interest has increased with the recent advances in computing power and the application of techniques, such as hidden Markov models (HMM) and artificial neural networks (ANN), to the speaker recognition problem (Furui, 1997).

There are two main reasons for this interest in automating speaker recognition. The first is man's perpetual desire to see if he can duplicate or better human performance through automation (Atal, 1976). In the case of speaker recognition, it is to develop a computerised system that can recognise a person from their voice with equal or greater accuracy than a human. While it may be true that speaker recognition systems have come a long way towards this goal, they are a lot more susceptible to the variations inherent in the human voice which humans find easier to accommodate. This limitation has proved to be the greatest stumbling block to their success (Rosenberg and Soong, 1992a).

The second reason for automating speaker recognition is more practical, as it involves the application of the technology to real-world situations (Doddington, 1985). Speech-based interfaces are seen as a viable proposition for the future, and speaker recognition (along with speech recognition) comprises an integral part of that interface (Furui, 1995). In particular, speaker recognition has a rôle to play in security applications where the identity of a user has to be confirmed with the utmost certainty.

The reason for choosing a person's voice as a means of identification is that it is considered to be a biometric identifier (Doddington, 1985). A biometric identifier is a characteristic that is supposed to be intrinsic and unique to a person and, as such, should not be reproducible by anyone else. Examples of biometric identifiers are fingerprints, retinal patterns and DNA. Biometric identifiers benefit from the fact that the person to be identified doesn't have to carry a card or a key that can be duplicated or stolen. Furthermore, a biometric identifier doesn't have to be remembered like a personal identity number (PIN) for an automatic teller machine (ATM) card.

Although a popular choice as a potential identifier, the extent to which a person's voice is a good biometric identifier is still open to question. DNA, fingerprints and retinal patterns are not susceptible to colds, laryngitis or emotional stress. However, speech by its very nature is variable and susceptible to many influencing factors. There are always differences between a speaker's repetitions of a word. This is particularly true if there are time lapses of a month or more between the repetitions (Furui, 1974; Furui, 1986). Furthermore, the prime purpose of speech is to convey a message, not the identity of the messenger (Rosenberg and Soong, 1992a). In addition to the speaker's message, speech also carries information about their identity, their language, their accent, as well as their emotional and physical state. As this information is secondary to the message being conveyed, it is difficult to determine it from the speech waveform. The message and the speaker's characteristics are non-linearly and inter-dependently encoded in the speech waveform (Furui, 1986). As yet, it is impossible to extract the characteristics that totally determine a person's voice from their speech waveform.

1.2 Speaker recognition

The speaker recognition task can be divided into two main categories: text-dependent recognition and text-independent recognition. These two tasks may, in turn, be either a case of verification or identification. The success of a speaker recognition system depends on which task is being undertaken (Atal, 1976; Doddington, 1985; O'Shaughnessy, 1987). (In this work, the term *utterance* refers to the recorded speech signal, rather than its content.)

In text-dependent (TD) speaker recognition, it is assumed that the speaker wishes to be recognised. The test utterance is predetermined and the speaker's model is dedicated to modelling the speaker for that utterance only. This definition comes from Rosenberg and Soong (1992a) where, in text-dependent tests, the content of the training and test utterances is the same. This is most often the case in security applications where a person may identify themselves, using their voice, to obtain security clearance or authorisation for a transaction. Some common examples of security applications are voice-activated locks, access to restricted computer data and voice verification for telephone-banking and ATM transactions.

Text-independent (TI) recognition differs from text-dependent speaker recognition in that there are no constraints on the speaker's vocabulary. The speaker is free to say whatever they like and in whatever manner they like. In this case the content of the training and test utterances is different (Rosenberg and Soong, 1992a). This means that the text-independent model must encompass the way the speaker says a variety of sounds, rather than concentrating on a particular word or words as in the text-dependent case. The possible applications for text-independent speaker recognition include forensic use, classification of intelligence data and passive security applications through voice monitoring (Doddington, 1985).

The term text-independent has also been used by Rosenberg and Soong (1987) and Yu, Mason and Oglesby (1995) for tests where the speaker model was trained with a set of digits and then tested using one of the digits. In this case the test word is one of the words used to train the model. This definition of text-independent does not apply to this work.

Since text-dependent speaker recognition models the speaker for predetermined phrases only, it has (in general) lower error rates than text-independent speaker recognition which must model the speaker's characteristics for a variety of speech sounds. However, in text-independent speaker recognition there are often longer utterances to analyse, and this increase in the information presented aids recognition (Doddington, 1985).

Text-dependent and text-independent speaker recognition can be divided into two categories: verification and identification. In speaker verification, the objective is to confirm a person's identity using their voice. This is the case when

someone uses a card or access code, that they alone should possess, and are asked to confirm their identity by using a special password. In this situation it is a true or false scenario because there are only two possible outcomes: either it is the claimed speaker or an impostor.

In closed-set speaker identification, the task is to identify the speaker as one of N possible speakers, in which case there are N possible outcomes. A possible identification task would be to match a criminal's voice with one from a list of suspects or to determine which actor from a cast is speaking. As the number of speakers increases, the likelihood of making a false identification also increases. For this reason, speaker identification for a large population is more difficult than verification. If there is also the possibility that the speaker comes from outside the group (known as open-set speaker identification), there are $(N + 1)$ possible outcomes, further increasing the chances of a false identification.

In this work, the speaker whose identity is being claimed in speaker verification will be referred to as the *genuine speaker*. Speakers other than the genuine speaker will be referred to as *impostors*. Since no presumption about identity is made in speaker identification, the genuine speaker is the speaker who actually said the utterance and every other speaker is an impostor.

As both text-dependent and text-independent speaker recognition are relevant to real-world applications, they are both studied in this work. Likewise, both speaker verification and speaker identification have different applications and so they too are considered throughout the thesis.

1.3 Thesis Outline

The thesis starts by investigating the application of artificial neural networks to the task of automatic speaker recognition. This led to the development of a novel means of ranking a speaker's impostors based on VQ *model vs. model* scores. These rankings were then used to select impostors for a score normalisation method which improved error rates for both VQ and PNN recognition systems.

The similar error rates of the PNN and VQ systems raised the question of whether further improvement in performance might be achieved through varying the feature set rather than the recogniser itself. Initial tests combined the scores from LP- and LP residual-based recognisers. However, a more successful approach

was found in sub-band processing. In a implementation previously untried for automatic speaker recognition, the speech signal was split into 16 channels, and each sub-band was modelled separately. The scores from the individual sub-bands were then combined to give a final score. This provided a more detailed speaker model, which was reflected in the improved performance of the sub-band processing system in comparison to the wide-band recognition system.

1.4 Thesis structure

The layout of the thesis is as follows. Chapter 2 covers the *in-house* and British Telecom (BT) Millar text-dependent speaker recognition databases and the text-independent TIMIT database. Chapter 3 reviews the speaker recognition methods that were used for the experiments, as well as the error measurements used to assess the performance of speaker recognition systems. Chapter 4 covers the experiments that were carried out on the in-house database using classifying neural networks. Chapter 5 describes the investigation of several methods of normalising scores for speaker verification. The results of the best normalisation, for both the text-dependent and text-independent databases, are presented in Chapter 6.

Chapter 7 returns to the use of artificial neural networks, by testing the text-dependent and text-independent databases with predictive neural networks. Chapter 8 investigates using a recognition system based on the LP residual in conjunction with an LPCC-based recogniser. Chapter 9 deals with sub-band processing, which uses several recognisers in parallel to model a speaker. Finally, Chapter 10 discusses the implications of the research and suggests further work that might be carried out.

1.5 Summary

This chapter has given a brief introduction to the field of automatic speaker recognition and the reasons why it is relevant to practical applications. It has also classified various types of speaker recognition situations and how they affect the success of the system. Finally it has presented an outline of the structure of the thesis.

Chapter 2

Speaker databases

2.1 Introduction

At the start of the work, a text-dependent database with multiple recording sessions and sufficient repetitions per session could not be found. For this reason it was decided to construct a small database, referred to as the *in-house* database, which was used for the initial artificial neural network experiments. However, approximately 15 months into the work a suitable text-dependent speaker recognition database was obtained from British Telecom (BT), which was used for the majority of text-dependent experiments. This database, called the Millar database, has also been used by other researchers in the field of speaker recognition (Yu *et al.*, 1995). The TIMIT database, which has been widely used in the area of text-independent speaker recognition (Artières and Gallinari, 1993; Besacier and Bonastre, 1997; Fredrickson and Tarassenko, 1995; Hattori, 1992), was obtained for the text-independent experiments. This chapter starts by describing how the speech was parametrised and then gives the details of each database.

2.2 Speech parametrisation

The most common feature sets for automatic speaker recognition are linear prediction cepstral coefficients (LPCC) and their temporal derivatives (Furui, 1997), and mel-frequency cepstral coefficients (MFCC) (Davis and Mermelstein, 1980). In a series of experiments comparing the robustness of various feature sets, Reynolds (1994) concluded that there was little to choose between the LPCC

and MFCC feature sets and that the channel compensation technique used is more important. For the majority of experiments carried out in this work, linear prediction cepstral coefficients were used to parametrise the utterances.

Unless otherwise specified, all utterances from all databases were parametrised in the same way, with 12th order linear prediction cepstral coefficients. An analysis frame of 20 ms, Hamming windowed and overlapping by 50%, was used. The 12th order linear predictor coefficients were generated using the autocorrelation method (Rabiner and Schafer, 1978). These were then used to create cepstral coefficients via the recursion described by Atal (1974), which is given in Appendix A.

2.3 Database considerations

There are several considerations to be taken into account when constructing a text-dependent speaker database. The choice of words used is crucial, as it will have a major influence on the results of any experiments (Hannah, Sapeluk, Damper and Roger, 1993a). To get reasonable results, words should be chosen that offer suitable variation between speakers. The longer a word is, and the more vowel sounds it contains, the better (Sambur, 1974). Sambur also found that nasals were a rich source of recognition information. For this reason a long word, "Allenwood", which has both vowel and nasal sounds, was chosen.

Another factor in designing and recording a database is to ensure that there are enough repetitions of each utterance. The database needs to contain utterances for both training and testing a speaker recognition system. Without enough utterances from the claimed speaker to test a system, the results will not be statistically significant.

Temporal variation is also crucial in speaker recognition. The human voice changes over time and it takes at least several recording sessions, spread over several months, to encompass all its variations (Furui, 1974; Furui, 1986).

Finally, it is desirable to select speakers who have similar accents and rhythms in their speaking manner, so that they will be quite likely to 'impersonate' each other, which should make the tests more rigorous.

2.4 In-house database

The in-house speaker database consisted of 20 male speakers with similar accents, saying "Allenwood" 20 times. The utterances were recorded over two sessions, approximately one month apart, in order to incorporate some temporal variation. The speech was recorded with a 16-bit A/D card at a sampling rate of 16 kHz, with a sixth-order low-pass filter to prevent aliasing. The recordings were made in a quiet laboratory. Each speaker's utterances were saved as integers in a binary file (`.idf`) (Hannah, 1997) with the name `spXuY.idf` where `X` stands for the speaker number and `Y` stands for the utterance number, e.g. `sp9u20.idf` would be the 20th utterance for speaker 9. Each utterance had its end-points located by hand. Finally, as MATLAB software would be used for all experiments, the files were converted from the `.idf` format to MATLAB format files. Each utterance was parametrised using LPCC as described in section 2.2.

2.5 BT Millar database

The BT Millar database is a text-dependent speaker recognition database. It consists of 46 male and 14 female native English speakers saying the words *one* to *nine*, *zero*, *nought* and *oh* 25 times each. The words were recorded in 5 sessions spaced out over 3 months. At each session the speaker was prompted visually to say the words in a random order.

The recordings were made in a quiet environment using a high-quality microphone. The speech was digitally recorded at a sampling rate of 20 kHz with a 16-bit A/D converter. As well as the 20 kHz recording, the database was also made available at an 8 kHz sampling rate. In this version, the speech had been bandpassed to telephone quality and then downsampled to 8 kHz. Only the 8 kHz version was used in the experiments.

Each of the recordings had a file header which gave the speaker's name, age and sex, as well as details about the time, quality and content of the recording. Rough start and end points for each utterance were also provided. However, these endpoints were neither consistent nor accurate enough for our purposes. In many cases the endpoints cut off the start or end of a word, and it was deemed better to have some silence at either end of a word than to remove part of the word. To correct this, an endpoint detection algorithm (based on Rabiner and Sambur

(1975)) was implemented and used on all 18,000 files in the database.

The BT Millar database was much better suited to the text-dependent tests than the in-house database. It had a range of 12 words, there were sufficient repetitions of each word and, most importantly, the utterances were recorded over several sessions.

For the majority of the work carried out, 31 male speakers of the same age-group and the words *one* to *nine* and *zero* were used for experiments. A restricted database used for some later tests is described in section 8.3. (The exact make-up of each speaker set may be found in Appendix B.) Each utterance was parametrised, after endpointing, using LPCC as described in section 2.2.

Typical training data for this database are the 10 repetitions from the first 2 recording sessions, with the 15 repetitions from the final 3 sessions being used for testing. Using test utterances from later recording sessions is important, as in a practical speaker recognition system there would be some time-lapse between training and testing the speaker model (Rosenberg and Soong, 1992a). This setup was used for nearly all experiments, with the exception of some temporal variation experiments where it was desirable to create a speaker model using utterances taken from all 5 recording sessions.

2.6 TIMIT database

The DARPA TIMIT database is one of the most widely used text-independent speaker-recognition databases. It consists of 630 speakers of American English each saying 10 sentences. The speakers were grouped into 8 regional dialects. The sentences consisted of 2 dialect sentences (SA), 5 phonetically compact sentences (SX) and 3 phonetically diverse sentences (SI). There were 450 phonetically compact and 1890 phonetically diverse sentences in total. The TIMIT sentences were recorded at 20 kHz, in a quiet recording booth using a high quality microphone. The recordings were later down-sampled to 16 kHz before storage. There was no need to endpoint the sentences as this had already been done. Each sentence was parametrised using LPCC as described in section 2.2.

A subset of 24 male and 14 female speakers from a single region was used for the experiments. (A list of the speakers used may be found in Appendix B.) In general, the SA and SX sentences are used for training and the SI sentences for

testing (Fredrickson and Tarassenko, 1995). However, for some experiments only the SX sentences were used for training, with both the SA and SI sentences used for testing (Artières and Gallinari, 1993). Using the SA sentences for testing is similar to a text-dependent test, as each speaker says the same phrase, whereas in conventional text-independent tests each speaker says something different.

2.7 Summary

The reasons for creating the in-house database and the method by which it was collected have been outlined. The details of the BT database and the text-independent TIMIT database were also given. The next chapter describes the speaker recognition methods used in the experiments and the error measurements used to evaluate their performance.

Chapter 3

Methods and evaluation

3.1 Introduction

Having explained the task of automating speaker recognition in Chapter 1 and described the speaker recognition databases and their parametrisation in Chapter 2, the means of modelling the speaker must now be dealt with. First of all, the importance of thresholds for speaker verification is mentioned, then the difference between the scores generated by classifying and distance models is explained. Then the methods by which these models were implemented for the experiments are given. Finally, the error rates used to measure recognition performance are explained.

3.2 Setting a threshold for verification

In conventional speaker verification, the test utterance is only presented to the genuine speaker model (Rosenberg and Soong, 1992a). To determine if it is a genuine speaker utterance or not, the score must be compared with a threshold. The setting of this threshold, so as to minimise the overall error rates, is one of the major problems in automatic speaker verification (Furui, 1981).

In closed-set speaker identification, the test utterance is presented to all models. The model with the best score determines the identity of the speaker. The score from any model is not taken in isolation, but compared to the scores from every other model, so there is no need to set a threshold.

A threshold may be set *a priori* (i.e. before the tests are run) or *a posteriori* (i.e. after the tests have been completed). If the threshold is set *a posteriori*,

when all the results have been collected, it may be optimised to reduce the overall error rate. This would not be possible, however, in a practical speaker recognition system, where the threshold would have to be set *a priori* because the test results are unavailable. The problem with setting the threshold *a priori* is that it has to be based on the training data, and this may not be a reliable indicator of the test data (Rosenberg, DeLong, Lee, Juang and Soong, 1992b).

3.3 Classifying and distance models

There are two main types of model used in speaker recognition: the classifying model and the distance model. In this work, the terms classifying model and distance model refer to the data used to create the model, the training method and the score generated by the model.

A classifying model refers to a model that is trained to discriminate between genuine speaker and impostor utterances. It uses genuine speaker and impostor information in the creation of the model, and calculates the likelihood of the test utterance belonging to the speaker model.

A distance model refers to a model that is trained using only genuine speaker information, and generates a score that is a geometric distance between the test utterance and the model. This means that the classifying model tries to incorporate information about both intra- and inter-speaker variation, whereas the distance model is concerned only with modelling intra-speaker variation.

The term *intra-speaker variation* refers to the variation in a speaker's way of saying an utterance from one instant to the next. Doddington (1985) gives an example of 5 spectrograms from the same speaker saying the same word which show considerable variation from utterance to utterance. Rosenberg and Soong (1992a) found that utterances from a single recording session are more highly correlated than those from separate recording sessions. So although a speaker never says a word the same way twice, utterances from a single recording session are more similar to each other than utterances from other recording sessions. Furui (1974) found that it took recordings over 3 months to encompass the long-term variations in a speaker's voice.

Inter-speaker variation for a particular word refers to the variation between speakers in how they say that word. In speaker recognition, it is desirable to

emphasise the variation between the speaker to be recognised and the other speakers. As an example of inter-speaker variation, Doddington (1985) has examples of identical twins who sound very similar to each other, but whose spectrograms show major differences from each other.

The classifying model returns a score which is the probability of the test utterance belonging to the speaker modelled. A distance model returns a score which is a measure of the geometric distance (c.f. section 3.5) between the test utterance and the speaker model. This score must be either compared with other model scores (for identification) or with a threshold (for verification) before the distance can be quantified as being close to or far from the model.

For a basic distance model, choosing the model to which an utterance is closest using the Mahalanobis distance (Mahalanobis, 1936) is tantamount to choosing the maximum likelihood class (Deller, Proakis and Hansen, 1993). The Mahalanobis distance is based on the inverse of the covariance matrix for the individual cepstral coefficients for each speaker. A simpler approach is that of Tohkura (1986), which uses the inverse variance of each coefficient rather than the covariance matrix. However, both of these schemes require data other than that used to train the original model.

In this work, the classifying model was implemented using classifying neural networks, and the distance models using predictive neural networks and vector quantisation.

3.4 Artificial neural networks

Two different neural network architectures were applied to the text-dependent speaker recognition problem: the back-propagation (BP) neural network and the radial basis function (RBF) neural network. Both networks are described in detail in Haykin (1994). A detailed account of the rôle of artificial neural networks in speaker recognition is given in Bennani and Gallinari (1994).

3.4.1 Back-propagation artificial neural network

The multi-layer perceptron architecture using the back-propagation learning algorithm (Rumelhart, Hinton and Williams, 1986) is one of the most widely-used neural networks. It commonly consists of three layers of processing units:

an input layer, a hidden layer and an output layer. The hidden and output layers have a non-linear *tanh* or *sigmoid* activation function. The back-propagation learning algorithm is a supervised learning algorithm that uses two passes through the network to calculate the change in network weights. In the forward pass, the weights are fixed and the input vector is propagated through the network to produce an output. An output error is calculated from the difference between actual output and the desired output. This is then propagated backwards through the network making changes to the weights as required.

3.4.2 Radial basis function artificial neural network

The radial basis function neural network (Moody and Darken, 1989) has both a supervised and unsupervised learning component. As with the more common back-propagation network, it is a three-layer network. The hidden layer of the RBF network is a series of 'centres' in the input data space. Each of these centres has an activation function, typically a Gaussian function. The activation of the centre depends on the distance between the presented input vector and the centre. The further a vector is from a centre the lower the activation of the centre and vice versa. The generation of the centres and their widths is generally done using an unsupervised k-means clustering algorithm. The centres and widths created by this algorithm then form the weights and biases of the hidden layer, which remain unchanged once the clustering has been done. The output layer is trained using the back-propagation learning rule.

3.4.3 Predictive and classifying neural networks

Both BP and RBF networks may be used for classifying and predicting. In the case of a classifier, the network is presented with examples from the classes to be identified. In speaker recognition there are two classes – the genuine speaker and an impostor. The network is trained to classify the input patterns into either genuine or impostor classes. A typical ideal output would be $[+1\ -1]$ for the genuine speaker and $[-1\ +1]$ for an impostor. Naturally the values would vary between these extremes as some test patterns would be of uncertain origin. The most important aspect of classifying networks is the training data. If the training data are not representative of the test data set, then the network may generalise

badly, and thus misclassify the test data.

In a predictive neural network, the network is trained to predict samples from a series based on previous samples of the series. In learning to predict future samples of the system from previous samples, the network attempts to model the underlying function of the series. In the case of speaker recognition, the network models the particular speaking manner of the speaker (Hattori, 1992). Then, when the network is presented with samples from the genuine speaker's utterance, it should be able to predict the next samples with high accuracy. The difference between the samples predicted and the actual samples is a measure of how good the network model is. If an impostor's utterance is presented to the network it should not be able to predict the following samples with the same accuracy, as it has less information about the underlying model of speaking. So a predictive network provides a score at the end of an utterance that is a measure of how closely the network predicted the speech presented to it. A low score would indicate the genuine speaker and a high score would indicate an impostor.

3.5 Vector quantisation

Vector quantisation (VQ) has been used extensively for both text-dependent and text-independent speaker recognition (Booth, Barlow and Watson, 1993; Rosenberg and Soong, 1987; Yu *et al.*, 1995). It is a data-reduction technique (Picone, 1993) in which similar vectors are grouped together and represented by their centroid. The grouping is repeated iteratively until the distance between each vector and its group centroid has been minimised. These centroids make up the codebook which models the data, and their number will determine the accuracy of the modelling. The standard LBG algorithm (Linde, Buzo and Gray, 1980) was applied to calculate the centres.

The distance, d_{jk}, between vectors j and k (codebook centres or test utterance frames) was calculated using the 'city-block' measure as:

$$d_{jk} = \frac{1}{M} \sum_{i=1}^{M} |c_{ij} - c_{ik}|$$

where M is the order of the predictor, c_{ij} is the ith cepstral coefficient of vector j, similarly c_{ik} for vector k. (As usual, c_0 was discarded.) When testing an utterance against a VQ speaker model, the minimum absolute distance between

each utterance vector and the codebook for each analysis frame was summed over the total number of frames of the utterance. This sum was then averaged over the number of frames to give the final score.

3.6 Genetic algorithms

A genetic algorithm is an optimisation technique rather than a pattern recognition technique like vector quantisation or classifying artificial neural networks. However, genetic algorithms may be used to generate weights for the cepstral vector elements used in the distance calculation, with the objective of lowering the error rates. Therefore, a brief account of how they work is now given. A detailed description may be found in Goldberg (1989).

Genetic algorithms are search algorithms based on the mechanics of natural selection and genetic mixing. For any given problem, a genetic algorithm starts with a population of possible solutions. These solutions are coded as binary strings. The success of each solution is determined, and the best solutions are 'reproduced' at the expense of the poorer solutions. That is, these better solutions are mixed with each other through swapping segments of the binary strings (known as *crossover*), and varied slightly through flipping the values of randomly selected bits (known as *mutation*). This creates a new population of strings which should, on average, give better results than the original population. This process is repeated for a fixed number of iterations (known as generations), with the expectation that the best solution of the final population will be a significant improvement on the best solution of the original population.

The advantage of genetic algorithms over other search algorithms is that the method uses randomness to avoid getting caught in local minima. However, this means that the algorithm must be run several times to increase the likelihood of finding one of the GA's better solutions. The search is directed by the reproduction of good solutions and the removal of poor solutions. A genetic algorithm does not guarantee an optimal solution, though the results may be more or less good depending on the problem in question.

In terms of speaker recognition, genetic algorithms have been used to weight the feature set so as to emphasise elements which show strong speaker-dependence (Charvet and Jouvet, 1997; Hannah, Sapeluk, Damper and Roger, 1993b). In

this work, genetic algorithms are used to emphasise the sub-bands which show the most speaker-dependence in the sub-band processing system of Chapter 9. In both cases the cost function is some measure of the speaker recognition error rate, which the GA tries to minimise.

3.7 Performance evaluation

When a system has been constructed and the tests run, the success of the system must be evaluated. The following sections cover the most common methods of assessing performance, as well as a less common one that will be used throughout the thesis.

It should be pointed out that these measurements alone cannot make or break a speaker recognition system. Many other factors must be taken into consideration, such as the size of the database, the utterance content, the quantity of training data available, the duration of the test utterances and whether there is temporal or channel variation between the training and test utterances. Ideas have been put forward as how best to unify these points in evaluating a speaker recognition system by Oglesby (1995).

3.7.1 Identification error rate

This is the most straightforward of the assessments. The test utterance is presented to all speaker models and the model with the best score determines the identity of the speaker. The identification error rate (IE) is then the percentage of false utterance attributions for the test set:

$$IE(\%) = \frac{f}{N} \times 100$$

where f is the number of false attributions and N is the total number of test utterances.

3.7.2 d' separation measure

d' is a measure of the separation between the distribution of genuine speaker and impostor scores. It is based on classical signal detection theory (Green and Swets, 1966), where the separation between two response distributions (signal

and signal-plus-noise, which correspond to genuine speaker and impostor score distributions respectively) of equal variance is taken to be:

$$d' = \frac{|\mu_s - \mu_{sn}|}{\sigma^2}$$

where μ_s and μ_{sn} are the means the signal and signal-plus-noise distributions respectively, and σ^2 is the common variance.

The assumption that the distributions share a common variance is unlikely to be satisfied in speaker recognition, so the formula was modified to take account of this:

$$d' = \frac{\mu_{imp} - \mu_{gen}}{S^2}$$

where μ_{imp} and μ_{gen} are the means of the impostor and genuine speaker distributions respectively, and S is the geometric mean of the standard deviations of the two distributions:

$$S = \sqrt{\sigma_{imp}\sigma_{gen}}$$

where σ_{imp} and σ_{gen} are the standard deviations of the impostor and genuine speaker distributions respectively. A diagram to clarify the measure is given in Figure 3.1. It shows a histogram of genuine speaker and impostor scores.

A high d' represents good separation between the impostor distribution and the true speaker distribution, with a d' of approximately six representing virtually complete separation (Hannah, 1997).

3.7.3 Average equal error rate

Commonly used terms when assessing system performance are *false rejection* and *false acceptance* error rates. The false rejection error rate (FR) is the percentage of genuine speaker utterances that were rejected as belonging to an impostor:

$$FR(\%) = \frac{f_r}{G} \times 100 \tag{3.1}$$

where f_r is the number of genuine speaker utterances that were rejected as belonging to impostors and G is the total of genuine speaker test utterances.

The false acceptance error rate (FA) is the percentage of impostor utterances that were accepted as belonging to the genuine speaker:

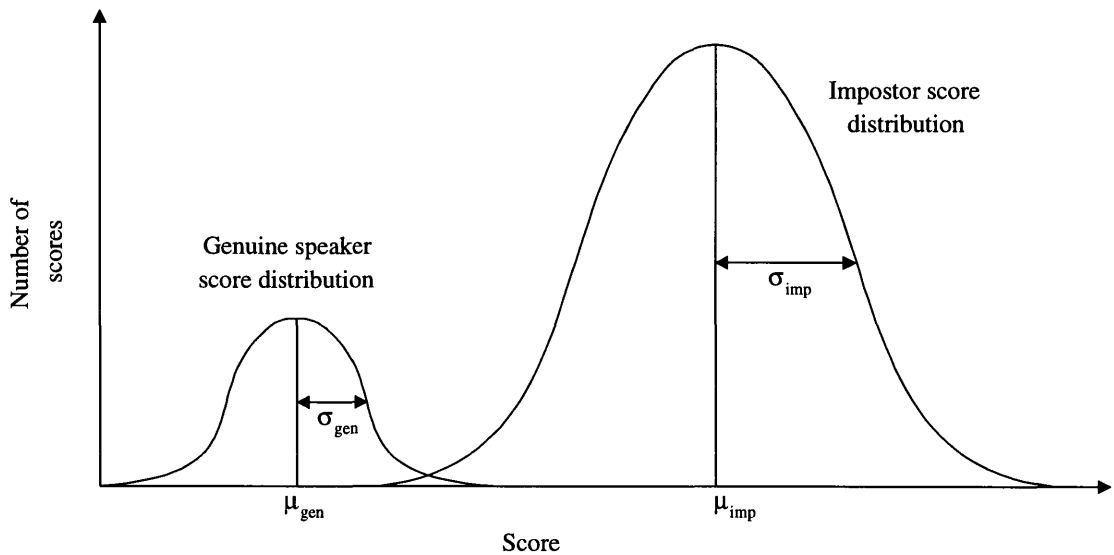

Figure 3.1: An example of genuine speaker and impostor score distributions.

$$FA(\%) = \frac{f_a}{I} \times 100 \tag{3.2}$$

where f_a is the number of impostor speaker utterances that were accepted as belonging to the genuine speaker and I is the total of impostor speaker test utterances.

The equal error rate (EER) occurs when the false rejection rate (Equation 3.1) equals the false acceptance rate (Equation 3.2):

$$EER(\%) = FR = FA$$

The equal error rate is found by varying a threshold. Depending on whether a good score is high or low, utterance scores on one side of a threshold are accepted as being genuine speaker utterances and scores on the other side are rejected as being impostor utterances.

In this work, the threshold was varied to minimise the difference between the false rejection and the false acceptance error rates. As there are rarely as many genuine speaker test utterances as there are impostor test utterances, quantisation error meant that the false rejection and false acceptance error rates were not always equal, though they usually varied by very little. To account for this, the equal error rate was calculated in terms of the total number of false rejections and false acceptances for all tests:

$$EER(\%) = \frac{f_r + f_a}{G + I} \times 100$$

The average equal error rate is one of the most common ways of presenting speaker verification results. It is the average of the equal error rates for all the speakers being tested:

$$Average\ EER(\%) = \frac{1}{M} \sum_{i=1}^{M} EER_i$$

where M is the number of speakers in the database and EER_i is the equal error rate for Speaker i.

3.8 Summary

This chapter has provided an overview of the methods and their evaluation that were used for the experiments in the following chapters. The differences between classifying and distance models, and the means of evaluating a speaker recognition system have been given. The following chapter starts the review of experimental results with the text-dependent experiments using classifying artificial neural networks.

CHAPTER 4

CLASSIFYING NEURAL NETWORKS

4.1 Introduction

The experiments related in this chapter concern the use of classifying neural networks. These networks are trained discriminatively using genuine speaker and impostor data. For a given test utterance they return a score which is a measure of the likelihood of it belonging to the genuine speaker. This is in contrast to speaker models which give the distance of the utterance from the model (c.f. section 3.3). Two neural networks were investigated: the multilayer perceptron (MLP) and radial basis functions (RBF) neural networks. (The multilayer perceptron was trained using the back-propagation (BP) learning algorithm and the two terms will be used interchangeably.) These have been the most popular networks for speaker recognition and have produced the best results (Bennani and Gallinari, 1994).

The back-propagation and radial basis function neural network experiments were carried out on the in-house database (c.f. section 2.4), as it was the only one available at the time. To reiterate, this database is text-dependent and the recordings were made in two sessions a month apart. Each speaker said the word "Allenwood" 10 times in each session.

4.2 Experiments

The experiments set out to investigate several aspects of applying classifying neural networks to speaker recognition. The first of these was to vary the make-up of the training set, in order to determine how important it is to the success

of the system. It was also desired to determine whether back-propagation or RBF networks were more suited to the speaker recognition problem, as work by Mak, Allen and Sexton (1994) and Oglesby and Mason (1991) suggests that RBF networks are better than MLP networks. This was tested by comparing the results of the two networks for the same test data.

Later experiments examined the relationship between the genuine speaker and their best impostors. If there was a clear relationship, then it might be possible to create training sets that emphasise the differences between the genuine speaker and their closest impostors. This was further investigated by attempting to predict these best impostors using vector quantisation models, so that the impostors could be identified before the neural networks had been trained. Finally, the effect of removing some of the speakers from the training set, but including them in the test set, was investigated. This simulated a practical speaker recognition task where not all impostors would be available for inclusion in the training set, which is an important engineering consideration.

The general setup for each experiment was similar. Each speaker had their own MLP and RBF network. Each network had 2 output nodes, one indicating the likelihood that the input vector belongs to the genuine speaker and the other the likelihood that it belongs to an impostor – although only the first of these was actually used in testing. Target values during training were $[+1\ -1]$ for a true speaker frame and $[-1\ +1]$ for an impostor frame.

Speech was presented to the networks as a sequence of 4 cepstral vectors, each of length 12. The presentation of 4 cepstral vectors at each instant allowed the networks to incorporate some short-term temporal speech information as well as static information.

The number of training patterns, N_T, used to train each network was typically 1250 (depending upon utterance length). In line with usual practice, the number of hidden nodes, learning rate, momentum etc. were set empirically. These variables were optimised to reduce the network error on the training data, as cross-validation test data were not used. In the case of the MLP, there was a single hidden layer of 64 nodes and a *tanh* activation function was used. The RBF network used $0.25 \times N_T$ hidden nodes.

The score for a test utterance for a given network was obtained as follows. Each set of 4 frames from the utterance was presented to the network and the

resulting score at the output, i.e. the first of the two output nodes mentioned above, was recorded. The average output value across all frames of the utterance was then computed, and taken to be the score. No use was made in this study of any measure of dispersion, such as the standard deviation, of the scores. For the verification tests, a fixed threshold of 0 was taken. Any value above 0 was deemed to be the true speaker and any below an impostor. The identification test was done by comparing the outputs of all networks for a particular utterance. The network with the highest output was judged to be the true speaker. No minimum difference between network outputs was required.

4.2.1 Varying the training set

The first experiment varied the content of the training sets to study how it affected the performance of the networks. If system performance is highly dependent on the make-up of the training set, then it may be worthwhile supervising the make-up of the training set (using knowledge about problematic speakers) rather than using random selection.

Ten training sets were created for each speaker, totalling 200 training sets. A separate network was created for each training set. Each training set consisted of 6 true speaker utterances and 19 false speaker utterances (one from each of the possible impostors). These utterances were chosen randomly for each training set. For the verification tests, this meant that there were 14 true speaker test utterances for each network, and 2800 true speaker test utterances for the 200 networks in total. There were 361 impostor test utterances per network and, hence, $361 \times 200 = 72200$ impostor test utterances in total. For the identification tests, there were 14 true speaker test utterances per network and, hence, 2800 true speaker tests in total. These training sets were applied to both the MLP and RBF networks.

4.2.1.1 Results

The results for varying the training set are presented in the graphs in Figure 4.1, which show the total number of failures per training set, the d' values for each training set and the number of false rejections and false acceptances per training set. In each case, graph (a) represents the results for the back-propagation networks and graph (b) the results for the RBF networks. Each training set

is referenced by the index of the x-axis. Every decade represents the results of a single speaker's training sets.

In Figure 4.1(i) the total area under graph (a) is clearly less than that of graph (b), indicating that the total number of errors for the RBF network is less than that of the back-propagation network. The RBF network had in fact only 219 failures out of a possible 75000 compared to 426 for the back-propagation network. The graphs also show that, if the RBF network had trouble separating the speakers, then the results for the back-propagation network were in general markedly worse.

Figure 4.1(ii) shows the d' values of the true speaker distribution against the impostor distribution for both networks. The average value for the RBF network is 7.9 compared to 6.5 for back-propagation. So the RBF system created a greater gap between the true and false speaker distributions. There is also a high correlation coefficient of 0.82 between the d' values for the RBF and back-propagation networks. This indicates that the success of both networks is highly dependent on training sets and that, in general, a good training set for the RBF network is a good training set for the BP network and vice versa.

Figure 4.1(iii) shows that the area under the curve for the back-propagation network is only slightly larger than that for the RBF network, indicating a similar level of false rejections. The back-propagation network had 162 false rejections while the RBF system had 142. So the RBF network did not succeed in making significant differences to the number of true speakers who were rejected. However, Figure 4.1(iv) shows that the RBF network significantly reduced the number of false acceptances, with only 77 compared to 264 for the back-propagation network.

The effect of training sets on the success of the classifying neural networks is summarised in Tables 4.1 and 4.2, where 'Total' refers to the total number of false rejections plus false acceptances. For the randomly-selected training sets, a single set of results was randomly chosen for each speaker and the overall results calculated. This was done 100 times and the average calculated. For the optimal results, the best-performing training set for each speaker was chosen and the overall results calculated. There is a significant difference for both networks between the randomly-selected and best-performing training sets. In all error measurements improvements are found when the best-performing training sets are used.

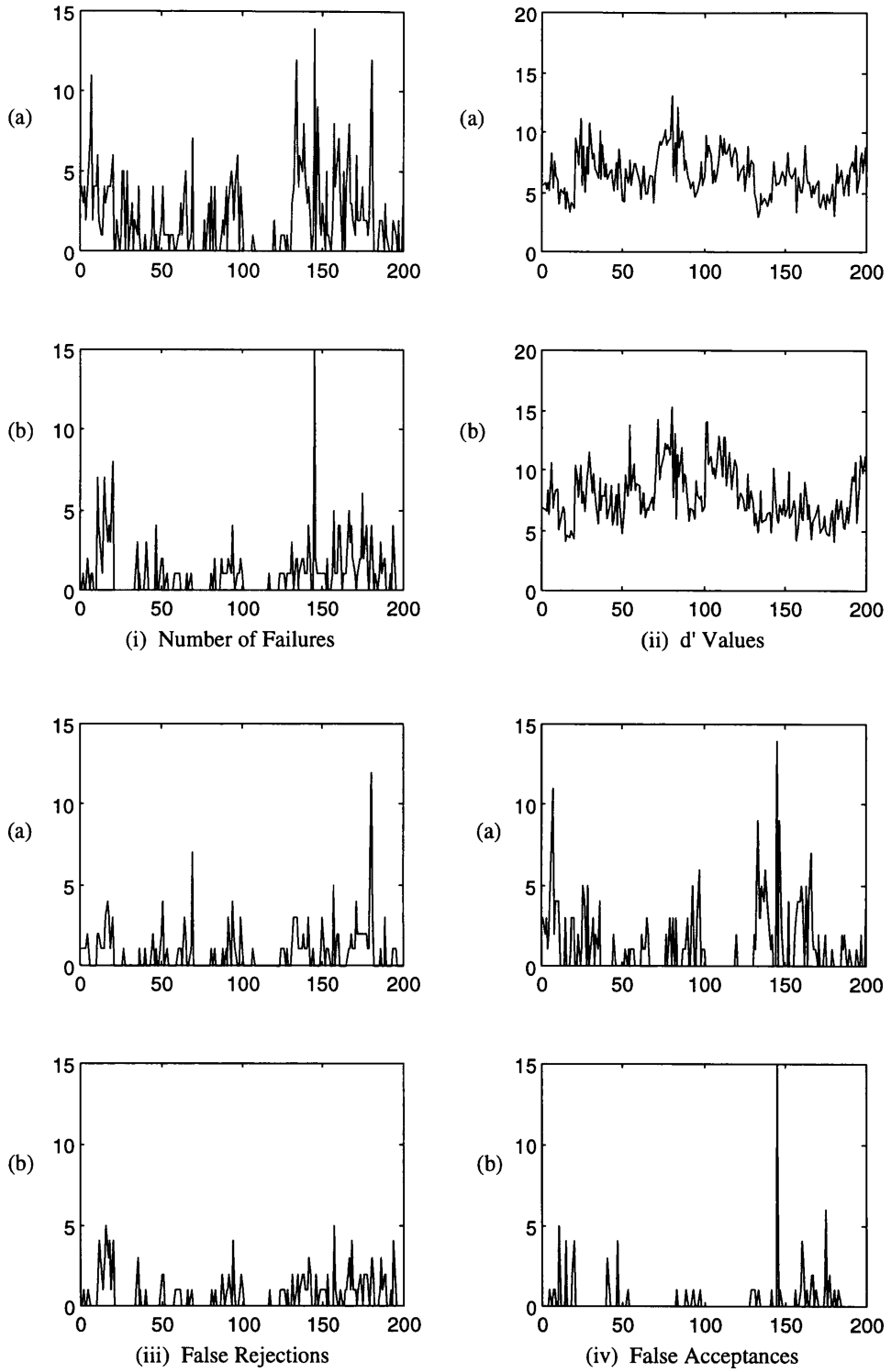

Figure 4.1: Comparison of results for (a) the BP and (b) the RBF networks for the 200 training sets.

4.2.1.2 Discussion

The difference between the results for the randomly-selected and best-performing training sets highlighted how important the make-up of the training set is to the success of the classifying neural network recogniser. An example of this is shown in Figures 4.2 and 4.3, where each utterance score is represented by a '$*$'. Figure 4.2 shows Speaker 15's results for a good training set. There is good separation between the impostor and genuine speaker score distributions (a d' of 10.2), and a threshold of zero gives an equal error rate of 0%. Figure 4.3 on the other hand, shows Speaker 15's results for a poor training set. In this case there is virtually no distance between the impostor and genuine speaker distributions and a threshold of zero would produce 15 false acceptances. Even if the false acceptances were reduced by raising the threshold, the absolute difference between the distributions is smaller than for the good training set (a d' of 5.9 compared to 10.2).

As the make-up of the training set is so important, section 4.2.3 looks at trying to determine speakers who consistently impersonate each other, so that they might be grouped together. Section 4.2.4 then looks at trying to determine the closest impostors for a given speaker before the network is trained and tested.

4.2.2 BP vs. RBF performance

Although back-propagation is the most commonly used artificial neural network, radial basis function networks have a better reputation for speaker recognition (Bennani and Gallinari, 1994). To investigate if this was warranted required a comparison of the results from the previous experiments as both networks were trained with the same data sets.

4.2.2.1 Results

The analysis of the graphs in Figure 4.1 (c.f. section 4.2.1.1) showed that the RBF network outperforms the BP on all counts. This is summarised by the results in Tables 4.1 and 4.2. In each case, the RBF has lower error rates than the BP network for both randomly-selected and best-performing training sets.

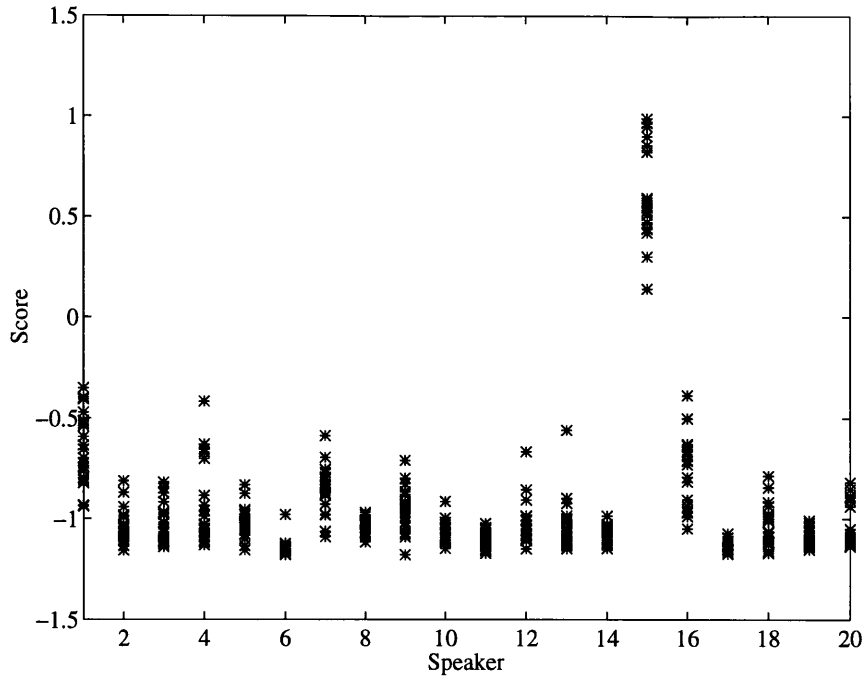

Figure 4.2: Example of a good training set for Speaker 15 where all the impostor scores are below the zero threshold.

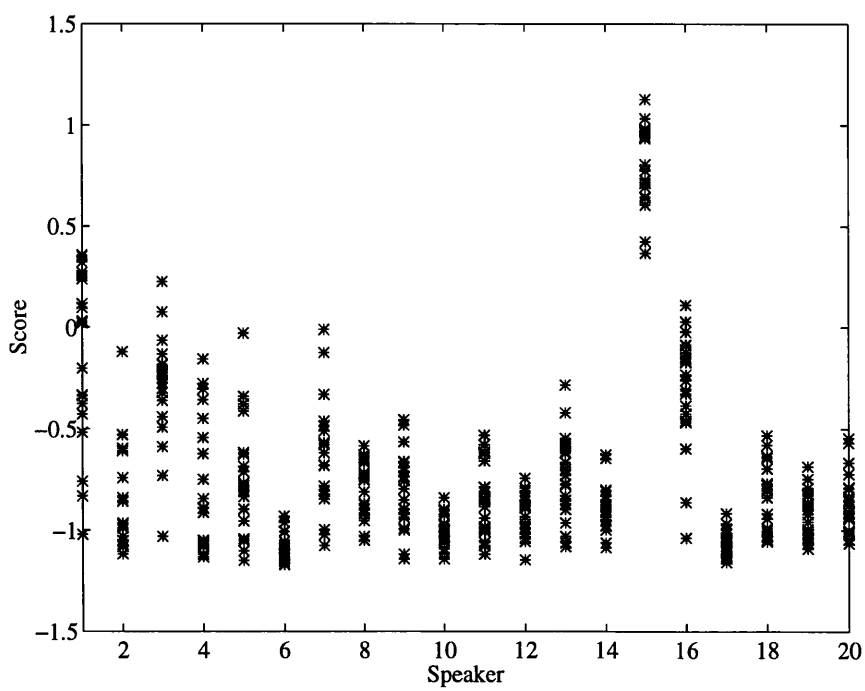

Figure 4.3: Example of a poor training set for Speaker 15 where several impostor scores are above the zero threshold.

	d'	IE(%)	FR(%)	FA(%)	Total(%)
Randomly-selected	6.5	2.0	5.8	0.4	0.6
Best-performing	8.0	0.0	1.4	0.1	0.1

Table 4.1: Results for the back-propagation neural network (IE = identification error, FR = false rejection, FA = false acceptance, Total = percentage of combined false rejection and acceptance errors).

	d'	IE(%)	FR(%)	FA(%)	Total (%)
Randomly-selected	7.9	1.1	5.2	0.1	0.3
Best-performing	9.8	0.0	0.7	0.0	0.0

Table 4.2: Results for the radial basis function neural network (IE = identification error, FR = false rejection, FA = false acceptance, Total = percentage of combined false rejection and acceptance errors).

4.2.2.2 Discussion

The RBF network outperformed the BP on all accounts for both randomly-selected and best-performing training sets. The superiority of the RBF network over the BP network agrees well with results reported by other researchers (Bennani and Gallinari, 1994; Oglesby and Mason, 1991; Mak *et al.*, 1994). In a comparison of the two networks using a sequence of 5 digits, Mak *et al.* (1994) got identification recognition rates of 90.2% for the MLP networks and 98.7% for the RBF networks. For a fixed 1% false acceptance rate, Oglesby and Mason (1991) got recognition rates of 17% for the MLP networks and 8% for the RBF networks.

The benefit of the RBF network seemed to be that it defined the impostors more clearly, thus reducing the number of false acceptances. While the RBF also had fewer false rejections than the BP, the difference was not quite as pronounced. However, while the RBF could improve on the BP results, there were instances where neither the RBF nor the BP networks could completely separate the genuine and impostor speakers.

4.2.3 Speaker similarity

On inspection of the results of the previous experiments, it was observed that certain groups of speakers seemed to score consistently well against each other's models. The inter-speaker difference between these speakers was smaller than for the rest of the population. This meant that, given Speaker A and Speaker B, Speaker A's utterances would score well against Speaker B's model, while Speaker B's utterances would score well against Speaker A's model. This would seem to indicate that the cepstral representations of these speakers must appear to the neural network to be quite similar. Quite often, the speakers' utterances would be so similar that they became the cause of many of the false rejections or acceptances. If this relationship could be confirmed, then it might be possible to group speakers together, which may lead to better training sets that allow the network to focus on differences between the genuine speaker and the speakers most likely to impersonate them.

An example of this is shown in Figures 4.4 and 4.5, where each utterance score is represented by a '$*$'. They show the results for Speakers 2 and 10 respectively. It is clear from the results that Speakers 2 and 10 are similar to each other, as they both score well against each other's model. It is also evident that they both share something in common with Speakers 6 and 14, who score well against both models.

Although this phenomenon was clearly visible by inspection of the results, it was necessary to quantify the similarity in performances. To do this requires impostor ranking. An impostor's ranking is determined by how well their utterances score against the genuine speaker model. An impostor whose mean utterance score is highest against the genuine speaker model will be ranked first and those who score lowest ranked last. Given these rankings for every speaker, it is possible to compare ranking positions for groups of speakers.

To determine the rankings the average score per utterance per training set was used. So for Speaker A against Speaker B's model:

$$S_{av}^{AB} = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{N}\sum_{j=1}^{N}s_{ij}^{AB}$$

where S_{av}^{AB} is the average score for Speaker A's utterances against Speaker B's model, M is the total number of training sets (ten in this case), N is the number of test utterances and s_{ij}^{AB} is the score for Speaker A's j^{th} utterance in training

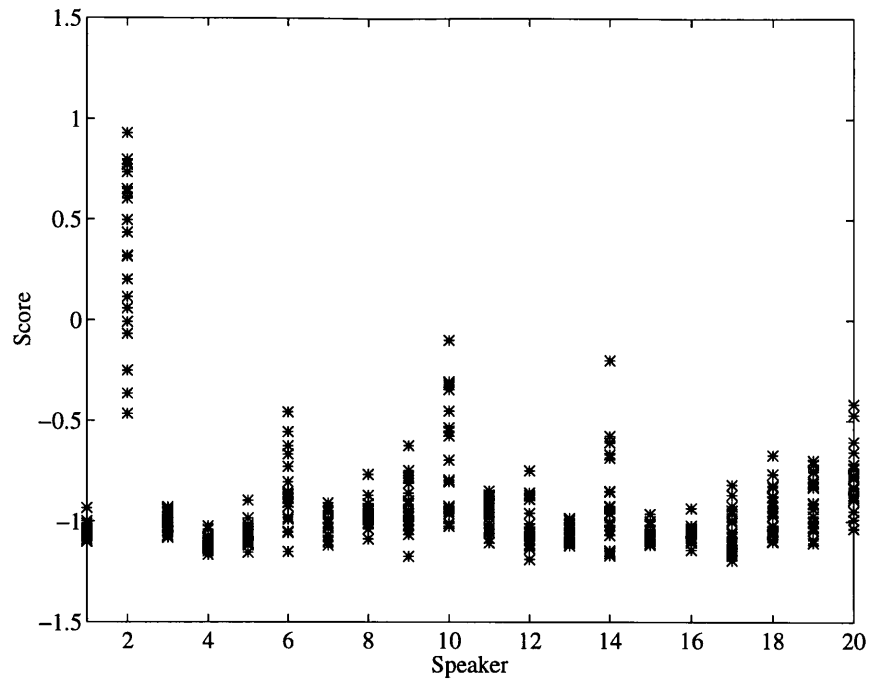

Figure 4.4: Results for Speaker 2.

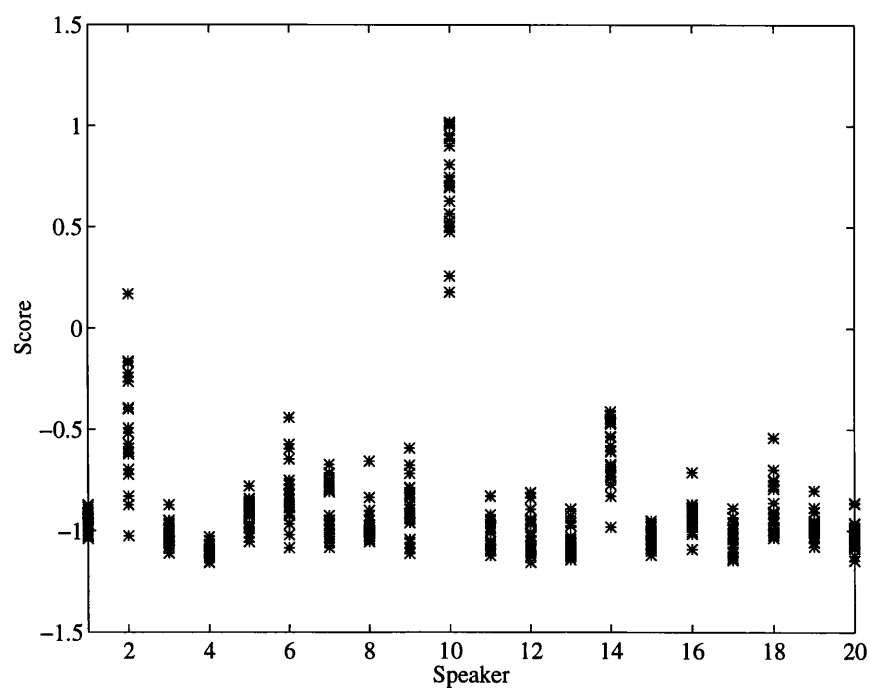

Figure 4.5: Results for Speaker 10.

set i against Speaker B's model. Using this method, a list of ranked impostors may be drawn up for each speaker.

Assuming that Speaker A's ranked impostor list is $\{B, C, D, E\}$, and that there is perfect agreement between the impostor rankings, Speaker A would be the best impostor of Speaker B, the second best impostor for Speaker C, the third best impostor for Speaker D and the fourth best impostor for Speaker E. The absolute difference between Speaker A's actual ranking and their theoretical ranking is used to measure the degree of similarity. So, if Speaker A was actually ranked second, rather than fourth, against Speaker E then that would be a ranking difference of 2 (the difference between the ideal and the actual rankings).

Using this ranking difference measure, the similarity between the speaker rankings for each position in the list may be calculated. This was done for both the BP and RBF networks.

An equivalent process was also performed for a VQ system, which used 6 words to create a 64-element codebook. This was done to test if this grouping of speakers was only a feature of the neural networks or if it also applied to VQ. The average score for each speaker's utterances against each speaker model was used to determine the ranking of the impostors.

4.2.3.1 Results

The results of the speaker similarity tests are shown in Figures 4.6 and 4.7. The x-axis represents the position in the impostor list, with 1 being the best impostor and 19 being the worst. The y-axis represents the average difference between the ranking according to a genuine speaker's impostor list and the actual ranking according to the results of the tests. An average ranking difference of 2 would mean that, on average, speakers were 2 places away from their expected position in the impostor rankings. The smaller the difference, the better the correlation between a speaker's impostor rankings and their position in other speaker's impostor rankings.

In Figure 4.6, the results for the neural networks show that the best correlation between speakers is for the closest impostors. In the back-propagation case, where the average ranking difference for the closest impostor is 1, if Speaker A is the best impostor for Speaker B, then Speaker B will be, on average, one of the best 2 impostors for Speaker A. In the RBF case, where the average ranking difference

for the closest impostor is 1.9, if Speaker A is the best impostor for Speaker B, then Speaker B will be, on average, one of the best 3 impostors for Speaker A. In general, the BP network has a better correlation between speaker positions than the RBF network.

This close correlation between best impostors is not found in the VQ results, which are shown in Figure 4.7. For VQ, if Speaker A is a good impersonator of Speaker B, then Speaker B will only be in the top 6 impostors for Speaker A. Nor is there any evidence that the best impostor is any more likely to be well correlated than any of the other impostor positions.

4.2.3.2 Discussion

The results of the speaker similarity tests showed that for the artificial neural networks at least, if Speaker A was a good impersonator of Speaker B, then the opposite was also likely to be true. This similarity between speakers indicated that it might be possible to group speakers as being similar using classifying artificial neural networks. However, this similarity between speakers was not found with the distance models of the VQ system, thus indicating that the similarity is probably a function of the networks' weighting of the cepstral vectors.

4.2.4 *A priori* impostor ranking

The previous experiment seemed to confirm that certain impostors appear to be particularly good at impersonating certain speakers. However, this could only be determined *a posteriori*, once the experiments had been completed. It would be useful to be able to predict before training an artificial neural network system (i.e. *a priori*) which speakers were most likely to impersonate each other. Then it might be possible to incorporate more utterances from problem speakers in the training set and thus reduce the error rates. There is no direct way of comparing the similarity of speakers' neural network models. However, using VQ speaker models, it may be possible to detect similarities between speakers.

A VQ model of each speaker was generated using all 20 utterances to create a 128-element codebook. The codebook for each speaker was tested against the codebook for every other speaker. Using these scores, it was possible to rank the speakers as impostors for a particular genuine speaker. (The ranking of impostors based on VQ models is described in detail in section 5.3.)

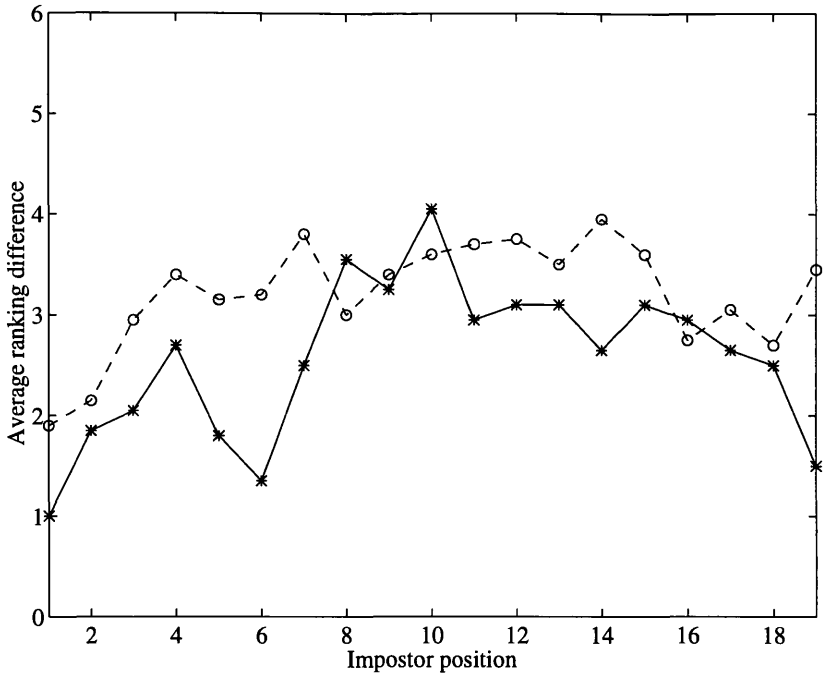

Figure 4.6: The average difference in ranking for the artificial neural network impostor positions ('$*$' = back-propagation network, 'o' = RBF network).

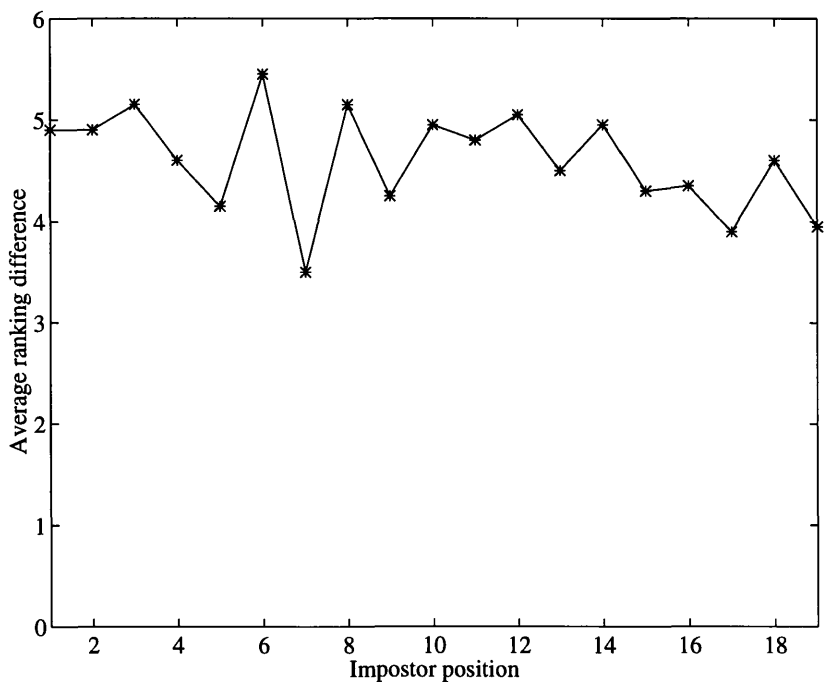

Figure 4.7: The average difference in ranking for the vector quantiser impostor positions.

	Average Ranking Difference
BP,RBF	2.1
BP,VQM	4.5
RBF,VQM	4.4
VQ,VQM	1.6

Table 4.3: Average difference in impostor positions for BP, RBF, VQ and VQ *model vs. model* (VQM) rankings.

The rankings for the BP and RBF neural networks were calculated from the average utterance score for each speaker against every other speaker, as described in the previous section (4.2.3). Impostor rankings for the VQ system were calculated in the same way (i.e. using impostor scores rather than *model vs. model* scores). This meant that there were 4 sets of impostor rankings for each speaker, 2 based on the BP and RBF networks, 1 based on the actual VQ results and the final one based on the VQ *model vs. model* (VQM) tests. The average difference in position between these rankings was calculated in the manner described above in section (4.2.3).

4.2.4.1 Results

The average difference in rankings between the networks and the VQ system are shown in Table 4.3. The RBF and BP rankings are correct to within approximately 2 places of each other. The BP and RBF rankings are 4.5 placings away from the VQ model rankings. Finally, the VQ test and the VQ model rankings are 1.6 places away from each other on average.

In effect this means that the rankings from the networks are quite similar to each other, but have little in common with the VQ model rankings. The VQ tests and the VQ model rankings are very close to each other, being less than 2 places away on average. This means that the VQ model rankings would not be particularly good at predicting the best impersonators for the neural network speaker models. They would, however, appear to be good at predicting the best impersonators for the VQ system. This aspect of the results is examined in more detail in the next chapter (section 5.3).

4.2.4.2 Discussion

The results of the impostor ranking showed that for an artificial neural network classifier it would be difficult to predetermine the speakers most likely to impersonate each other before the networks are trained. If the RBF and BP rankings had agreed with those of the VQ models, then it would have been possible to use the VQ model rankings to find problematic speakers. Unfortunately, the rankings did not agree. There was strong similarity between the RBF and BP rankings themselves. There was also strong agreement between the VQ rankings using scores and those using models.

4.2.5 Reducing the impostor set

In a final experiment, the effect of reducing the set of impostors used in training the neural networks was investigated. This was done to investigate what might happen in a larger database, where it would not be possible to include utterances from all the impostors. To do this RBF networks were trained for each speaker using the first training set of the previous experiments, but only including the first 10 impostors for the impostors' training utterances, i.e. Speakers 1–10 were used as impostors for Speakers 11–20 and Speakers 1–11 (excluding the genuine speaker) for Speakers 1–10. This would significantly reduce the amount of impostor and, hence, inter-speaker information presented to the artificial neural network.

4.2.5.1 Results

The results for these reduced training sets are compared to the results using the entire impostor population in Table 4.4. It is clear that reducing the impostor information has greatly reduced the accuracy of the speaker model. The only result which shows an improvement is the false rejection percentage. As the speaker model had less inter-speaker variation data to train on, genuine speaker utterances that may have been rejected before are now accepted. However, this also means that more impostors are accepted than before and thus the false acceptance ratio, the total error, the identification error and the d' measure all deteriorated.

	d'	Error(%)	FR(%)	FA(%)	Total Error(%)
All impostors	7.7	1.1	5.7	0.2	0.4
Limited set	6.8	1.8	3.6	0.8	0.9

Table 4.4: Results for reducing the number of impostor utterances in the training set.

4.2.5.2 Discussion

These results indicate a possible limitation of classifying artificial neural networks for speaker recognition using large databases, where it may not be possible to incorporate information about each impostor in the training set. The effect of removing some of the impostors from the training set was to increase all error rates except the false rejection. The reduction in inter-speaker information used to train the network reduces the ability of the network to discriminate between genuine speaker and impostor utterances.

The reason why this is important is that in a real system, with a large population of speakers, it would be impractical to incorporate all impostors in the training set. To overcome this would require some means of creating training sets that incorporated sufficient impostor utterances to represent the impostor population in general.

4.3 Discussion of the classifying results

The results of the experiments indicate that classifying neural networks perform well for a small-scale text-dependent database. However, there were indications that a larger database might pose problems for the networks as implemented for these experiments. The reason for this is that a training set composed of significantly more impostor utterances than genuine speaker utterances could lead to a network which fails to generalise well. To overcome this a more specialised training set, which has a better balance between genuine speaker and impostor utterances, would be required for each speaker.

One possible solution to this problem is to use a reduced set of impostors, preferably the ones most likely to impersonate the genuine speaker. However, the tests with the VQ models showed that the relationship between speakers, according to the neural networks, wasn't a simple matter of having similar cepstral

vectors. The weighting of the neural networks affected the speakers that appear to be similar each other, so that the impostors could only be ranked *a posteriori*, after the network had been trained and tested.

There is also the issue of a continuously-varying database with speakers frequently being added or removed from the database. Having to retrain networks regularly or create new ones to deal with the addition of new speakers may not be practical. On account of this, Artières and Gallinari (1993) moved from work on classifying neural networks to predictive neural networks (PNN), which are free from this problem because they are trained using genuine speaker data only. For this reason predictive neural networks were also investigated, the results of which are presented in Chapter 7. However, before that, score normalisations for distance models (such as PNN) are investigated. This is necessary as distance model verification error rates may be reduced through the use of score normalisations.

4.4 Summary

This chapter has described experiments to assess the use of classifying artificial neural networks for speaker recognition. Although they give very good results on a small closed set of speakers, it is possible that larger database may present problems with modelling the impostor set. The next chapter looks at improving distance model results by using impostor cohorts for score normalisation.

Chapter 5

Score normalisation for distance models

5.1 Introduction

The previous chapter detailed experiments using speaker models that classified an utterance as belonging to the genuine speaker or an impostor. The models use both genuine speaker and impostor utterances in training, and internal weights emphasise the difference between the two. In this chapter, the speaker models are created using only genuine speaker utterances. No impostor utterances are used in the speaker model and no weighting is performed to emphasise any differences. Rather than classifying the utterance, the distance between the utterance and the speaker model is calculated. The closer the utterance to the model, the more likely it is to come from the genuine speaker and vice versa.

The distance of an utterance from the genuine speaker model needs to be quantified. In speaker identification this is done by comparing the scores from all the speaker models, with the best scoring model determining the speaker. However, in speaker verification the test utterance is only presented to the genuine speaker model, so some reference is required to determine whether the score is close enough to belong to the genuine speaker or not. Normally this is provided by a threshold, which is determined from the training data. However, the training data usually produce better scores than the test data. Furthermore, variations in speaking behaviour and recording conditions, etc. may cause wide variations in scores and make the assignment of thresholds even more difficult (Rosenberg *et al.*, 1992b).

Score normalisation takes a different approach to that of setting a threshold. It compares a measure of the likelihood of the utterance belonging to the genuine speaker with that of it belonging to an anti-speaker (i.e. an impostor). If the likelihood is greater that the utterance belongs to the genuine speaker than the anti-speaker then it is accepted and vice versa. The anti-speaker model may be represented by a cohort of impostors or a single 'world' model. The score against the anti-speaker model is used to normalise the genuine speaker score. This brings all scores into a similar range and more importantly increases the separation between the genuine speaker and impostor score distributions (c.f. section 3.7.2), thus reducing the verification error rate.

This chapter compares several normalisation methods, with the intention of choosing one for the distance-model experiments. The initial sections describe how the anti-speaker model may be created using impostor cohorts or a world model. Then a new means of ranking a speaker's impostors is presented, which may be used to select impostors for speaker-specific cohorts. Finally, the experiments investigate different approaches to normalising the scores.

The majority of experiments in this chapter were carried out on a single word, *one*, from the British Telecom Millar database. The first 10 utterances were used for training and the remaining 15 for testing. Each speaker was modelled using a 32-element vector quantisation codebook. Yu *et al.* (1995), using the same database and training sets, found a codebook of this size to be sufficient to model a speaker.

5.2 Score normalisation

In a speaker verification system without score normalisation, the decision to accept or reject a claimed identity is based on comparing the score from the genuine speaker model with a threshold. However, using a Bayesian formulation of the verification problem, the claimed identity would be accepted if it was more likely to belong to the genuine speaker than not (Rosenberg *et al.*, 1992b):

$$p(\mathbf{U} \mid I) > p(\mathbf{U} \mid \bar{I})$$

where $\mathbf{U}$ is the sequence of vectors from the test utterance and $p(\mathbf{U} \mid I)$ is the likelihood of the utterance belonging to the claimed identity I and $p(\mathbf{U} \mid \bar{I})$ the likelihood for any identity other than I.

The likelihood of the utterance not belonging to the genuine speaker is measured by the likelihood of it belonging to an anti-speaker (in effect, an impostor). In the following experiments, two methods of implementing a generalised impostor model are investigated. The first uses a cohort of impostor models and the second a single model of the impostor set. In the work presented here, the cohort may or may not be speaker-specific (though normally it is). The second approach uses a single model of the impostor population to represent the anti-speaker, and is referred to as world-model normalisation. This means that the world-model is the same for all speakers, while the cohort method may be varied to suit each speaker. The other difference between the cohort and world model normalisations is that for cohort normalisation the utterance must be tested against several impostor models, whereas for the world-model approach the utterance is only presented to a single anti-speaker model.

5.2.1 Cohort normalisation

In cohort normalisation, the anti-speaker model is represented by a subset of impostors. The scores from these impostors may be used to normalise the genuine speaker score in a number of ways. Li and Porter (1988) and de Veth, Gallopyn and Bourlard (1993) used the mean and standard deviation of the scores to normalise the genuine speaker score, Booth *et al.* (1993) and Matsui and Furui (1992) used the mean score, while Rosenberg *et al.* (1992b) investigated various measures such as the best score, the mean and the median scores. Both the normalisation using the mean and standard deviation, and that using the mean on its own are investigated in the experiments.

Following the work of Higgins, Bahler and Porter (1991), the cohort is usually made up of the closest impostors to the genuine speaker, though this may be done *a priori* or *a posteriori*. In the *a priori* case, each speaker's training utterances are used to test their similarity to other speakers, and a list of ranked impostors is drawn up for each speaker. This method has been used by de Veth *et al.* (1993) and Rosenberg *et al.* (1992b). The *a posteriori* approach presents the test utterance to all the impostor models and takes the models with the best scores. This is more time-consuming than the previous method, as the utterance must be tested against all models rather than just a limited cohort. This method has been used by Booth *et al.* (1993), Higgins *et al.* (1991) and Matsui and Furui

(1992).

The advantage of the cohort method is that it can be made speaker-specific by choosing a different cohort for each speaker based on their closest impostors (Rosenberg *et al.*, 1992b). This also means that the cohort may be updated if new speakers are added to the database who are good impostors of a genuine speaker. Speaker-independent cohorts were also investigated.

5.2.2 World-model normalisation

In this approach, favoured by Carey and Parris (1992), Hattori (1994) and Matsui and Furui (1995), the anti-speaker model is represented by a single world model. A subset of impostor utterances is used to generate a single model, which is then used to normalise all speaker scores. Although it is possible to update the world model for the addition of new speakers (Matsui and Furui, 1995), it still remains speaker-independent, unlike the cohort normalisation which is normally speaker-dependent.

5.3 Impostor ranking

Rosenberg *et al.* (1992b) and de Veth *et al.* (1993) used the *a priori* closest ranking impostors for each speaker's impostor cohort. In effect, they selected the impostors with the smallest inter-speaker distance between them and the genuine speaker. This was done by testing all the impostor training utterances against that speaker's model. This can be time-consuming if there is a large number of training utterances or impostors. So a new approach was taken, whereby only one test is performed per impostor. This is done by comparing the speaker models with each other. In effect, each impostor speaker model is treated as a generalised test utterance for that impostor. Testing with the model generates a *single* score for each impostor which may be used to determine the impostor's ranking against the genuine speaker.

Examples of using *model vs. model* tests are shown in Figures 5.1 to 5.4. The *model vs. model* scores are represented by '$*$' and the *test vs. model* scores by 'o'. It is important to remember that the test utterances come from the final three recording sessions while the models were created using data from the first two sessions (c.f. section 2.5). Despite this inter-session variability, which

can be so detrimental to speaker verification, the *model vs. model* scores would appear to give a very good indication of the final impostor rankings – being generally positioned at or below the low end of the *test vs. model* distributions. (As Speaker 8 is actually the genuine speaker, the *model vs. model* score is zero in this case.)

To determine how good the *model vs. model* rankings are, we require some *reference* rankings for comparison. Unfortunately, there is no definitive way of ranking impostors. Even using final, *a posteriori* test scores, the rankings could be arrived at on the basis of the best impostor score, the mean impostor score, the number of impostor scores below the threshold, etc. Clearly, rankings based on different criteria are unlikely to agree completely with each other, so that small differences should be expected when comparing any two rankings. In these experiments, all reference rankings were determined on the basis of the mean test utterance score per speaker.

The average difference between the *model vs. model* and reference rankings was calculated as follows. First, the absolute difference between an impostor's position, r_m, in the *model vs. model* ranking and that person's position, r_r, in the reference rankings is calculated as $|r_m - r_r|$. This is done for each impostor in the rankings and the average calculated. Once this difference has been evaluated for each speaker, it is then averaged across all speakers. The smaller this value, the closer is the agreement between the ranking method and the *model vs. model* scoring strategy under investigation. Using simple combinatorics, it is easily shown that – since there are only 30 possible ranking positions – the upper bound on the possible average difference is 15 places. When 100 completely random rankings (uniformly distributed) were tested against each other, the average distance was 9.7 places. Thus, if the rankings were in the region of this score, it would indicate that there was no relationship between them.

The average ranking differences for the ten test words are shown in Figure 5.5. The *model vs. model* rankings are compared to three reference rankings, generated by taking the mean score of all possible utterances (both test and training), of the training utterances alone and of the test utterances alone. The figure shows that the *model vs. model* rankings are closest to the training utterance rankings. This is only to be expected, as they were used to create the models. Also, as expected, differences based on the test utterances alone were furthest from the *model vs.*

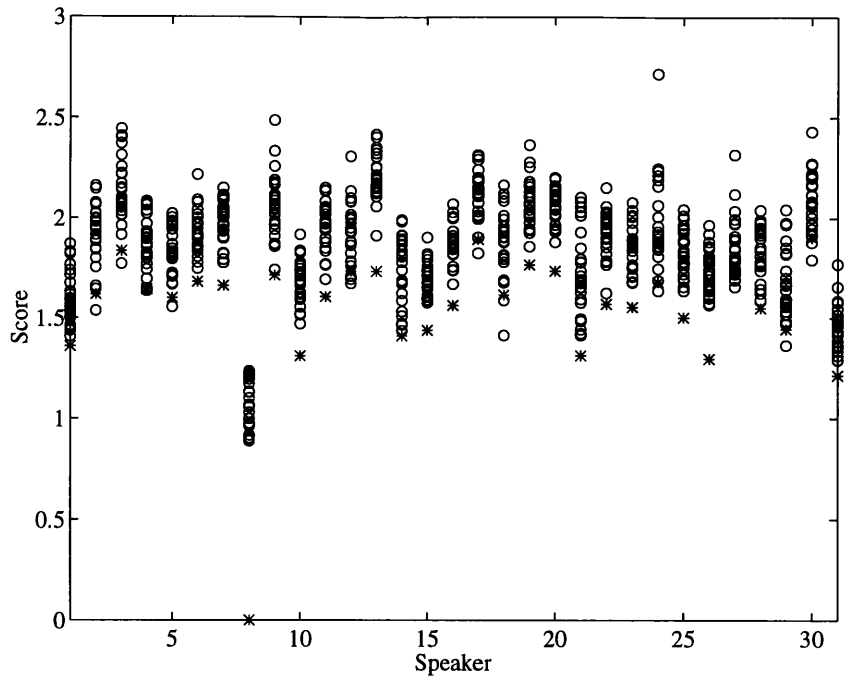

Figure 5.1: Utterance ('o') and *model vs. model* ('*') scores for Speaker 8, word *one*.

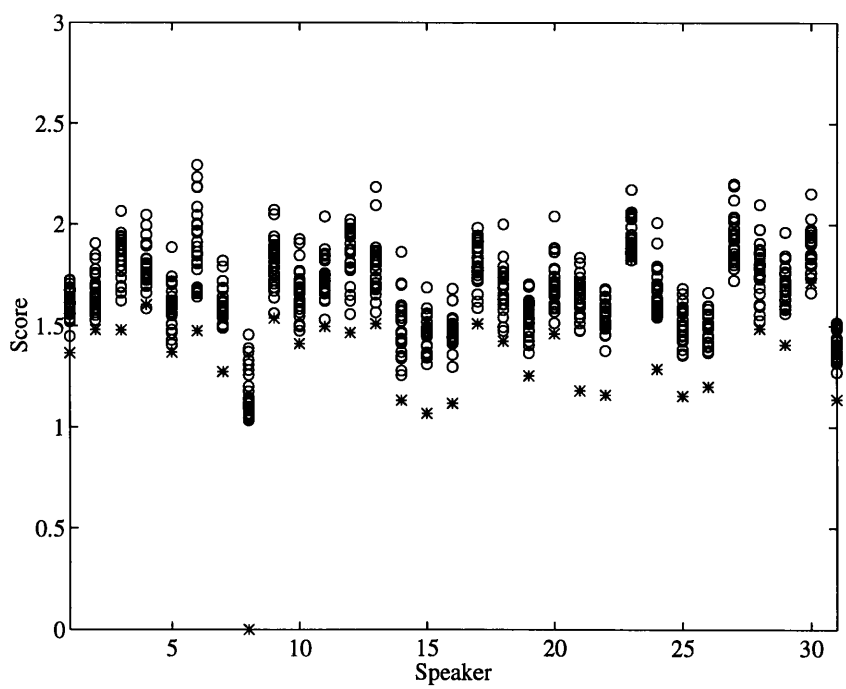

Figure 5.2: Utterance ('o') and *model vs. model* ('*') scores for Speaker 8, word *three*.

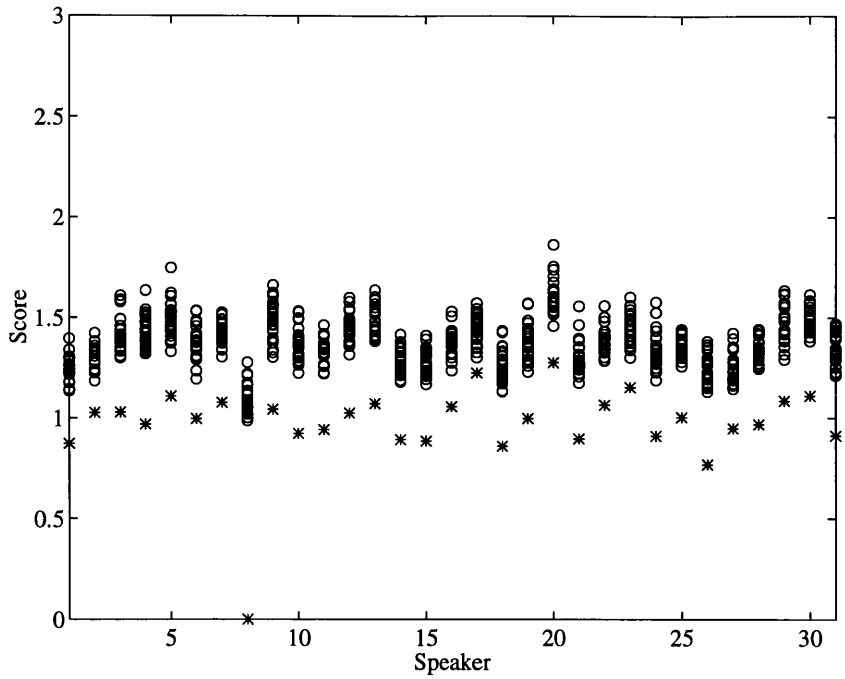

Figure 5.3: Utterance ('o') and *model vs. model* ('*') scores for Speaker 8, word *six*.

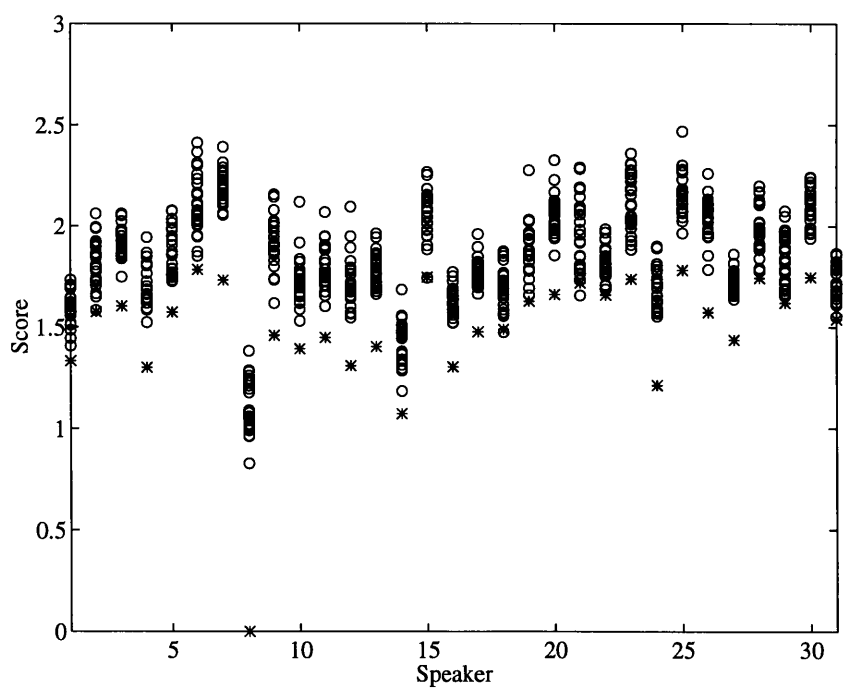

Figure 5.4: Utterance ('o') and *model vs. model* ('*') scores for Speaker 8, word *zero*.

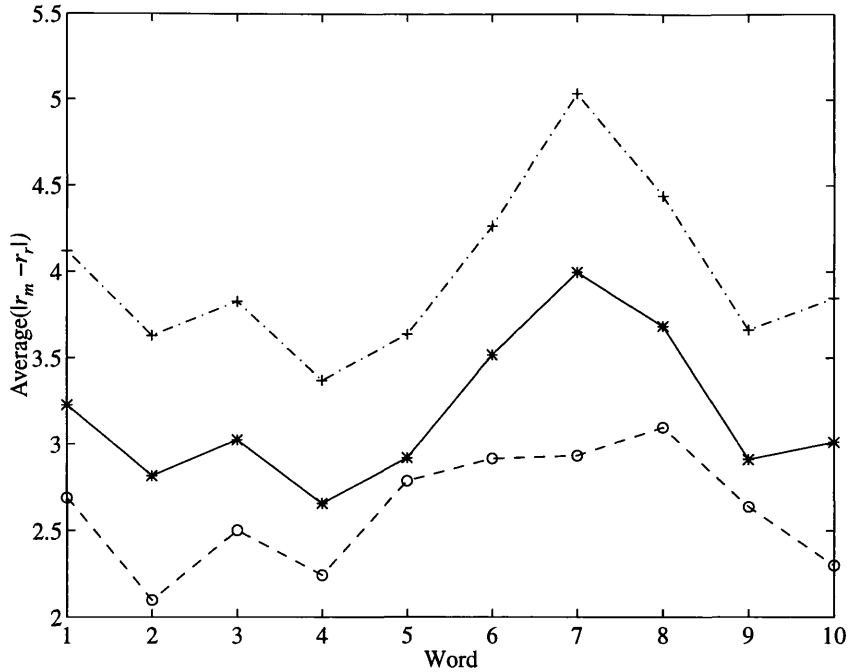

Figure 5.5: Average difference in *model vs. model* and reference rankings for each word. There are 3 different reference rankings: '—' denotes rankings based on all possible utterances, '– –' denotes those based on the training utterances alone and '– · –' denotes rankings based on the test utterances alone. See text for specification of $|r_m - r_r|$.

model scores, while the combination of both test and training utterances gave differences somewhere in between. It should be emphasised again that none of these reference rankings is in any way definitive, as the mean of the test scores may not be an exact measure of an impostor's likelihood of impersonating a speaker. However, the average difference in ranking of about 4 per word is significantly lower than the value of 9.67 generated by random rankings. So, although by no means perfect, the *model vs. model* rankings were judged to be sufficiently good to make it worthwhile creating and evaluating impostor cohorts.

The same tests were run, but for training sets that used the first 2 utterances from each of the 5 recording sessions. These training sets should generate models that are more representative of the test utterances as the temporal variation will not be as pronounced. The results for the difference in ranking are given in Figure 5.6. In this case, the difference between rankings for the training utterances and the test utterances is much smaller. Furthermore, the overall difference in ranking of around 3 is smaller than that for the original training sets.

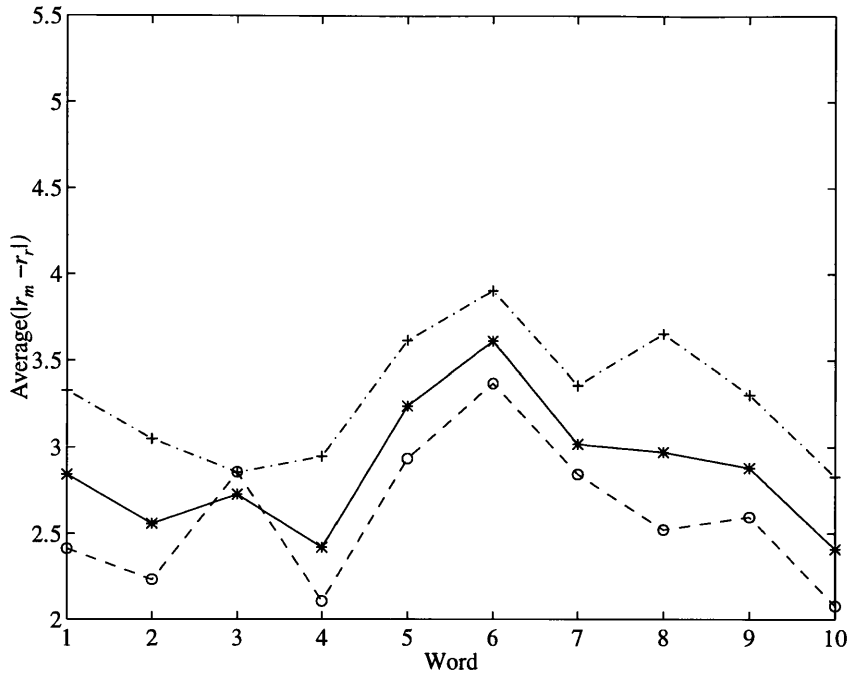

Figure 5.6: Average difference in *model vs. model* and reference rankings for each word, using the temporal variation models. There are 3 different reference rankings: '—' denotes rankings based on all possible utterances, '– –' denotes those based on the training utterances alone and '– · –' denotes rankings based on the test utterances alone. See text for specification of $|r_m - r_r|$.

The rankings also showed for the VQ case, that just because Speaker A was a good impersonator of Speaker B this did not mean that Speaker B would be a good impersonator of Speaker A. That is, if Speaker A was ranked at position i against Speaker B's model, and Speaker B was ranked at position j in Speaker A's rankings, then $i \neq j$ in general. Figure 5.7 shows the difference $|i - j|$ for a particular impostor position averaged across speaker pairs. The value is quite different from zero – indeed, it may be anything from 5 to 9 places. The implication for normalising using impostor cohorts is that although Speaker A may be similar to Speaker B, this does not mean that an impostor cohort selected for Speaker A would be suitable for Speaker B.

It was also found that impostors vary in their ability to impersonate the genuine speaker depending on the word in question. This is shown by the fact that, for a particular speaker, the ranking of the impostors varies from word to word. To measure this, the absolute difference between rankings according to a particular word, r_w, and every other test word, r_t, was calculated as $|r_w - r_t|$.

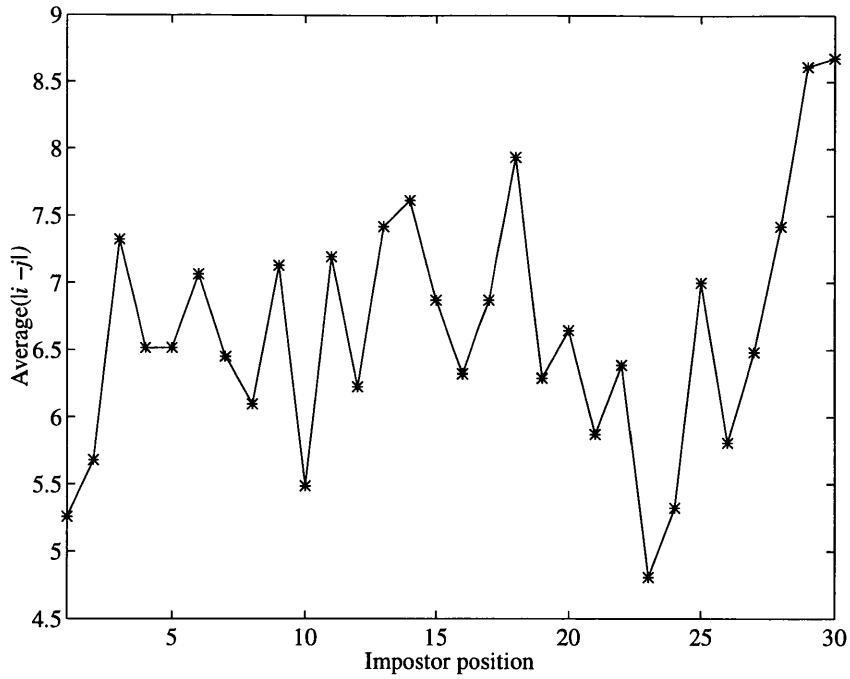

Figure 5.7: The average ranking asymmetry when comparing two speakers; i.e. comparing Speaker A as an impostor of Speaker B with Speaker B as an impostor of Speaker A. See text for specification of $|i - j|$.

This was then averaged across the remaining 9 test words. The results are illustrated in Figure 5.8, where the rankings are seen to vary from around 7 to 8.6 places – indicating that impostor position varies from word to word. Thus, testing with more than one word should be beneficial in lowering the error rates. The likelihood of accepting an impostor will be reduced in this case, as it is unlikely that any impostor will consistently impersonate the genuine speaker across all test words.

5.4 Normalisation experiments

The following experiments investigate various means of normalising scores. Particular attention is paid to cohort normalisation, where the *model vs. model* rankings are tested as a means of selecting speaker-specific cohorts.

Although there are 12 words in the database, only the word *one* was used for the normalisation experiments. The average d' for the word *one* was 3.8, slightly higher than the average d' for all the words of 3.6. The success of the normalisations was measured using the d' as an indicator of the verification

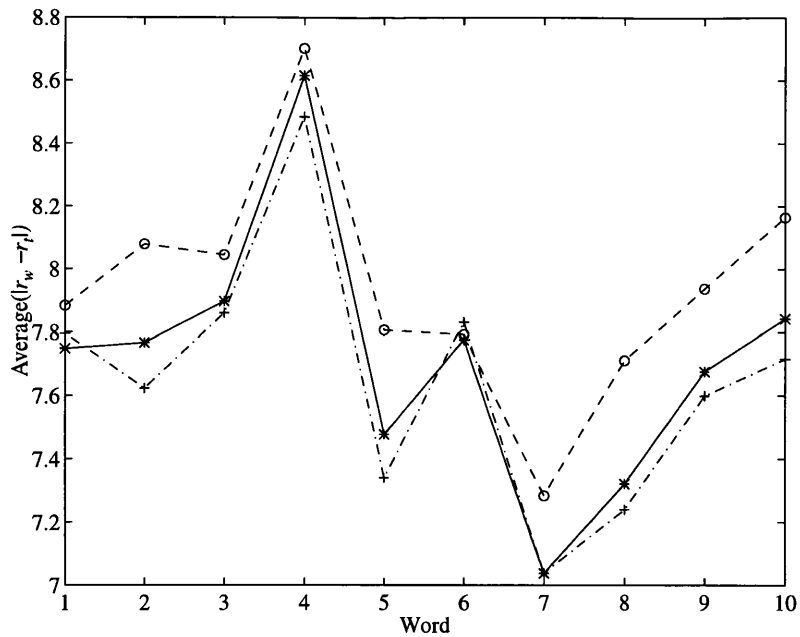

Figure 5.8: Average difference between ranking position on the basis of a specific word, r_w, and ranking position on the basis of every other test word, r_t. There are 3 different reference rankings: '—' denotes rankings based on all possible utterances, '– –' denotes those based on the training utterances alone and '– · –' denotes rankings based on the test utterances alone. See text for specification of $|r_w - r_t|$.

error rate (i.e. the separation between the genuine speaker and impostor score distributions) and the identification error rate. More comprehensive tests for the most promising normalisation method, using the words *one* to *nine* and *zero*, are presented in the next chapter (section 6.2).

5.4.1 Subtracting the mean of the impostor cohort

This method of score normalisation (SMIC) has been used by Booth *et al.* (1993), Matsui and Furui (1992) and Rosenberg *et al.* (1992b). Rosenberg *et al.* (1992b) tried several statistical measures of the impostor cohort to normalise the genuine speaker model score and found that the mean was the best. So the mean score from a set of impostor models is used to normalise the score against genuine speaker model as follows:

$$S_{norm} = S_{gen} - \mu_{cohort}$$

where S_{norm} is the normalised score, S_{gen} is the original score against the genuine

speaker model and μ_{cohort} the mean of the impostor cohort scores.

Matsui and Furui (1992) used the *a posteriori* best scoring impostors, which meant testing each speaker model with the test utterance. In tests using hidden Markov models, Rosenberg *et al.* (1992b) used the *a priori* nearest impostors, which were determined by testing each speaker's training utterances against every other speaker's model. This means of determining the closest impostors is more time-consuming than testing the models against each other, but the latter is not possible with hidden Markov models. Both *a priori* and *a posteriori* approaches were tested in the following experiment.

5.4.1.1 Results

The results for SMIC are shown in Figures 5.9 and 5.10. The x-axis represents the number of speakers used to form the cohort.

Two methods were used to create the impostor cohort. The first results, shown in Figure 5.9, used the impostors with the lowest scores to form the cohort (i.e. the *a posteriori* approach). This meant testing an utterance against all the speaker models in order to determine the lowest scoring models. In a large database, this could be very time-consuming. The method works well in improving the d' from 3.8 to 4.8 for a cohort of one or two speakers. However, as more impostors are added to the cohort, the d' begins to decrease, though it always remains higher than that of the unnormalised results. The method leaves the identification error rate unchanged, as the order of the scores remains the same (the lowest scoring model will always retain the lowest score).

The second method used the nearest impostor models, as determined by the impostor ranking (c.f. section 5.3), to create the impostor cohort (i.e. the *a priori* approach). This has the advantage of not testing against all the speakers in the database. However, as Figure 5.10 shows, the identification error is higher for this method than for the unnormalised case. To benefit from the improvement in d', while still keeping the identification error within reasonable limits, a cohort of about 15 speakers is required.

5.4.1.2 Discussion

This improvement in results has also been found by other researchers. Matsui and Furui (1992) achieved a reduction in equal error rate from 3.6% to 1.1% using

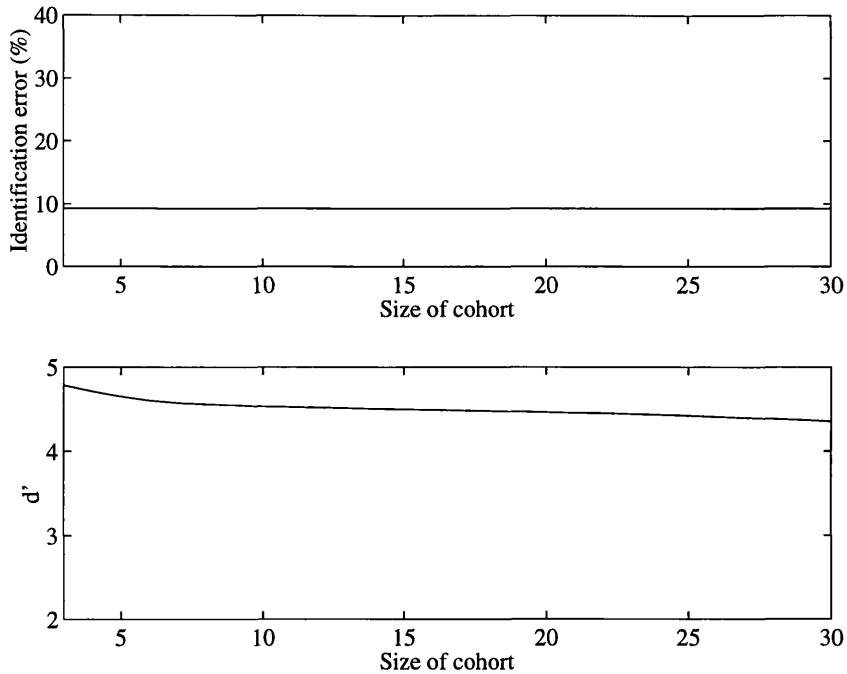

Figure 5.9: Results for the SMIC normalisation, using the lowest scoring impostors.

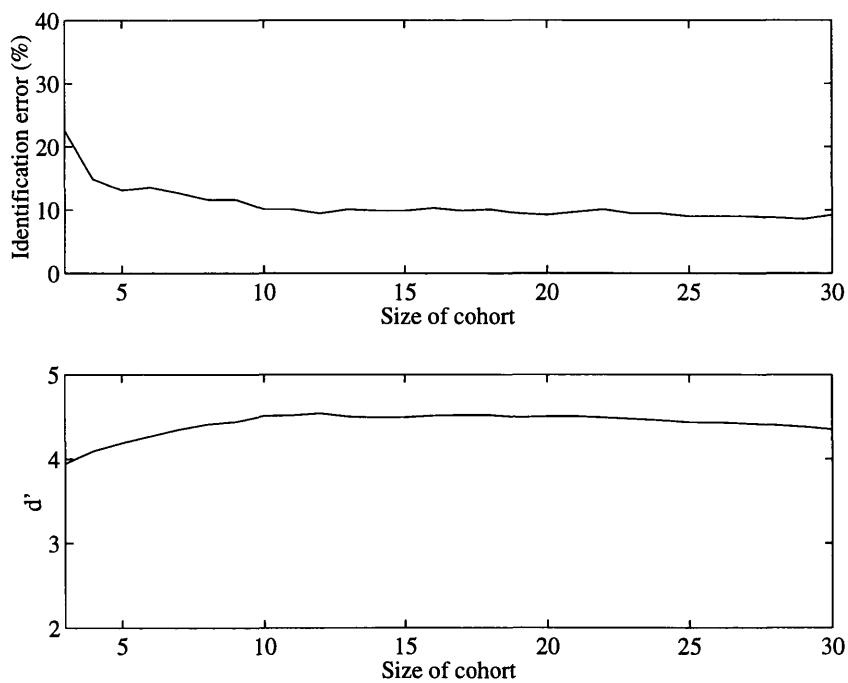

Figure 5.10: Results for the SMIC normalisation, using the closest impostors for each speaker.

an *a posteriori* cohort of 15 impostors. Using an *a priori* cohort of 5 impostors, Rosenberg *et al.* (1992b) reduced the equal error rate from 2.9% to 1.8%.

The *a posteriori* approach gave better results, with a maximum d' of 4.8 (for a cohort of one or two impostors) compared to 4.5 for the *a priori* method. However, the limitation of the *a posteriori* method is that the good scores for the small impostor cohorts are due to the fact that the impostor's model is present in the cohort and, hence, generally gives a lower score for the utterance than the genuine speaker model. This is therefore a closed set only technique, equivalent to the SLIMS method described in section 5.4.3.

5.4.2 Impostor cohort normalisation

An alternative to using the mean cohort score is to use the mean and the standard deviation of the cohort scores to normalise the genuine speaker score. This approach is called impostor cohort normalisation (ICN) and was put forward by Li and Porter (1988). It is based on the fact that the genuine speaker score remains fairly stable relative to the impostor score distribution, although the impostor score distribution itself may vary considerably (Li and Porter, 1988). However, rather than presenting the test utterance to all possible impostors in order to determine the impostor score distribution, it is presented to only a limited number of impostors. This means that the method is independent of the number of speakers in the database.

For impostor cohort normalisation, each utterance is presented to the genuine speaker model and a limited set of impostor models. The genuine speaker score is then normalised by the mean and standard deviation of the impostor scores as follows:

$$S_{ICN} = \frac{S_{gen} - \mu_{cohort}}{\sigma_{cohort}}$$

where S_{ICN} is the normalised score, S_{gen} is the original genuine speaker model score and μ_{cohort} the mean and σ_{cohort} the standard deviation of the impostor cohort scores. This is in contrast to the previous method, SMIC, where there is no division by the standard deviation. ICN scores are measured as the number of standard deviations from the mean, whereas the SMIC scores are the distances from the mean impostor score. This means that all ICN scores must fall in a similar range, which may not be true of the SMIC scores.

There is still the requirement to set a threshold, after the scores have been normalised, but the variation in threshold between speakers is much less. This means that in principle, a common threshold may be set for all speakers. A comparison between using a common threshold and a separate threshold for each speaker is given in the next chapter (c.f. section 6.2.5).

Two factors were varied for the impostor cohort normalisation: the number of impostors in the cohort and the means of selecting impostors for the cohort. The number of impostors in the cohort was varied from 3 to 30. The selection of the impostor cohort was tested in 3 ways.

In the first approach the impostor cohort was selected from the ranked impostors. Starting with the most likely impersonators, the cohort was increased using the next most likely impersonator. In the second case, a fixed set of impostors was used for each speaker. This set was created using the ranking of the speaker models (according to their average impostor model score), starting with the lowest scoring models. This approach is closer to world-model normalisation as it uses the same anti-speaker model for all speakers. The last approach used a random set of impostors to create a cohort, which was then used for all speakers. This was to test if it was necessary to rank the impostors, or whether enough randomly selected impostors would give equally good results.

5.4.2.1 Results

The results for using the *model vs. model* ranked impostors are shown in Figure 5.11. Although the identification error rate is higher than the unnormalised results to start with, the errors drop off quite quickly and by about 15 impostors in the cohort the error rate (10.3%) is close to that of the unnormalised results (9.3%). However, d' for an impostor cohort of 15 (4.5) is considerably larger than that for the unnormalised results (3.8). Of the two error measurements d' is more important than the identification error, as it is the closest indicator of the average equal error rate. So, although the identification error is slightly higher for the normalised case, the average equal error rate is likely to be lower.

The results for the second approach of using a fixed cohort are shown in Figure 5.12. Although the identification error for this impostor cohort is similar to that using the nearest ranked impostors of the last selection method, d' is much poorer. Using 15 impostors, d' is 4.0. This would seem to indicate that this

particular selection of impostors is not particularly representative of the impostor scores in general.

The results of using randomly selected impostors for the cohort are shown in Figure 5.13. In general this method worked well, doing better than the fixed set of the previous method. However, it lacked the consistency found using the nearest ranked impostors and is prone to some high identification errors.

5.4.2.2 Discussion

The results of all 3 methods of selecting the impostors show how important it is to have at least 10 impostors, so that the standard deviation used to normalise the results is meaningful. The scores as a result of this normalisation are measured in standard deviations away from the impostor mean. The number of impostors in the cohort was not as important a factor in the SMIC method which does not normalise by the standard deviation, but the scores from this approach are not limited in their range, as they only measure the distance between the genuine speaker score and the mean impostor score.

5.4.3 Subtracting the lowest impostor model score

In this normalisation method, referred to as SLIMS, the lowest of the impostor scores is subtracted from the genuine speaker model score. In effect this uses identification results to normalise the scores for verification. It works on the basis that if an utterance is a genuine speaker utterance then the lowest score for the impostor models will be higher than that for the genuine speaker model. Subtracting the two values will give a negative score. If, however, the utterance is an impostor utterance, the impostor score should be lower than the genuine speaker score and the difference between the two will be positive. This, in effect, polarises the results, with any score below zero being a genuine speaker utterance and any above zero an impostor utterance. A threshold of zero is thus created. Although the zero threshold could be used, the false rejection rate would probably be too high, as it would be equal to the identification error rate. This means that a threshold slightly greater than zero must be found to reduce the false rejection rate.

However, this method of normalisation only works for a closed set as it is dependent on the impostor being one of the speakers in the database. In the

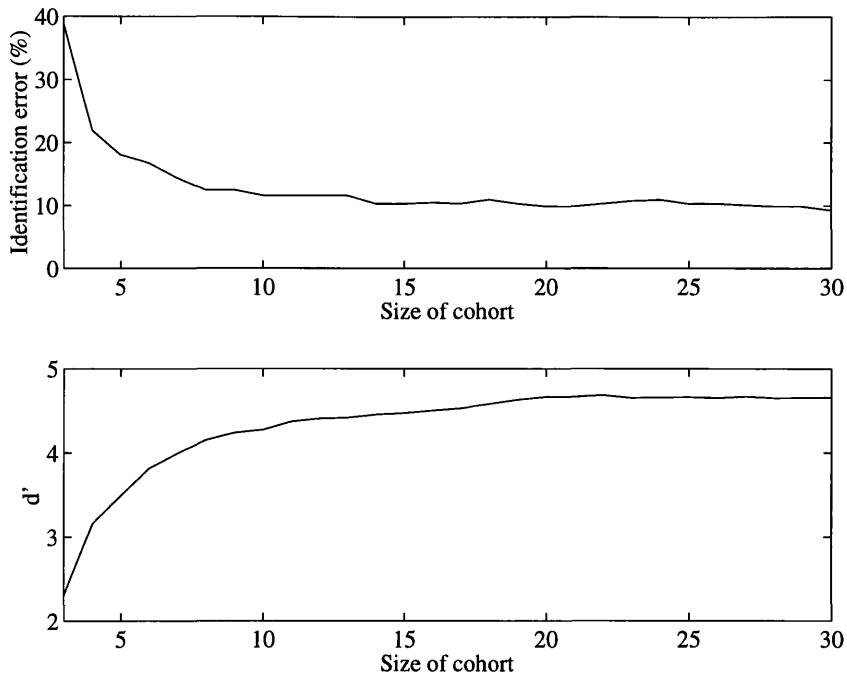

Figure 5.11: Results for impostor cohort normalisation using the impostor rankings.

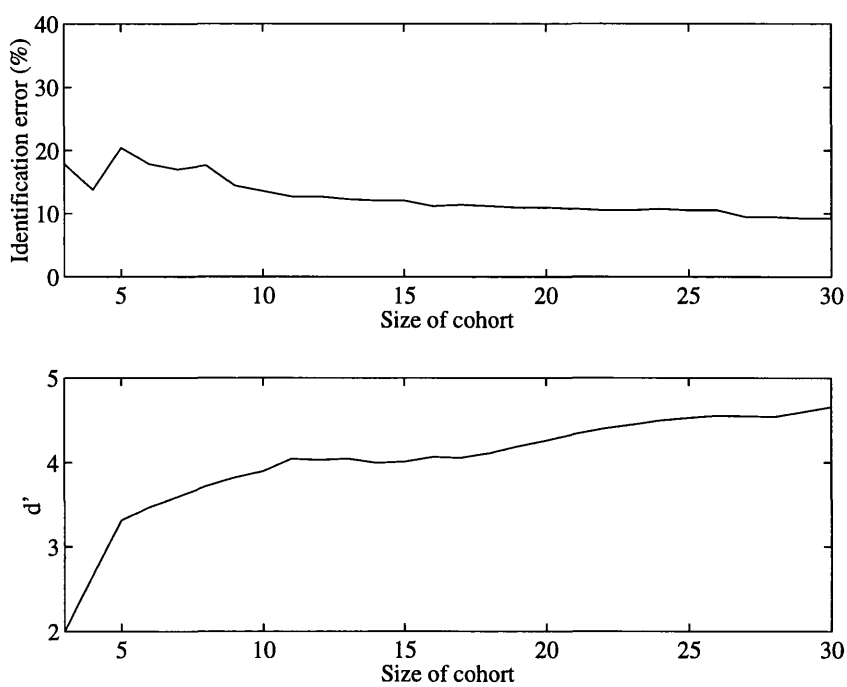

Figure 5.12: Results for impostor cohort normalisation using a fixed set based on model ranking.

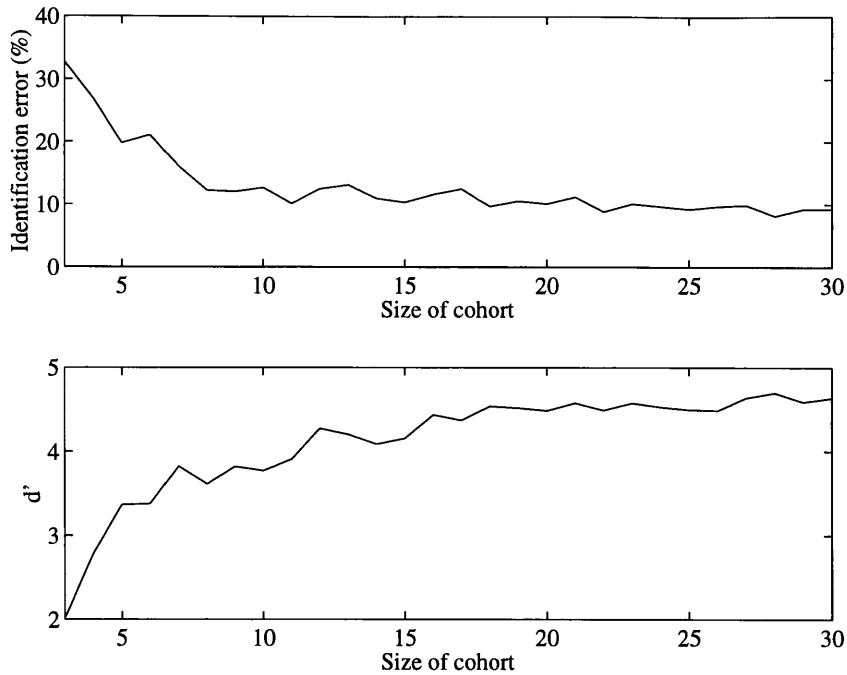

Figure 5.13: Results for impostor cohort normalisation using randomly selected impostors.

	d'	IE(%)
VQ	3.8	9.3
VQ+SLIMS	5.0	9.3

Table 5.1: Comparison of the unnormalised and SLIMS results (IE = identification error).

open set case, if an impostor's model is not present, then the lowest impostor model score is less likely to be lower than the genuine model score, and so the relationship breaks down.

5.4.3.1 Results

The results using SLIMS are presented in Table 5.1. Subtracting the lowest impostor model score from the genuine speaker model score has no effect on the identification error as the ranking of the lowest scores will always stay the same. However, it does improve the d' from 3.8 to 5.0.

5.4.3.2 Discussion

The benefit of this method may be seen in Figures 5.14 and 5.15. The figures give an indication of the degree of overlap between the genuine and impostor speaker distributions for the unnormalised and the SLIMS cases. The scores for the genuine speaker ('o') represent the sum of the average score and the standard deviation of the genuine speaker's test scores (i.e. the scores lying furthest from the genuine speaker model). The scores for the impostors ('$*$') represent the average impostor score minus the standard deviation of the impostor scores (i.e. the scores closest to the genuine speaker model). Thus the scores are a measure of how close distributions are to each other. Inspection of the plots shows that the normalisation has improved the separation for problem speakers such as Speakers 2, 3 and 20, where the unnormalised system had pronounced overlap between the genuine speaker and impostor score distributions.

5.4.4 Comparison of VQ and RBF results

In order to quantify the results and to compare them with the classifiers of the previous chapter, some tests for the word *one* were run using an RBF classifier. The network configuration was very similar to that described in Section 4.2. The true speaker training set consisted of the same utterances as those used for the VQ codebook generation. A single random utterance was taken from each of the 30 possible impostors as the impostor training set.

In a second test, the RBF network and the VQ system were trained with a training set containing utterances from all five recording sessions, known as the temporal variation (TV) training set. The first two utterances from each session were used to create the training set, so that it had 10 utterances altogether. This is similar in size to the original training set, but the fact that utterances from later recordings are included increases the amount of intra-speaker variation present.

The VQ system used the 32-element codebook. The results were evaluated for the ICN and SLIMS conditions.

5.4.4.1 Results

The results of these tests are shown in Table 5.2. Although the RBF network gives better results than the VQ system, the improvements are not dramatic.

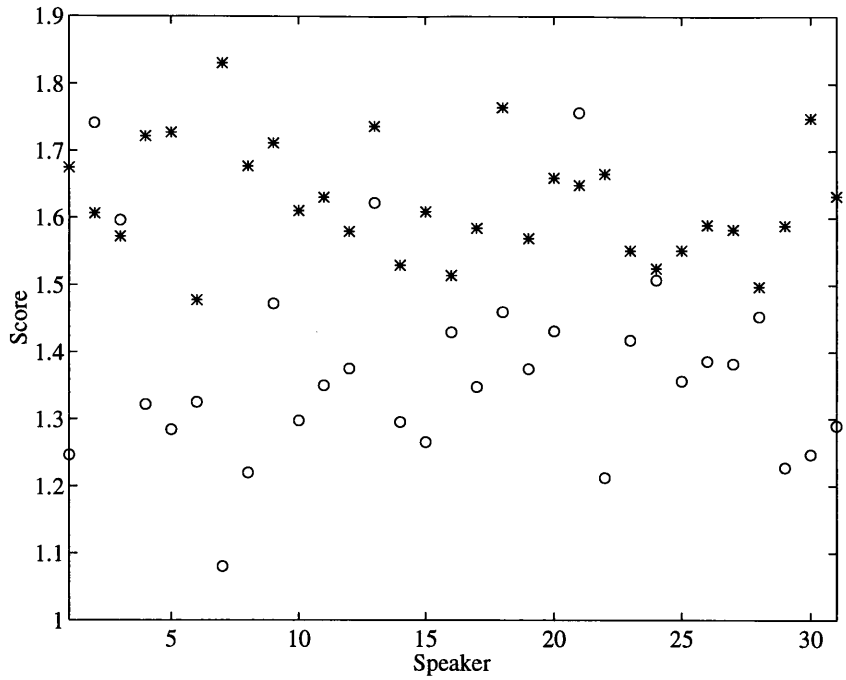

Figure 5.14: Degree of overlap between impostor ('*') and genuine speaker ('o') distributions without normalisation.

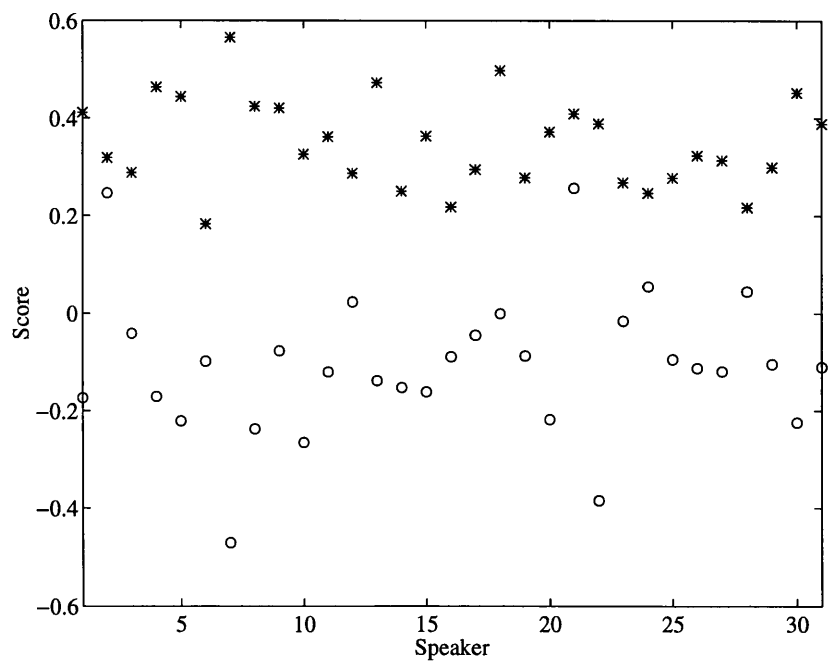

Figure 5.15: Degree of overlap between impostor ('*') and genuine speaker ('o') distributions with SLIMS normalisation.

	d'	IE(%)	FR(%)	FA(%)	ERR(%)
VQ(32)+ICN	4.5	10.3	4.3	4.4	4.4
VQ(32)+SLIMS	5.0	9.3	2.8	2.9	2.9
RBF	5.6	8.4	3.7	3.8	3.8
RBF+SLIMS	4.8	8.4	2.4	2.4	2.4

Table 5.2: Comparison of VQ and RBF results (IE = identification error, FR = false rejection, FA = false acceptance, EER = average equal error rate).

	d'	IE(%)	FR(%)	FA(%)	EER(%)
VQ(TV)+ICN	5.4	1.9	0.7	0.8	0.8
VQ(TV)+SLIMS	6.3	1.1	0.2	0.2	0.2
RBF(TV)	7.9	1.1	0.7	0.7	0.7
RBF(TV)+SLIMS	8.2	1.1	0.2	0.2	0.2

Table 5.3: Comparison of temporal variation VQ and RBF results (IE = identification error, FR = false rejection, FA = false acceptance, EER = average equal error rate).

The average equal error rates for the RBF and VQ systems vary less than the d's. However, the high d's of the RBF networks represent the large separation in score distributions that we saw in the classifier results (Figure 4.2). This polarisation of results is not possible with the VQ system as it did not use discriminative training. Therefore, one would never expect the VQ d's to be as big as the RBF d's. In this case, d' is not a suitable measurement for system comparison.

5.4.4.2 Discussion

There are greater differences between the RBF and VQ results when the standard training set is used than when the temporal variation training set is used. This would seem to indicate that the RBF has learned to generalise and, hence, can cope with the unseen recording sessions better than the VQ system, which has less ability to generalise.

5.4.5 Normalisation by a world model

Normalisation by a world model is an alternative to cohort normalisation. In the text-dependent case it involves creating a model of the utterance using several

speakers' utterances. The test utterance's score against this world model is then used to normalise its score against the genuine speaker model. This is done by dividing the utterance score against the genuine speaker model by that against the world model, i.e. the ratio of the two distances.

This normalisation method is used quite often in HMM and PNN systems (Carey and Parris, 1992; Hattori, 1994; Matsui and Furui, 1995). As Carey and Parris (1992) and Matsui and Furui (1995) both used HMM systems, whose scores are in the form of log likelihoods, the world model score was subtracted from the genuine speaker score. In Hattori (1994) the PNN returns a distance score, and, as in this work, the ratio between the genuine speaker score and the world model score was used as the normalised score.

The experiments in this section were carried out using a reduced database of 12 speakers, which is described in detail in section 8.3. However, ICN was also implemented for this database, so that the world-model and ICN methods could be compared.

5.4.5.1 Experiments

The world model was created in two different ways. One version used all 12 speakers' training utterances (the first 10 utterances) to generate the speaker-independent model. The other method used only the fixed impostor cohort set of 6 impostors' training utterances. Codebooks of 128 and 32 centres were created for each of the models. ICN was applied using the 5 member cohort described in section 8.3.

5.4.5.2 Results

The results of using the world model, and a comparison with ICN, are presented in Table 5.4. Using all possible speakers, the world-model gives the best average equal error rate results. The size of the codebook did not seem to be too important, with little difference in total error rate between the 128 centre and 32 centre codebooks (3.4% vs. 3.5%). However, using the limited set of 5 speakers is more comparable to ICN where only a subset of the impostors is used to normalise the score. In that case, ICN is better than the world-model approach, both in terms of d' and average equal error rate.

	d'	FR(%)	FA(%)	EER(%)
ICN	4.2	1.7	3.8	3.7
All (128)	4.0	1.1	3.6	3.4
All (32)	4.1	1.7	3.7	3.5
Subset (128)	4.0	2.8	5.6	5.4
Subset (32)	4.0	2.8	4.7	4.5

Table 5.4: Results of using the world model normalisation.

5.4.5.3 Discussion

The world-model normalisation proved to be a successful means of normalising scores, with a d' of around 4.0 compared to 3.6 for the unnormalised system. This decrease in error rates has also been found by Matsui and Furui (1995), who got an average reduction in equal error rates from 4.3% to 1.8% for text-independent tests using a 60 speaker database.

The advantage of the world-model approach over ICN is the fact that the test utterance need only be presented to the world model rather than to several impostor models. The main disadvantage is that it is not as flexible as impostor cohort normalisation. Impostor cohort normalisation may be speaker-specific, but a world model approach uses the same model for each speaker. The impostor cohort may also be varied as speakers are added or removed from the database. While a world model may also be updated (Matsui and Furui, 1995), the effect of adding a new speaker may be less pronounced than for ICN, where a speaker may be removed from the cohort to include the new speaker.

5.5 Discussion of the normalisation results

The *model vs. model* ranking proved to be a novel way of ranking impostors *a priori*, without having to test each speaker model with the impostor training utterances. It was then successfully incorporated in impostor cohort normalisation, which increased the separation between the genuine speaker and impostor cohort score distributions.

Subtracting the mean of the impostor cohort (SMIC) gave reasonable separation between the genuine and impostor score distributions, but had one or two limitations. When the cohort was derived from the lowest scores it was biased

by the model of the speaker who said the utterance. It also required testing an utterance against all the impostor models. Choosing a different set of impostors for each speaker, based on their impostor rankings, gave better separation than the unnormalised case – but the scores returned were not normalised to a fixed range like ICN scores.

The ICN method worked well in increasing the separation between the genuine and impostor distributions. An impostor cohort of approximately 15 impostors gave good results, without having to test against too many impostors. It also kept the identification error rate close to that of the unnormalised results. The other advantage of this normalisation technique was that only a subset of the impostors was required for testing. This is in contrast to SMIC (using the best scoring impostors) and SLIMS which require an utterance to be presented to all possible impostors. Of the 3 selection methods investigated, using the nearest *model vs. model* ranked impostors gave the most consistent results for d'.

The SLIMS method gave the best results for the d' separation measure, but suffers from the fact that it only works for a closed set of speakers. It requires a test utterance to be presented to all possible impostors and this could be time-consuming if there was a large database. Also the method would not work as well if the impostor came from outside the database. So, although the ICN doesn't give as great a separation, it has the benefit of being open-set.

The RBF network gave better results than the VQ and normalised VQ systems for all conditions. However, overall the normalised VQ system performed well against the discriminating power of the classifying neural network, particularly when the temporal variation training set was used.

The world model gave similar results to ICN. Its main drawback is that it isn't speaker-dependent and thus may serve some speakers better than others. For this reason it was decided to use the speaker-dependent ICN approach for the remaining experiments.

The normalisation chosen for the subsequent experiments was ICN with a cohort of 15 speakers. In the tests just described this had a d' of 4.5 and an identification error rate of 10.3% compared to a d' of 3.8 and an identification error rate of 9.3% for the unnormalised system. The increase in d' indicates that the average equal error rate will be lower, and the score normalisation means that a single threshold for all speakers can be contemplated. This is investigated

further in the more comprehensive experiments of the next chapter.

5.6 Summary

A new means of *a priori* ranking impostors has been put forward, which may be used to select impostors for cohort normalisation. Several normalisation methods have been investigated, of which ICN proved to be the most flexible and consistent. The next chapter gives the results of using this normalisation method for more comprehensive text-dependent and text-independent experiments.

CHAPTER 6

VECTOR QUANTISATION

6.1 Introduction

The normalisation experiments of the previous chapter only used the word *one*. More comprehensive text-dependent tests were therefore required, the results of which are presented in this chapter. Tests were also carried out to confirm that the temporal variation introduced by the multiple recording sessions plays an important rôle in speaker recognition. Finally, it was decided to investigate if the *model vs. model* impostor rankings and their subsequent use in selecting normalisation cohorts could be usefully applied to text-independent speaker recognition.

The following text-dependent experiments used the same speaker set as before, but included the words *one* to *nine* and *zero*. The text-independent tests were performed using 38 speakers from the TIMIT database. The results obtained for these experiments cover both identification and verification. The identification error rate was calculated for both the unnormalised and ICN conditions, the verification error rate for the ICN condition only.

6.2 Text-dependent VQ experiments

In the following text-dependent experiments, the first two recording sessions were used as training data and the last three as test data. This gave a total of 310 (10×31) training utterances and 465 (15×31) impostor test utterances per speaker. For identification, this meant that there were 465 test utterances. For verification, it meant that in total there were a possible 465 false rejections

	Unnormalised		ICN				
	d'	IE(%)	d'	IE(%)	FR(%)	FA(%)	EER(%)
VQ(8)	3.1	14.4	4.2	15.4	5.0	5.1	5.1
VQ(32)	3.6	7.5	4.6	8.3	3.2	3.3	3.3
VQ(128)	3.8	6.5	4.7	7.0	2.7	2.7	2.7

Table 6.1: Results of varying the codebook size (IE = identification error, FR = false rejection, FA = false acceptance, EER = average equal error rate).

and a possible 13,950 ($15 \times 30 \times 31$) false acceptances per word.

6.2.1 Varying the size of the codebook

One of the most important aspects of vector quantisation is the size of the codebook (Booth *et al.*, 1993; Picone, 1993; Yu *et al.*, 1995). In general, it was found that the larger the codebook the better the results, though the improvements usually became less pronounced as the size of the codebooks increased. In order to quantify our results properly, varying sizes of codebooks were tested to examine their effect on the results. Three sizes of codebook were tried: 8, 32 and 128. These tests are referred to as VQ(8), VQ(32) and VQ(128) respectively.

6.2.1.1 Results

The results for the variation of codebook size are shown in Table 6.1. Increasing the size of the codebook leads to improvements for all error measurements. However, it does suffer from the law of diminishing returns, as the benefit of increasing the codebook size from 32 to 128 is not as marked as that of increasing the codebook size from 8 to 32.

6.2.1.2 Discussion

In the case of varying the size of the codebook, it is clear that increasing the complexity of the system can increase the recognition rates. However, there are limits to how far system complexity can be increased before the greater computational burden outweighs the benefits in recognition rates. This finding is in keeping with the results from Yu *et al.* (1995). Using the same database and

test conditions, they found that a 32-element codebook, though not optimal, was deemed to be sufficient to model a speaker.

It is also possible to reach a stage where there are not enough training data for very large codebooks. *In extremis*, each vector in each training utterance would be a codebook element. Booth *et al.* (1993) found that verification performance degraded when there weren't enough training examples for the larger codebooks. On the basis of these results a 32-element codebook was chosen for all subsequent experiments.

6.2.2 Delta cepstrum

One way of improving recognition performance is the inclusion of more speaker-relevant information. One such feature is the delta cepstrum, which has been found to improve error rates for some speaker recognition systems (Picone, 1993; Rosenberg and Soong, 1987). The second-order delta cepstrum for a frame i is calculated as follows:

$$\delta_i = \frac{c_{i+2} - c_{i-2}}{5} \tag{6.1}$$

where δ_i are the delta cepstral coefficients for frame i, c_{i+2} are the cepstral coefficients for frame $i+2$ and c_{i-2} are the cepstral coefficients for frame $i-2$. The delta cepstrum vectors were combined with cepstral vectors to form vectors of 24 elements.

6.2.2.1 Results

A comparison of the results for a 32-element codebook with and without delta cepstral coefficients is shown in Table 6.2. The benefit of using the delta cepstrum is that the identification error is decreased. Without normalisation, the d' is actually less than that for the baseline system (4.3 vs. 3.6). However, as ICN relies on the identification error rate to normalise the scores, the ICN error rates are lower for the delta cepstrum than the basic system (2.1% vs. 3.3%). In fact, the ICN verification error using the delta cepstrum is better than any codebook created without use of the delta cepstrum.

	Unnormalised		ICN				
	d'	IE(%)	d'	IE(%)	FR(%)	FA(%)	EER(%)
VQ(32)	3.6	7.5	4.6	8.3	3.2	3.3	3.3
VQ(delta)	3.4	4.4	5.1	5.0	2.1	2.1	2.1

Table 6.2: Results of incorporating the delta cepstrum (IE = identification error, FR = false rejection, FA = false acceptance, EER = average equal error rate).

6.2.2.2 Discussion

In keeping with the results of Rosenberg and Soong (1987), the addition of the delta cepstral coefficients proved to be an effective way of improving the results, in particular the identification error rate. This information proved to be useful in discriminating between speakers, without increasing the computational burden as much as the 128-element codebook. The overall size of the codebook was only twice that of the original 32-element codebook, compared to four times as large for the 128-element codebook. The delta cepstrum also outperformed the 128-element codebook, giving a better increase in performance for a lower increase in computation.

6.2.3 Temporal variation

One of the most important aspects of this database is the fact that the recordings were made over 5 sessions. As the speaker models were constructed from the first two sessions, they contain less intra-speaker information variation than if recordings from all 5 sessions were used.

In order to confirm the importance of temporal variation and to quantify it for this database, new models were created that used the first 2 repetitions from each of the 5 sessions to incorporate temporal variation into the model. In the previous experiments the speaker model had no information about the speaker's voice in the final 3 sessions. This test used a 32-element codebook and is referred to as VQ(TV).

6.2.3.1 Results

The results for the codebooks generated using utterances from all recording sessions are presented in Table 6.3. The final total error rate of 1.0% is less than

	Unnormalised		ICN				
	d'	IE(%)	d'	IE(%)	FR(%)	FA(%)	EER(%)
VQ(32)	3.6	7.5	4.6	8.3	3.2	3.3	3.3
VQ(TV)	4.2	1.8	5.3	2.0	1.0	1.0	1.0

Table 6.3: Results of incorporating utterances from each recording session (IE = identification error, FR = false rejection, FA = false acceptance, EER = average equal error rate).

a third of the original rate of 3.3% and less than half that using delta cepstral coefficients (2.1%).

6.2.3.2 Discussion

The importance of temporal variation is made clear by the results, which show that the inclusion of utterances from all recording sessions dramatically reduces the error rates. The d' increases from 3.6 to 4.2 and the average EER falls from 3.3% to 1.0%. This trend has been found by other researchers when training data is taken from more than one recording session. Furui (1986) found that if training utterances were recorded over as long a period as 10 months, then error rates could be decreased by 50%, and that 95% accuracy could be obtained, even if the interval between training and test speech is more than three months. Yu *et al.* (1995), using the Millar database, noted a pronounced drop in identification error rates as soon as data from the second recording session were added to a training set which consisted of utterances from the first recording session only.

This indicates that a major reason for the less than perfect performance is that the information in the first two recording sessions encompasses less intra-speaker variation than the utterances which are taken from each of the recording sessions. This kind of problem is extremely difficult to overcome, though Rosenberg and Soong (1987) used a somewhat idealised means of updating speaker models, which reduced the error rate from 7.5% to 4.9% for single-digit text-dependent tests.

6.2.4 Using more than one word

We saw in Chapter 5 that a speaker's impostor rankings varied from word to word. Therefore, it seemed likely that combining the results from several words

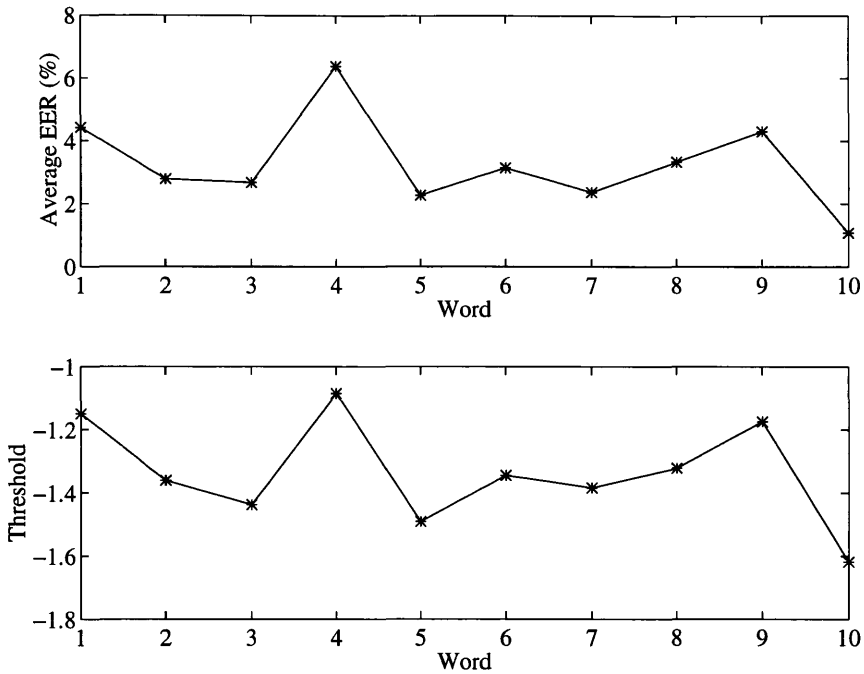

Figure 6.1: Relationship between ICN threshold and overall verification error.

together would improve the performance because any single impostor would find it hard to impersonate the genuine speaker across several words.

Furthermore, examination of the word error rates shows that some words are significantly better for recognition purposes than others. This is shown in the upper graph of Figure 6.1 where the word *zero* (word index 10) has the lowest error rate and *four* has the worst.

6.2.4.1 Results

Results were obtained by summing the scores of different utterances from the different words together. In Figure 6.2 the difference between using the words in a random order and using the best words first is highlighted. By starting with the best words first (which has an error rate of 0% after 2 words), the error rate can be reduced significantly faster than by using the words in a random order (which has an error rate of 0% after 4 words).

6.2.4.2 Discussion

The results show the benefit of using more than one word. By adding the words scores together, the identification error rate drops dramatically. This

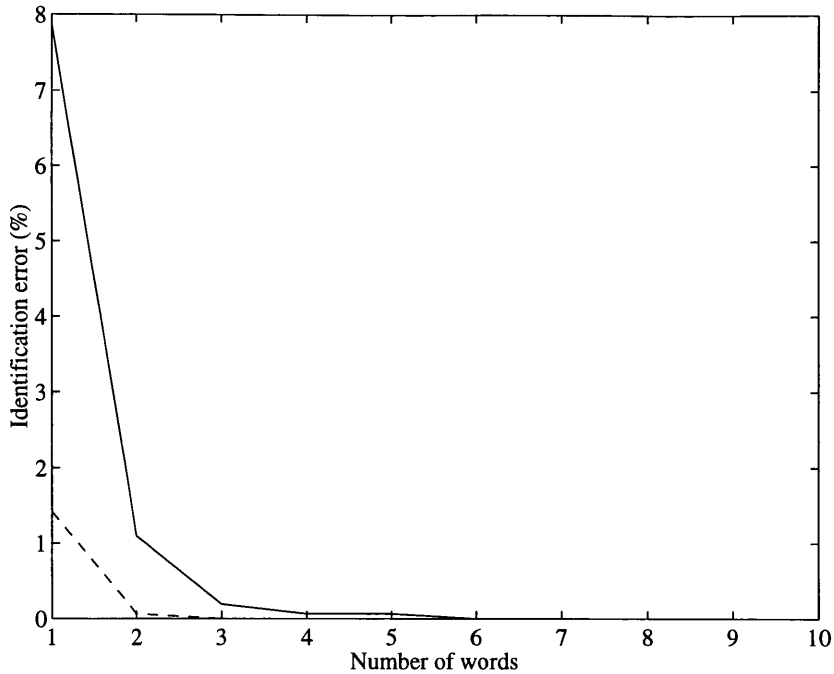

Figure 6.2: The average identification error rate when using more than one word ('–' = using randomly selected word order, '– –' = using the best words in order).

improvement is even more pronounced if the best words are used first, with 0% identification error rate if 3 words are used. This improvement in performance would be mirrored in the ICN verification error rate, as it is dependent on the identification error rate. Similar results for using more than one word were found by Hannah (1997) and Rosenberg and Soong (1987), where the error rate decreased from 7.5% for single-digit tests to 0.8% for seven-digit tests. Carey and Parris (1992) found that concatentating two five-digit strings together reduced the equal error rate from 1.4% to 0.4%, while Booth *et al.* (1993) got a reduction in the equal error rate from 17.3% for single-utterance tests to 3.3% for ten-utterance tests.

6.2.5 Thresholding

Using ICN, the results have been normalised in terms of the number of standard deviations away from the mean score. As the normalised scores now lie within the same range, a single speaker-independent threshold may be set for each word, which meant setting ten thresholds. The threshold was set by minimising the difference between the false rejection and false acceptance error rates (because of

	ICN		
	FR(%)	FA(%)	EER(%)
VQ(32)	4.3	4.4	4.4
VQ(delta)	2.2	2.2	2.2
VQ(TV)	0.7	0.8	0.8

Table 6.4: Results for the word *one* using a single threshold for all speakers (FR = false rejection, FA = false acceptance, EER = average equal error rate).

quantisation it was not always possible to make the error rates equal). Figure 6.1 shows the relationship between threshold and overall verification error. The higher the error rate, the higher the threshold, as the poorer models generate genuine speaker scores that are further from the model.

Speaker-specific thresholds were set for the word *one* to determine whether there was a significant difference between using a single threshold for a word or setting an individual threshold for each speaker.

6.2.5.1 Results

Tables 6.4 and 6.5 emphasise the difference between setting a speaker-independent threshold for each word and setting an individual threshold for each speaker. Table 6.4 shows the error rates for the word *one* using ICN and a single threshold for all speakers. Table 6.5 shows the effect of setting a separate threshold for each speaker. The results make two things clear. The first is that setting a single common threshold is worse than setting an individual threshold for each speaker. Taking VQ(32) as an example, the average equal error rate for a fixed threshold is 4.4% compared to 1.8% for individual thresholds. This trend also occurs for the delta cepstrum and temporal variation results. The second is that even when a separate threshold is set for each speaker, ICN still improves the error rate over the unnormalised system. Again, taking VQ(32) as an example, the average equal error rate is 5.4% without normalisation compared to 1.8% using ICN.

6.2.5.2 Discussion

Although using a speaker-independent threshold per word is an attractive proposition, the results are markedly worse than setting individual thresholds for

	Unnormalised			ICN		
	FR(%)	FA(%)	EER(%)	FR(%)	FA(%)	EER(%)
VQ(32)	4.1	5.4	5.4	1.7	2.7	1.8
VQ(delta)	3.0	4.5	4.4	0.4	1.4	1.3
VQ(TV)	0.4	1.9	1.9	0.0	0.4	0.4

Table 6.5: Results for the word *one* using an individual threshold for each speaker (FR = false rejection, FA = false acceptance, EER = average equal error rate).

each speaker. In each case, setting a speaker-specific threshold reduced the error rates by nearly 50%. For this reason the average equal error rate was determined using individual thresholds for the experiments in the remaining chapters.

6.2.6 Discussion of the text-dependent results

Each of the above results tells us something important about the speaker recognition problem. They confirm that ICN is a good way of normalising results for verification, as well as confirming the problem of temporal variation as one of the biggest hindrances to success. They also highlight that some words are better for distinguishing between speakers than others. Finally, setting a single threshold per word is not as successful as setting speaker-specific thresholds.

6.3 Text-independent VQ experiments

Although score normalisation had been shown to work for the text-dependent case, it had yet to be tested for the text-independent case. The tests used the TIMIT database (c.f. section 2.6). VQ codebooks of 256 elements were used to model the speakers.

In one set of experiments, the seven SA and SX sentences were used as training sentences and three SI sentences for testing. As there were only three genuine speaker test utterances, a meaningful d' could not be generated for these results. The d' depends on the standard deviation of the genuine speaker scores and three scores is too few for this to be calculated meaningfully. For identification, there were 114 (3×38) test utterances. For verification, there were a possible 114 (3×38) false rejections and a possible 4218 ($3 \times 37 \times 38$) false acceptances.

In a second test, only the five SX sentences were used for training and the SX and SI sentences were used for testing. This meant that there were five genuine speaker test utterances and so a meaningful d' could be calculated. For identification there were 190 (5×38) test utterances. For verification there were a possible 190 (5×38) false rejections and a possible 7030 ($5 \times 37 \times 38$) false acceptances.

6.3.1 Impostor ranking

Although the impostor ranking had worked well in the text-dependent case, there may be problems with the text-independent case. The reason is that in the text-dependent case the ranking of the impostors varied for each word and therefore no speaker consistently impersonated another for every word. As TI inherently uses many words, the impostors may vary so much from word to word, that the difference in ranking may not be as pronounced. However, tests showed that the ranking of the impostors did correlate well with the results of the tests. An example of the ranking is shown for Speaker 6 in Figure 6.3. The *model vs. model* scores are represented by '$*$' and the *test vs. model* scores by 'o'. One of the utterances for Speaker 21 appears to score very well against Speaker 6's model. In fact, this utterance scored well against all models and was found to be badly endpointed with almost half the utterance consisting of silence. This would seem to indicate that an utterance with a lot of silence may be prone to impersonate other speakers more than a well-endpointed utterance.

The rankings were tested for two cases. In the first case, the SA and SX sentences were used to create the model and the SI to test it. In the second instance, only the SX sentences were used to create the model and the SA and SI sentences to test it. The average $|r_m - r_r|$ value, calculated using only SI test utterances, was around 4 places. It was around 5 places for the SA+SI tests, indicating the loss of information in the model when only the SX sentences are used for training. (Random rankings had an average difference of around 12 places.) This is very similar to the average ranking difference of around 4 for the test utterances in the text-dependent case. Because the text-independent data were recorded in a single session, one might expect the average ranking difference to be smaller than for the text-dependent BT Millar data, as the latter are subject to inter-session variability. However, the text-independent models

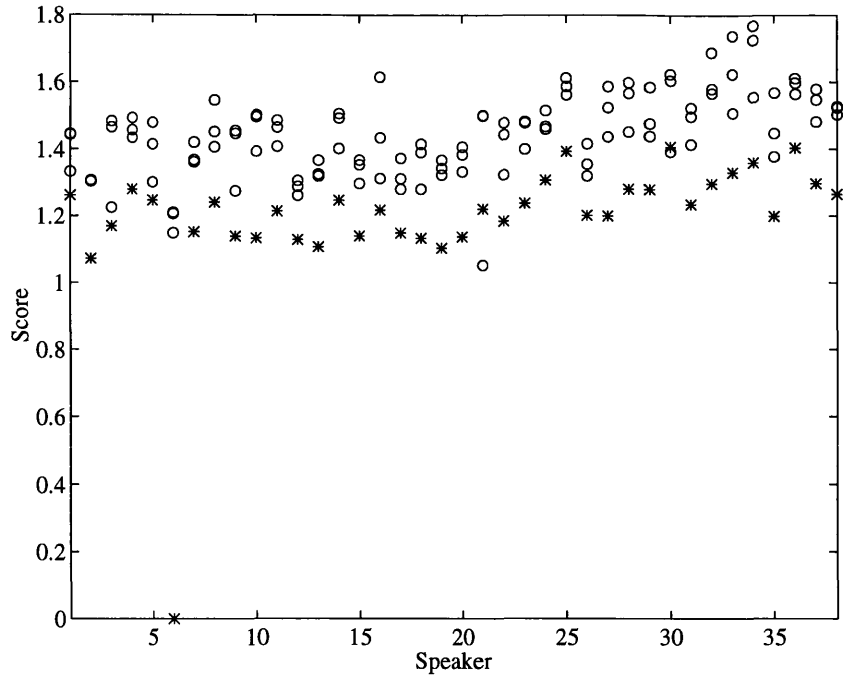

Figure 6.3: Utterance ('o') and *model vs. model* ('∗') scores for Speaker 6.

	Average ranking difference	
	SI	SA+SI
VQ(256)	4.2	5.2

Table 6.6: Average difference in impostor ranking for the text-independent models.

are much more general than the text-dependent models, where only a limited number of phonemes and contexts are being considered. These two competing factors appear to balance one another.

As the ranking was successful, it was used to generate the impostor cohort for ICN as in the text-dependent case (c.f. section 5.3).

6.3.2 ICN and delta cepstrum

The first experiment tested whether the delta-cepstrum, which decreased the error rates for the text-dependent case, is also of benefit in the text-independent case. As the sentences used for training contain more frames than the text-dependent words, larger codebooks are required to model each speaker. A codebook of 256 elements was found to be large enough. Then delta cepstral

	Unnormalised	ICN			
	IE(%)	IE(%)	FR(%)	FA(%)	EER(%)
VQ(256)	0.0	0.0	0.0	0.1	0.1
VQ(delta)	0.0	0.0	0.0	0.2	0.2

Table 6.7: Results of incorporating the delta cepstrum for the text-independent tests (IE = identification error, FR = false rejection, FA = false acceptance, EER = average equal error rate).

coefficients were added to the cepstral vectors and a new set of codebooks generated (c.f. section 6.2.2).

6.3.2.1 Results

The results for the cepstral vectors both with and without the delta cepstral coefficients are given in Table 6.7. Without normalisation, both cases have zero identification error by the end of the sentences. However, the results for the impostor cohort normalisation indicate that using the delta cepstral coefficients is not beneficial for text-independent speaker verification. In the delta cepstrum case the false acceptance rate is three times that of the ordinary system, though both error rates are very low (0.2% and 0.1% respectively).

6.3.2.2 Discussion

The text-independent VQ results indicated that ICN could be used successfully for the text-independent situation. This was because the speaker rankings, based on the *model vs. model* tests, worked for text-independent as well as text-dependent tests. The VQ+ICN system has an identification error of 0.0% at 3 seconds, which is better than an identification error of 0.6% for a classifying RBF system trained using the same speaker set (Fredrickson and Tarassenko, 1995).

The use of delta cepstral coefficients did not improve the text-independent results. This is not in keeping with the results of Rosenberg and Soong (1987), who found that combining LPCC and delta-cepstra led to a drop in identification error rates from 18.0% to 7.5%. In contrast to our experiments, the test utterance (a digit) was part of the original training set. However, Thévenaz and Hügli (1995), in a series of experiments using different training and test sets, also found that the delta cepstrum did not improve the results. The most probable reason is

that the delta cepstrum acts as a context enhancer (i.e. it depends on the vectors that come before and after the frame in question) and as phonemes may find themselves in different contexts for text-independence tasks, the delta cepstrum was less useful than in the text-dependent case, where the context of a phoneme does not change.

6.3.3 Variation of the training set

In order to test the effect of constraining the test utterances presented to the text-independent model (i.e. making every speaker use the same phrase, rather than each speaker saying something different), models were created using only the SX sentences, and the SA and SI sentences were used for testing. This meant that the text-independent models would be tested using both text-dependent (the SA sentences were the same for every speaker) and text-independent sentences (the SI sentences varied from speaker to speaker). Testing with the SA sentences reduces the variation in utterance content normally associated with testing text-independent models.

6.3.3.1 Results

The results for using only the SX sentences for training are presented in Table 6.8. They show that the reduction in training information increases the error rates compared to using both SA and SX sentences for training (VQ(256)). However, of the two test sets, the SA sentences had fewer errors, with an identification error of 0.0% and an average equal error rate of 0.4% compared to 0.9% and 0.9% for the SI sentences. A graph showing the decay in total identification error rate as the length of the test utterance is increased is given in Figure 6.4.

6.3.3.2 Discussion

Though the error rates increased when the SA sentences were moved from the training to the test sets, the text-dependent SA sentences got lower error rates than the text-independent SI sentences.

	Unnormalised		ICN				
	d'	IE(%)	d'	IE(%)	FR(%)	FA(%)	EER(%)
VQ(256)		0.0		0.0	0.0	0.1	0.1
SA		0.0		0.0	0.0	0.4	0.4
SI		0.9		4.4	0.9	0.9	0.9
SA+SI	4.3	0.5	5.9	2.6	0.5	0.5	0.5

Table 6.8: Results of using only the SX sentences for training and the SA and SI sentences for testing (IE = identification error, FR = false rejection, FA = false acceptance, EER = average equal error rate).

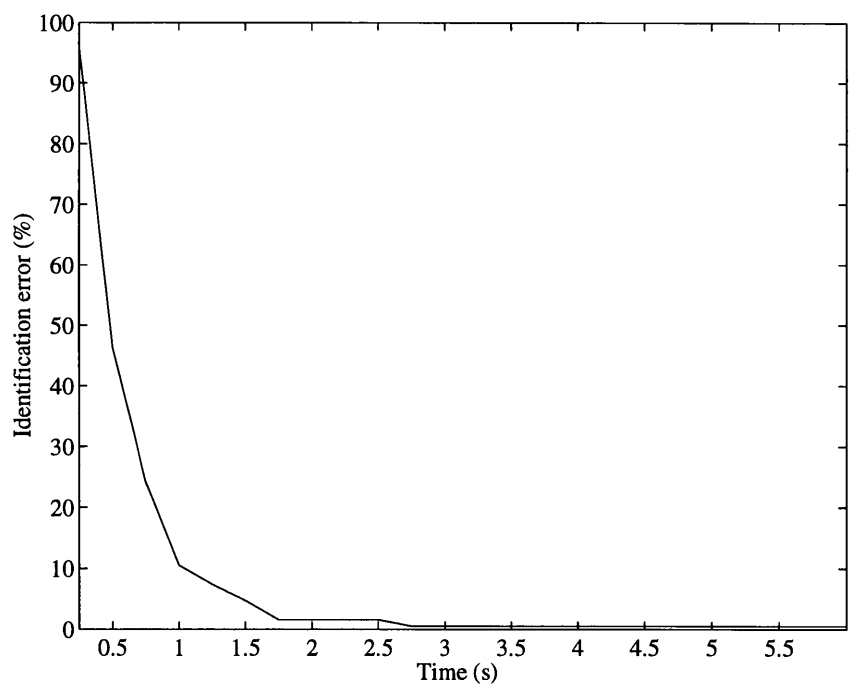

Figure 6.4: The average identification error rate for the text-independent experiments using both SA and SI sentences for testing.

6.3.4 Discussion of the text-independent results

The experiments have shown that *model vs. model* ranking is applicable to the text-independent case. They have also shown that ICN improves on the unnormalised results and that the delta cepstrum does not have the same benefit as it does in the text-dependent case. Finally, constraining the input to the text-independent speaker models gives lower error rates than text-independent testing.

6.4 Summary

The VQ tests, both text-dependent and text-independent, have served to show the usefulness of ICN in lowering verification error rates and have also created a base-line system against which further work may be judged. Now distance models using predictive neural networks and impostor cohort normalisation are investigated in the next chapter.

Chapter 7

Predictive neural networks

7.1 Introduction

Having found a suitable score normalisation method in Chapter 5, and proved its usefulness for both text-dependent and text-independent tests in Chapter 6, neural network distance models may now be investigated. This chapter covers both text-dependent and text-independent experiments using predictive neural networks, as well as methods of rejecting frames with low discriminating power.

Predictive neural networks (PNN) may be used to create speaker models without the use of discriminative training (Ambikairajah, Keane, Kelly, Kilamartin and Tatterstall, 1993; Artières and Gallinari, 1993; Hattori, 1992). They use only genuine speaker information to create a non-linear model of the genuine speaker's speaking characteristics. Hence, addition of new speakers to the database doesn't require any retraining of the networks, which is a possibility with classifying neural networks. Most predictive neural networks work on the basis of predicting the next frame in a sequence of the previous frames (Artières and Gallinari, 1994). Using back-propagation or a similar algorithm, the networks are trained to minimise the difference between the predicted frame and the actual frame. The non-linearity of the networks is very important in capturing the speaker characteristics (Hattori, 1994). However, as the predictive neural networks also encode the speech as well as the speaker information, there is a lot of redundant information in the speaker model.

7.2 Text-dependent experiments

Predictive neural networks have been used for text-dependent speaker recognition (Ambikairajah *et al.*, 1993), though they are more commonly used for the text-independent case. The experiments reported here used the 31 male speakers from the BT Millar database saying the word *one* (c.f. section 2.5), which was used for the normalisation experiments in Chapter 5.

The basic network consisted of 2 input frames and 1 output frame. This was based on the model used by Hattori (1994) for his predictive neural network experiments. As each frame consisted of 12 cepstral coefficients, there were 24 input and 12 output parameters. The number of hidden nodes was varied as part of the experiments.

7.2.1 Network size

The number of nodes in the hidden layer was varied while the numbers of inputs and outputs were kept constant. If there are too few hidden nodes the data may not be modelled properly, but if there are too many hidden nodes then the network may overfit the data and reduce its ability to generalise.

7.2.1.1 Results

The results of varying the size of the hidden layer are shown in Figure 7.1. d' increases initially as the number of nodes increases, but begins to drop again as too many nodes are used. The identification error follows a similar pattern, with the highest error rates for 1 and 6 hidden nodes.

7.2.1.2 Discussion

The results fitted the expected pattern, with too few nodes being inadequate to model the data, and too many nodes overfitting the data. On balance, 3 or 4 hidden nodes gave the best results.

7.2.2 Error rates

To quantify the error rates of the PNN, a predictive neural network with 3 hidden nodes was compared to the standard 32-element VQ system of the last chapter

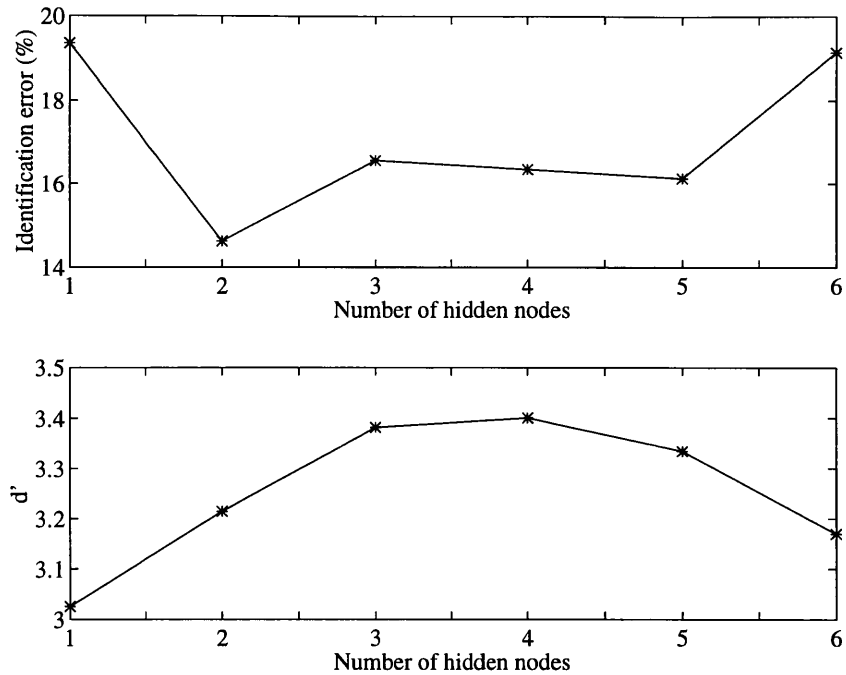

Figure 7.1: Results of varying the number of hidden nodes for the text-dependent case.

	d'	IE(%)	FR(%)	FA(%)	EER(%)
PNN	3.4	16.3	6.5	8.1	8.1
VQ(32)	3.8	9.3	4.1	5.4	5.4

Table 7.1: Results for the text-dependent predictive neural networks.

for the 31 speakers saying the word *one*. The results were evaluated for d', the identification error and the average equal error rate using no normalisation and individual thresholds.

7.2.2.1 Results

The results using PNN with 3 hidden nodes compared to the standard VQ system are given in Table 7.1. d' for the PNN is lower than that for the VQ (3.4 vs. 3.8) and the average EER is much higher (8.1% vs. 5.4%). However, a weakness of the PNN results is the identification error rate, which at 16.3% is nearly twice that of the VQ system (9.3%).

7.2.2.2 Discussion

The high identification error of the predictive neural network means that using the impostor cohort normalisation, which depends on good identification results, would be less successful for the PNN system than for the VQ system. So although the network has achieved reasonable separation between genuine speaker and impostor score distributions, with a d' of 3.4, it is not amenable to improvement through impostor normalisation because of its poor identification error rate.

7.2.3 Discussion of the text-dependent results

With an identification error rate nearly twice that of the VQ system, the PNN system implemented would require some modification to be useful for text-dependent speaker recognition. The high identification error rate means that score normalisation based on impostor cohorts would not be as successful as for the VQ case. This isn't to say that the PNN's didn't learn some speaker-specific characteristics, as shown by a d' of 3.4 compared to that of 3.8 for the VQ system.

7.3 Text-independent experiments

These experiments used the SX sentences for training and the SA and SI sentences for testing, as used by Artières and Gallinari (1993). The basic network consisted of 2 input frames and 1 output frame. This structure was also used by Artières and Gallinari (1993) for predictive neural network experiments using the same database. As each frame consisted of 12 cepstral coefficients there were 24 input and 12 output parameters. The number of hidden nodes was varied as part of the experiments.

7.3.1 Network size

Similar to the previous text-dependent experiment, the number of nodes in the hidden layer was varied to determine how it affected the success of the network. A balance has to be struck between having enough nodes to model the speaker information and having too many nodes, leading to overfitting of the data which reduces the ability of the network to generalise (Hattori, 1993). The number of inputs and outputs was kept constant.

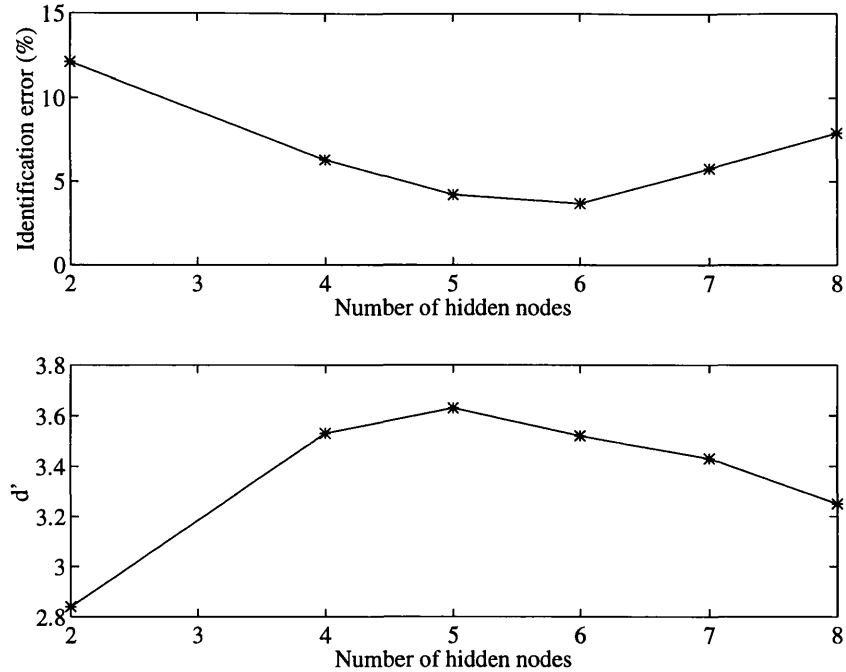

Figure 7.2: Results of varying the number of hidden nodes for the text-independent case (using both SA and SI test sentences).

7.3.1.1 Results

The results for varying the size of the hidden layer (Figure 7.2) show that 6 hidden nodes gives the best results in this case. Anything else leads to poorer generalisation. d' is similar for 4, 5, 6 and 7 hidden nodes, though the identification error varied more markedly.

The percentage identification error for the SA and SI sentences is shown in Table 7.2. The SA sentences consistently had a lower error rate than the SI sentences. The SA sentences are the same for all speakers and, therefore, are comparable to a text-dependent test of the text-independent speaker model.

7.3.1.2 Discussion

Again we have seen a pattern similar to that for the text-dependent case, with too few and too many nodes producing poorer results. This dependence on the number of hidden nodes is in keeping with the findings of other researchers (Artières, 1995a; Hattori, 1993), but the results cannot be compared as they used different numbers of inputs and outputs. Furthermore, the identification error for the text-dependent SA sentences is lower than for the text-independent

	Identification error (%)	
Nodes	SA	SI
2	7.9	14.9
4	2.6	10.5
5	2.6	5.3
6	0.0	6.1
7	6.6	9.7
8	9.2	13.2

Table 7.2: Variation in identification error for the SA and SI sentences as the number of hidden nodes is increased.

SI sentences, as found in the last chapter (c.f. section 6.3.3). This finding was not reported by Artières and Gallinari (1993) who tested with both SA and SI sentences. Hattori (1992) also trained with SX sentences, but tested with only SI sentences.

7.3.2 Frame rejection

Many researchers have tried to reject frames that don't have a high discrimination factor (Artières and Gallinari, 1993; Li and Porter, 1988). These are frames that are not very distinctive of the genuine speaker and could easily be mistaken as belonging to an impostor. Artières and Gallinari (1993) found that using highly distinctive frames, picked by hand, it is possible to identify speakers using as few as 10 frames. However, picking such frames automatically has proved to be very difficult. Two methods, based on the frame scores for all speaker models, were considered.

In the first method, the frames are ranked with respect to the standard deviation of the scores across all speaker models for each frame. Li and Porter (1988) found that the genuine speaker's score was more likely to be the lowest for frames that had a large standard deviation of scores. Thus, only using frames that had a large standard deviation in their scores should give better results. A gradually increasing percentage of the frames with the largest standard deviation of scores was used to calculate the results.

In the second method, frame score normalisation (FSN), each speaker's score for a frame was normalised with respect to the standard deviation of that frame's

scores against the other speakers' models. Given a frame i, from a sentence with N frames, the frame normalised score is calculated as follows:

$$s^i_{FSN} = \frac{s^i_{gen} - \mu^i_{imp}}{\sigma^i_{imp}}$$

where s^i_{FSN} is the normalised score for frame i, s^i_{gen} is the original genuine speaker model score for frame i and μ^i_{imp} the mean and σ^i_{imp} the standard deviation of the impostor scores for frame i. Using all possible frames, the final score for a sentence with N frames is:

$$S_{FSN} = \frac{1}{N}\sum_{i=1}^{N} s^i_{gen}$$

where S_{FSN} is the final score and s^i_{gen} is the frame normalised score for frame i.

These normalised scores emphasise when a speaker's score for that frame is considerably lower or higher than the overall distribution of scores. As the genuine speaker model should create some of the lowest errors, this normalisation should emphasise the genuine speaker scores. The percentage of best normalised scores for each speaker was varied from 10% to 100%.

7.3.2.1 Results

Figure 7.3 shows how the identification error and d' vary with the percentage of large standard deviations used. It is clear that the use of frame scores on the basis of their standard deviations leads to only minor variation in the d' separation of the genuine and impostor speaker distributions. There is, however, considerable variation in the identification error. Using only the top 10% of the largest standard deviation frames, there was an error rate of 34.3%. Only when 60% or more of the frames are used does the error rate return to the levels associated with no frame rejection.

The reason why using the largest standard deviation frames does not improve the results is shown in Figures 7.4 and 7.5. Figure 7.4 shows a good example of how the genuine speaker scores tend to be toward the bottom of the larger standard deviations, as found by Li and Porter (1988). However, as the poor example in Figure 7.5 shows, this phenomenon does not always occur. Here, the genuine speaker scores fail to fall at the lower edge of the larger distributions. This indicates that the genuine speaker model is not modelling the speaker's voice as well as the impostor models.

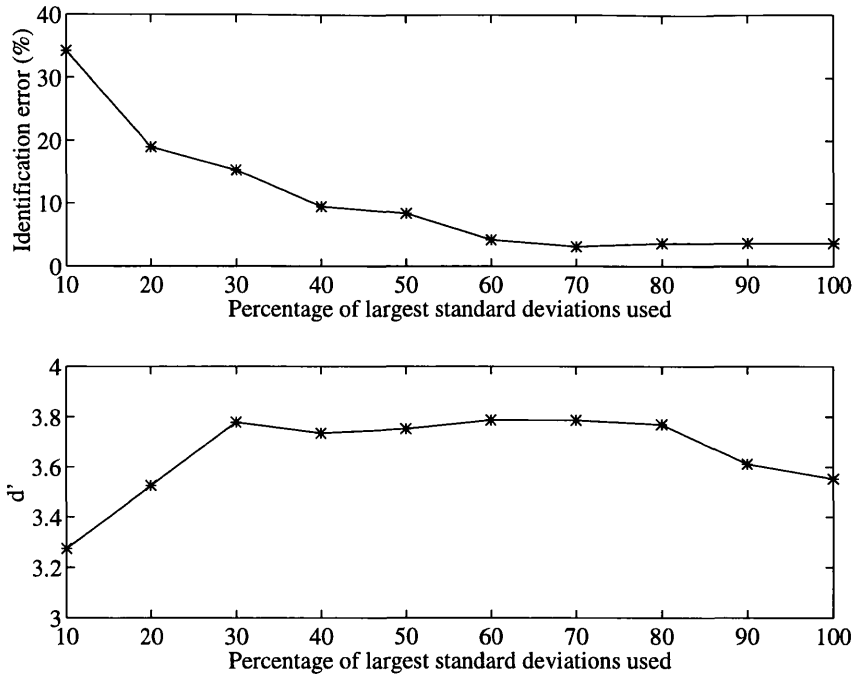

Figure 7.3: Results of varying the percentage of frames with large score standard deviations used to generate scores.

Figure 7.6 shows how using the best frame normalised scores for each speaker affects the results. When 50% or more of the frame normalised scores are used the identification error rate is around that of the basic system. For certain percentages, the identification error rate is actually less than that for the unnormalised system. However, the d' for these scores is much higher (4.5) (Figure 7.6) than for the basic system (3.5) (Figure 7.2). So using the frame normalised scores helps to separate the genuine and impostor distributions.

7.3.2.2 Discussion

Of the two methods, frame score normalisation produced better results than rejecting frames on the basis of the standard deviation of the frame scores. However, there is little to be gained from rejecting frames on the basis of their frame normalised scores, as using 100% of the frames gives similar results to using 40%. So FSN may be used as a score normaliser rather than as a means of rejecting less discriminant frames.

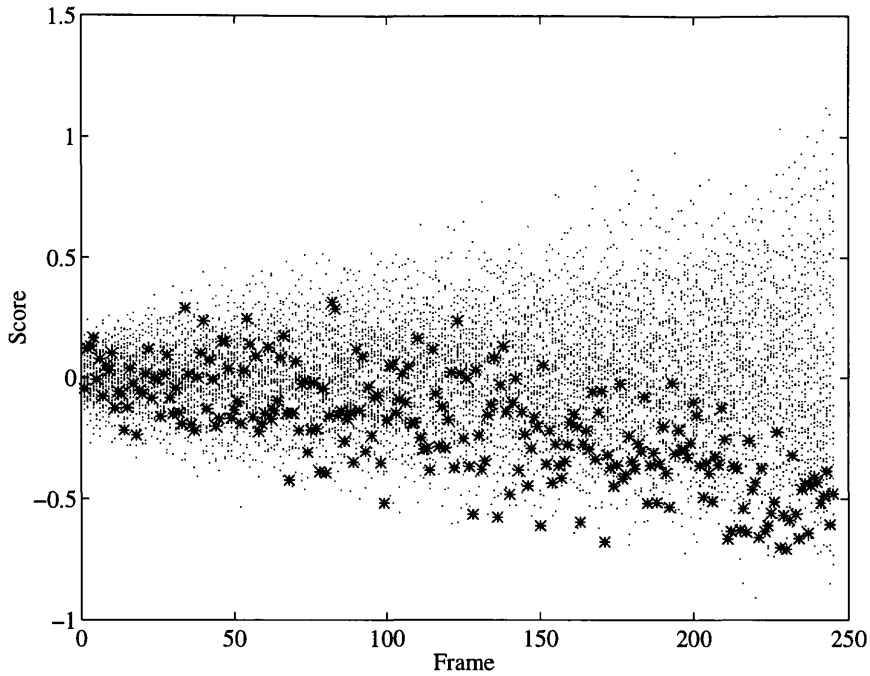

Figure 7.4: An example of a sentence where the genuine speaker scores fall at the lower edges of the larger standard deviations ('$*$' = genuine speaker scores and '$\cdot$' = impostor scores).

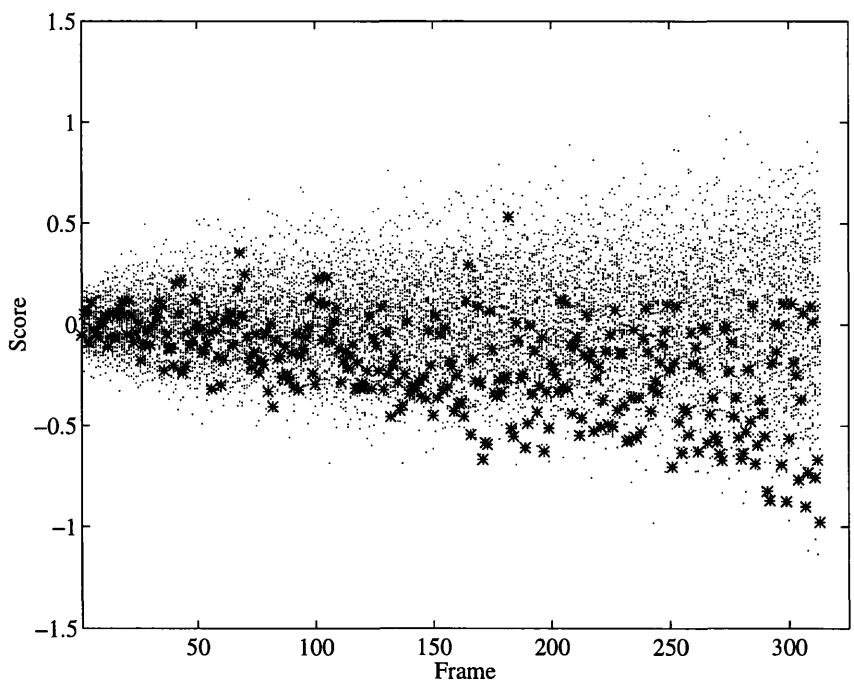

Figure 7.5: An example of a sentence where many of the genuine speaker scores do not fall at the lower edges of the larger standard deviations ('$*$' = genuine speaker scores and '$\cdot$' = impostor scores).

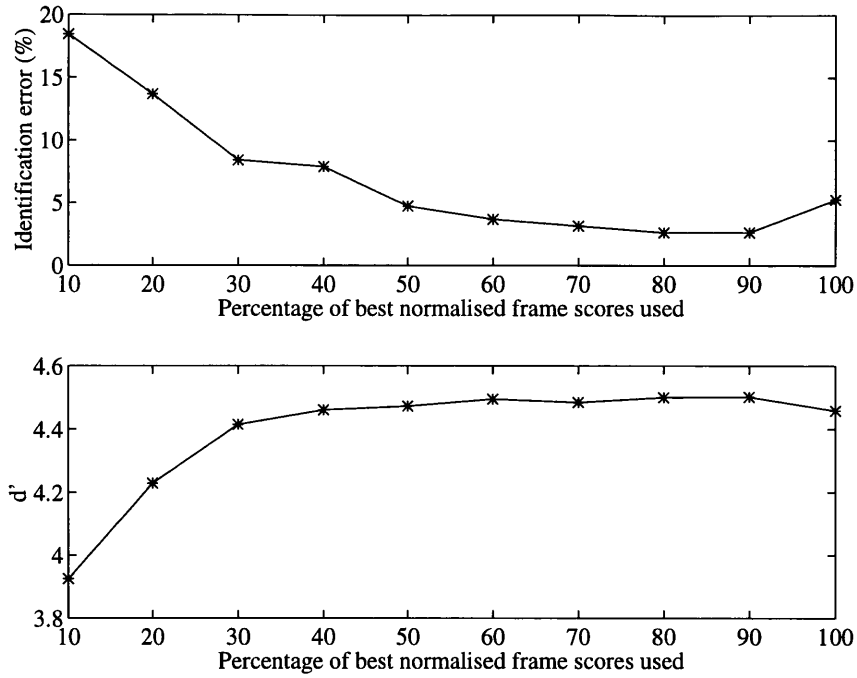

Figure 7.6: Results of varying the percentage of best normalised frame scores used to generate scores.

	Average ranking difference
VQ,PNN	4.7
VQ,VQM	5.2
PNN,VQM	6.5

Table 7.3: Average difference in impostor position between the PNN, VQ and VQ *model vs. model* (VQM) rankings.

7.3.3 Impostor cohort normalisation

The impostor cohort normalisation method (c.f. section 5.4.2) was also investigated. Two methods of selecting the impostors for the cohort were investigated. The first used randomly selected impostors from the speaker population. The second used impostors chosen on the basis of the impostor rankings using VQ codebooks (c.f. section 6.3.1). The differences between the PNN, the VQ and the VQ *model vs. model* (VQM) rankings are given in Table 7.3. The difference of 6.5 indicates that there is some correlation between the PNN and VQM rankings, though not as much as between the VQM and VQ rankings (5.2).

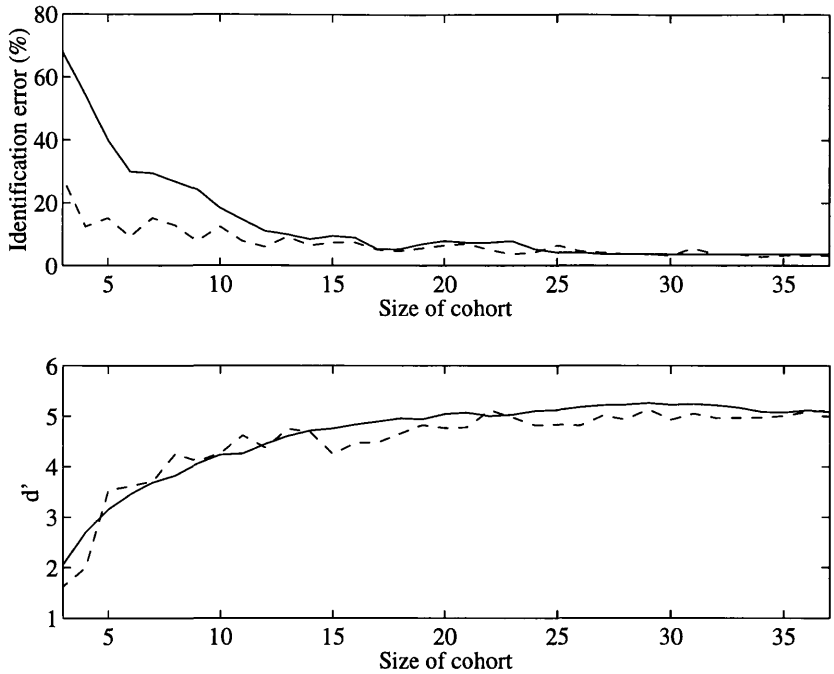

Figure 7.7: Results of varying the number of impostors in the cohort ('–' = cohort based on VQ impostor rankings, '- -' = cohort based on random selection from the impostors).

Using ICN to normalise predictive neural network scores is in contrast to the work done by Hattori (1994). In that case, rather than test an utterance against several impostors' models, a single model of the impostor population was created. The test utterance was then presented to both the genuine speaker model and the generalised impostor model. The score from the generalised impostor model was then used to normalise the genuine speaker model's score.

7.3.3.1 Results

The results of using the randomly-selected impostor cohort and the VQ selected cohort are shown in Figure 7.7. The impostor cohort picked on the basis of the impostor rankings from a VQ test proved to be more consistent than the random selection of impostors. The d' generated using ICN is larger than that achieved using FSN. An impostor cohort of 15 speakers, based on the VQ impostor rankings, was used for the subsequent tests.

The results in Table 7.4 compare the average error rates using the various methods. It is clear that both ICN and FSN are an improvement on the basic system. ICN (1.5%) appears to be slightly better than FSN (2.0%), while a

	d'	FR(%)	FA(%)	EER(%)
PNN	3.5	1.1	6.0	6.0
FSN	4.5	0.5	2.0	2.0
ICN	4.8	0.0	1.6	1.5
FSN+ICN	4.8	0.0	1.4	1.4
VQ(32)+ICN	5.9	0.0	0.6	0.6

Table 7.4: Results for the PNN using normalisations (PNN = unnormalised system, FSN = frame score normalisation, ICN = impostor cohort normalisation, FR = false rejection and FA = false acceptance, EER = average equal error rate).

combination of the two (1.4%) doesn't give any major advantage over ICN on its own.

7.3.4 Discussion of the text-independent results

The success of predictive neural networks for speaker recognition is highly dependent on the number of hidden nodes used. This may be a limitation, as it means a system may not be developed without preliminary testing to optimise the network architecture.

The fact that the SA sentences get lower error rates than the SI sentences indicates that using text-dependent tests (in that all speakers say the same test phrase) on a text-independent speaker model gives better results than pure text-independent tests. One reason for this may be that the training data model the SA sentences better than the SI sentences. Also, impostor models may model certain phonemes better then the genuine speaker model, if that phoneme is more prominent in the impostor training set than the genuine speaker training set.

Rejecting frames on the basis of the standard deviation of the frame scores was not beneficial. However, normalisation of the frame scores with respect to the scores for other models did lead to a considerable improvement in d' and the average EER.

The tests using ICN showed that VQ speaker rankings can be used to form impostor cohorts for PNN systems. The ICN method itself also worked well for PNNs. Both ICN and FSN seemed to perform a similar function. Using both together was only marginally better than using either method alone.

Although the basic PNN system implemented here can benefit from normalisation, it is not as good as a VQ system (Finan, Sapeluk and Damper, 1997b). However, multi-state ergodic PNN models allow more comprehensive modelling of the speaker. For a subset of 20 female speakers from the TIMIT database, Hattori (1994) got identification error rates of 4.3% for a 1-state model and 0% for a 4-state ergodic model using complete sentences. Further improvements may be achieved by training the multi-state neural networks using Viterbi training (Rabiner, 1989). Artières and Gallinari (1995b) achieved recognition rates of over 95% for 1.5 seconds of speech tested on 3-state ergodic models trained using the Viterbi algorithm. The basic normalised VQ system of the last chapter had an error rate of 4.7% for 1.5 seconds (see Figure 6.4).

As both the VQ and PNN systems were using the same feature set and achieving similar results, it was felt that a better speaker recognition system might be developed through enhancing the feature set being presented rather than changing the recognition technique. This depenence on the cepstrum (whether LPCC or MFCC) as the predominant feature set in speaker recognition has also been questioned by Furui (1997). He proposes that the inclusion of other features may be required in order to improve the performance of current speaker recognition systems. To this end the next chapter investigates the use of the LP residual as a feature set, both on its own and in combination with the LPCC-based recognition system.

7.4 Summary

This chapter has investigated the use of predictive neural networks for text-dependent and text-independent speaker recognition. Although frame rejection and score normalisation helped to improve the results, the final results were poorer than those of the VQ system. The next chapter investigates whether better performance might be achieved by looking at a different feature set, rather than the recognition method.

Chapter 8

Linear prediction residual

8.1 Introduction

In Chapter 8 we saw that, whether a VQ or multi-state ergodic PNN system is used, the results using the LPCC feature set are similar. This raised the question of whether the LPCC feature set, rather than the recognition system, may be the limiting factor in lowering the error rates. Although cepstral coefficients (be they linear prediction or mel-frequency) are one of the best speaker recognition feature sets, it may be that better performance may be achieved through enhancing the feature set by providing complementary information.

Further evidence that the cepstrum may be limited in its ability to represent speaker-dependent characteristics is more subjective. When studying the results of previous experiments, it was clear that utterances the cepstral-based representation found to be similar were not necessarily similar to the human ear. Quite often no similarity could be heard between the impostor and genuine speaker utterances, though similarities were quite clear for the genuine speaker's own utterances. This would seem to imply that the cepstral feature set fails to detect acoustic cues that the human ear does.

In his review of speaker recognition technology, Furui (1997) askes 16 questions that are central to advancing the field of speaker recognition. Several of the questions concern the dependence of recognition systems on the cepstrum and whether other features could be found to enhance recognition rates. One feature that may complement the LPCC is the linear prediction residual, as it corresponds to that element of the speech waveform not modelled by the LP-derived cepstral coefficients. This chapter looks at the speaker-dependent information in the

LP residual and whether it can be used to enhance the information already present in the LP cepstrum.

8.2 LP residual

One of the most obvious sources of speaker-dependent information, other than the LP cepstral coefficients, comes from linear predictive analysis itself. For a given frame of speech, linear predictive analysis produces filter coefficients, a gain factor and an error signal (Atal and Hanauer, 1971). The LP coefficients represent a filter whose frequency response models that of the vocal and nasal tracts. The gain is a measure of the energy of the frame. The error signal is the difference between the predicted signal and the actual signal. As LP analysis is supposed to separate the excitation source and the frequency response of the vocal and nasal tracts, the error signal is a model of the excitation signal. While the linear predictive coefficients are used to generate the cepstral coefficients, the gain and error signal are most often ignored.

The excitation signal does have speaker-dependent properties. Most often, it is modelled as the pitch or fundamental frequency f_o, i.e. the rate of excitation of the vocal cords, though it can be problematic to extract it reliably (Furui, 1994; Matsui and Furui, 1990). On its own, pitch is a poorer speaker recognition feature than the traditional LPCC (Atal, 1976; Dubreucq and Vloeberghs, 1994), though when used in combination with the LPCC it can lead to reductions in the error rates (Dubreucq and Vloeberghs, 1994).

However, there is reason to believe that the actual excitation waveform itself, rather than just the frequency of excitation, may contain speaker-dependent information. Thévenaz and Hügli (1995) have investigated using the LP residual as a speaker recognition feature, both alone and combined with LP cepstral coefficients. The LP residual was modelled using the real cepstrum (the inverse FFT of the log-spectrum), which for voiced speech has a peak marking the period of the excitation frequency. In a series of text-independent verification tests, it was found that combining the results of the LPCC and the LP residual recognisers together gave lower error rates than either the LPCC on its own or the combination of results from different LPCC-based recognisers. This suggests that the LP residual does have information complementary to the LP cepstral

coefficients.

8.3 Restricted database

Because of the overheads associated with the following experiments and those of the next chapter, it was necessary to create a restricted database. This consisted of 12 speakers saying the word *seven*. The word *seven* was chosen because it contains voiced, mixed and unvoiced excitation. Although considerably smaller than the original population of 31 speakers, the 12 speakers chosen were hand-picked, allowing the creation of a database with many of the worst speakers and some good speakers. A list of the speakers used for this reduced database may be found in Appendix B.

The purpose of selecting the worst speakers was to see if any of the new approaches improved their performance, while the good speakers were included to make sure that it didn't make their performance any worse. In speaker identification tests, Thompson and Mason (1994) found that the 10% 'poorest' performing speakers accounted for almost one third of the total errors for the system. Thus improving the results for the poorest speakers should improve the overall results.

The shift in emphasis to focus on the poorer speakers is clear in the comparison of the individual d's and average equal error rates shown in Figure 8.1, and the overall results for the two databases, given in Table 8.1. The deterioration in d' and average EER reflect the relative increase in problem speakers. The decrease in identification error is to be expected as the restricted database has less than half the number of speakers of the full database, and identification error increases with population size (Doddington, 1985).

When impostor cohort normalisation was used in the following experiments, there was a slight variation from the method used in the previous chapters. Rather than using a separate cohort for each speaker, a fixed cohort was used for all speakers. This was only possible because of the small size of the database and the fact that the same speakers kept turning up as the best impostors. The 6 most common impostors were selected to form the cohort. Then the best 5 impostors were selected for a speaker's cohort unless that speaker was a member of the cohort, in which case the sixth impostor was used, to make sure the speaker

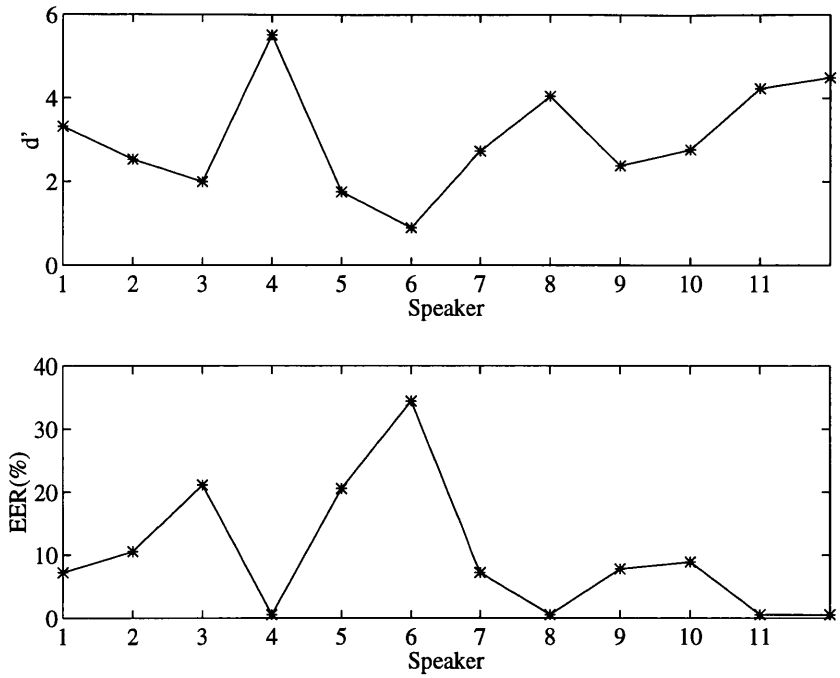

Figure 8.1: The d' and equal error rate (EER) for each speaker in the smaller database.

	d'	IE(%)	FR(%)	FA(%)	EER(%)
31 speakers	3.9	4.7	5.0	6.0	5.9
12 speakers	3.1	3.3	8.9	10.1	10.0

Table 8.1: Comparison of the results for the 31 speakers used for previous experiments and the restricted database of 12 speakers for the word *seven* (IE = identification error, FR = false rejection and FA = false acceptance, EER = average equal error rate).

wasn't part of their own cohort. This meant that the normalisation was closer to the world-model approach as half the speakers had the same impostor cohort, while the cohorts for the six speakers selected as the impostor cohort only differed by one speaker.

8.4 LP residual-based recognition

The difference between the signal from the linear predictor and the actual signal is known as the error signal or LP residual. On the basis of the LP model, the residual represents the excitation signal. In most speech processing applications, this residual is ignored. However, attempts have recently been made to use it in

speaker recognition (Thévenaz and Hügli, 1995).

Examples of the residual for unvoiced and voiced speech are shown in Figures 8.2 and 8.3. (The tapering of the speech towards the beginning and end of the frames is due to the Hamming window used in the autocorrelation LP analysis.) There is no visible structure in the unvoiced residual, but the voiced residual has peaks at the fundamental frequency. In an ideal world, these peaks may be used to determine the speaker's pitch, though in practice the peaks are rarely as clearly defined as in the example given.

It was thought worthwhile to use the residual because it makes up the missing component of LP analysis. Using both the LPCC and the residual would mean using all the speech information and, in theory, getting a more exact model of the speaker. The residual was only modelled for voiced speech, as information about the excitation signal may be gained from that, whereas unvoiced speech is inherently too noisy to extract speaker-dependent information.

8.4.1 Experiments

Only the residual of voiced frames, which were determined using an algorithm based on Cheng and O'Shaughnessy (1989), were used in the experiments. The residual was analysed in four ways. For each of the methods, a 32-element VQ codebook was used to model the resulting feature vectors. The 12 speakers of the restricted database were used for the tests.

First the FFT was taken of the residual of each voiced frame. This gives a measure of the spectral energy of the residual, and the harmonics of the fundamental frequency were clearly visible in the FFT (see the lower diagram of Figure 8.4). The next method used the LPCC of the residual. This was done as the residual spectrum was not completely flat, and there still appeared to be some peaks and troughs in the spectrum which could be modelled using linear prediction. The LP coefficients were then used to generate the cepstral coefficients. An example of this is given in Figure 8.4, which shows a typical voiced residual, its log spectrum and the smoothed spectrum generated by the impulse response of the LP filter. The peaks and troughs, though not as deep as those of a voiced speech, are still modelled by the LP analysis.

The third method used the real cepstrum of the residual, which is generated by taking the inverse Fourier transform of the log-spectrum (Rabiner and

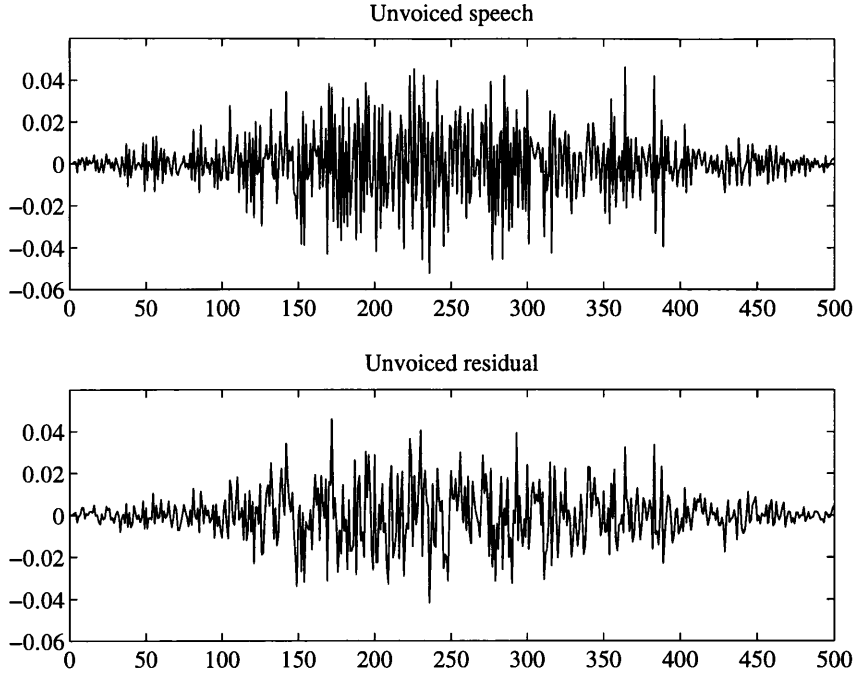

Figure 8.2: LP residual for an unvoiced frame.

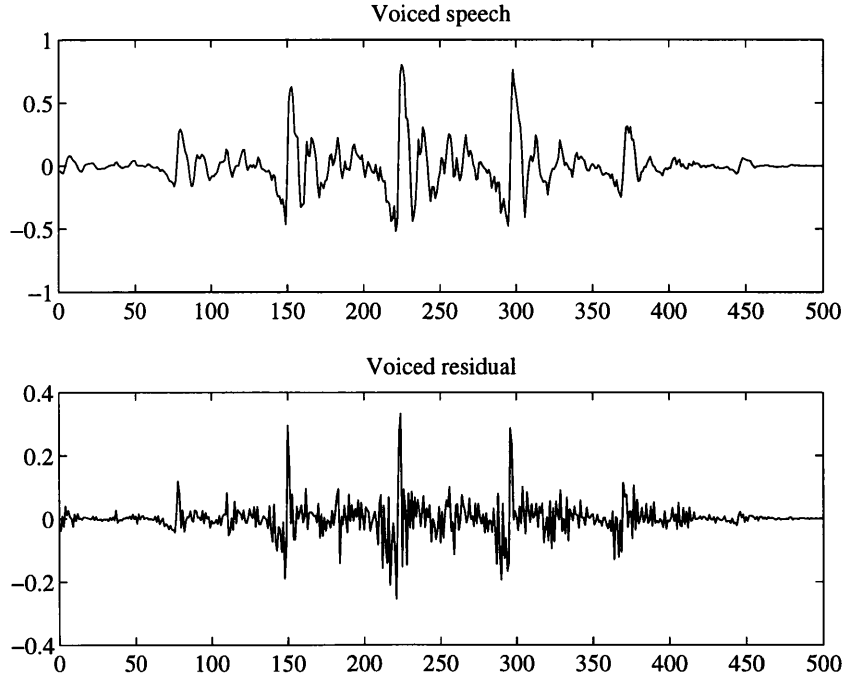

Figure 8.3: LP residual for a voiced frame.

	d'	IE(%)
FFT	0.5	53.3
LPCC	1.0	30.0
RCEP	0.1	91.7
PDSS	0.9	43.9

Table 8.2: Results for the LP residual.

Schafer, 1978). This is similar to the method used by Thévenaz and Hügli (1995), where the fundamental frequency appears as a peak in the real cepstrum. An example of this is given in the upper diagram of Figure 8.5, where the peak at 0.014 in the quefrency domain represents a pitch of around 71 Hz.

Finally a measure called the *power difference of spectra in sub-bands* (PDSS) was used. This is based on the paper by Hayakawa, Takeda and Itakura (1997). Although it claims to use the harmonic structure of the residual, it appears to rely on energy. The log-spectrum of the residual is split into evenly distributed bands, and the PDSS for a given band is calculated by subtracting the ratio of the geometric mean to the arithmetic mean from 1. An example is given in the lower diagram of Figure 8.5.

8.4.2 Results

The results for the 4 methods are presented in Table 8.2. The LPCC of the residual gives the best combination of d' (1.0) and identification error (30.0%). The PDSS d' of 0.9 is close to that of the LPCC, but the identification error of 43.9% is much higher.

8.4.3 Discussion

None of the methods came close to the results obtained using an LPCC-based recogniser. However, feature sets don't have to be of equal singular ability in order to be complementary to each other. Therefore, despite its poor results, the LP residual may yet improve overall error rates when combined with the LPCC. As the LPCC of the residual had the best d' and lowest identification error of the results, it was decided to try combining it with the LPCC results.

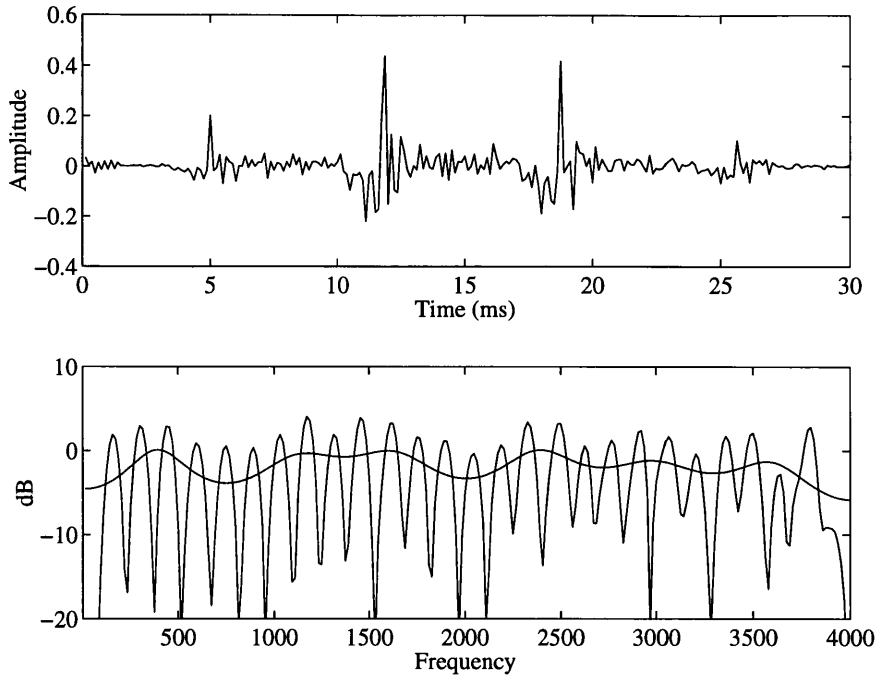

Figure 8.4: An example of an LP residual for a voiced frame, its log-spectrum and the smoothed spectrum of the LP analysis.

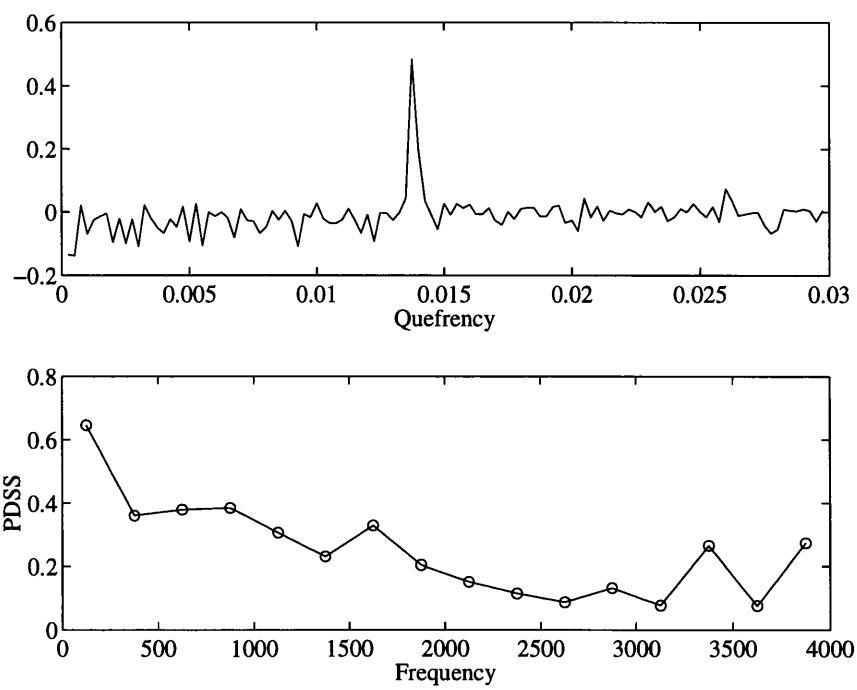

Figure 8.5: The real cepstrum and PDSS representations for the same voiced residual.

	Unnormalised					ICN		
	d'	IE	FR	FA	Total	FR	FA	Total
LPCC	3.1	3.3	8.9	10.1	10.0	1.7	3.8	3.7
Residue LPCC	1.0	30.3	33.3	35.5	35.3	11.1	13.6	13.4
Combined	2.9	2.2	12.2	13.2	13.2	1.7	3.2	3.1

Table 8.3: Results of combining the LPCC and residual together.

8.5 Combined LPCC and LP residual

To see if the LPCC of the residual contained complementary information for the LPCC, the results of the two systems were combined by adding them together.

8.5.1 Experiment

As mentioned above, the experiment consisted of adding the scores from the LPCC and LPCC residual recognisers together. The raw scores were used without any weighting to emphasise one set of scores over the other. As the LPCC system had an average score of 1.4 and the residual system one of 0.4, the unweighted case was biased toward the LPCC results.

8.5.2 Results

The results of combining the LPCC and the residual LPCC together are presented in Table 8.3. The results are very similar to those for the LPCC on its own. However, the combination does lead to a drop in the identification error rate from 3.3% to 2.2% (in effect a drop from 6 to 4 misclassified utterances for this 12 speaker database). This benefit of lowering the identification error was clear when ICN was used to normalise the results, leading to a better ICN average equal error rate (3.1%) compared to the LPCC system (3.7%). This was also the case for the delta cepstrum, where the average equal error rate was poorer before normalisation, but was better after normalisation because of its better identification error rate.

8.5.3 Discussion

There would appear to be some complementary information to be found in the LP residual, as evidenced by the decrease in identification error rate. This improvement is in keeping with the findings of Thévenaz and Hügli (1995), where combining the LPCC and the LP residual led to a reduction from 5.7% to 4.0% for the overall verification error rate. However, the LPCC and the LP residual represent only one way of focusing recognition on different areas of the speech signal. Another approach, known as sub-band processing, has recently been used in automatic speech recognition (Bourlard and Dupont, 1996a). This combines the scores from recognisers dedicated to different frequency ranges rather than the LP residual and the LPCC. This approach is investigated in the following chapters, which describe sub-band processing and its implementation, and the results of the sub-band processing experiments.

8.6 Summary

The results of combining the LPCC and the LP residual have shown that the LP residual does offer complementary information to the LPCC. This idea of using recognisers which focus on different areas of speech spectrum is dealt with in the following chapters.

CHAPTER 9

SUB-BAND PROCESSING

9.1 Introduction

In the last chapter, we saw that the combination of results from two recognisers (based on the LP cepstral coefficients and the LP residual) gave better results than the LPCC recogiser on its own (the identification error rate dropped from 3.3% to 2.2%). This approach of combining the outputs of two recognition systems, which focus on complementary aspects of the speech signal, may also be applied directly to the speech signal rather than using linear prediction and its residual. A series of filters is used to split the whole frequency band into several sub-bands on which different recognisers are independently applied. The scores from the recognisers are combined at certain speech unit levels (i.e. phoneme or word boundaries) to create a global score and a global decision system. This is known as sub-band processing (Bourlard and Dupont, 1996a; Bourlard, Hermansky and Morgan, 1996b). This chapter covers the novel implementation of sub-band processing used in this work, and describes how it affects the spectrum of the speech signal.

Although similar in concept, the grounds for using this approach come less from the complementary nature of the LP and its residual, than from work on speech perception by humans. Allen (1994) suggests that the decoding of speech signals is based on decisions in narrow frequency bands that are processed independently of each other. The combination of the decisions from these frequency bands is done at "certain levels" so that the global error rate is equal to the product of the "band-limited" error rates within the independent frequency channels. This means that if any of the frequency bands yields a zero (or low)

error rate then the resulting global error rate would also be zero (or very low), almost independently of the error rates of the remaining bands.

There are several engineering reasons why such sub-band processing might be applied to speaker recognition:

- The effect of narrow band noise may be reduced. If noise only affects some of the frequency bands, then the remaining clean sub-bands should supply sufficient information to reach the correct decision (based on the idealised combination of results given above, where, in theory, only one error free sub-band is required for correct recognition).

- Some sub-bands may contain more speaker-specific information than other sub-bands. Weighting these sub-bands to emphasise their contribution should lead to better recognition rates, as sub-bands containing little speaker-specific information would have less influence. In fact, some sub-bands might be better for some speakers than others, so that speaker-specific sub-band weighting may be possible.

- Different recognition strategies might be applied to different sub-bands, i.e. longer analysis frames for low frequency bands and shorter frame lengths for high frequency bands, in order to track different speech events.

These points, coupled with the findings from combining the results of the LPCC and LP residual recognisers in parallel, encouraged the investigation of using multiple sub-bands for automatic speaker recognition. However, there are several practical issues to be resolved before these advantages might be realised:

- The number, size and location of the frequency bands must be optimised for speaker recognition. Sub-bands designed for speech recognition may not be suitable for speaker recognition, and it may be that sub-bands could be created on a speaker-specific basis for speaker recognition.

- For the above to work well, some knowledge of which bands contain the most speaker-dependent information is required. The scores from these bands may then be emphasised to increase the speaker-dependent information.

- The point at which the scores are recombined must also be tested. Depending on whether the tests are text-dependent or text-independent, and the

type of recognition system used, it could be done at the end of a frame, phoneme, syllable, word or sentence.

- It may be that different frequency bands should be treated differently and be combined at a later stage than others.

The review of previous work in the next section covers how other researchers have approached solving these problems and forms the background for our work.

9.2 Previous work

Besacier and Bonastre (1997) applied sub-band processing to text-independent speaker identification. Tests were carried out on all 630 speakers of the TIMIT database, using the SX sentences for training and the SA and SI sentences for testing.

Their sub-band processing approach used filter-bank energies as the basic feature set. Twenty-four channels were created using mel-scale triangular filter-bank coefficients, calculated from the FFT power spectrum using a logarithmic scale. These channels were grouped incrementally in sets of 4 to create 20 sub-bands (Sub-band 1: channels 1–4, Sub-band 2: channels 2–5, etc.). The recognition system used second-order statistical measures (Bimbot and Mathan, 1994) with a 1-nearest neighbour decision rule. The recombination strategy was to compute the arithmetic mean of the separate sub-band distances, which was performed after 3 seconds and 6 seconds of speech.

A block diagram of this approach is given in Figure 9.1. First, the wide-band time-domain speech signal is analysed using the Fourier transform, then the sub-bands are created by grouping the channels together in sets of four. Sub-bands are tested independently and then the score combined using an arithmetic mean.

Besacier and Bonastre found that certain sub-bands contained more speaker-specific information than others: in particular the low-frequency sub-bands below 600 Hz and the high-frequency sub-bands above 2 kHz. The sub-bands between these points contained less speaker-specific information. This helps to explain the poorer performance rates for telephone-quality speech, where these critical sub-bands are absent.

Besacier and Bonastre also found that when the number of sub-bands was varied, using more parameters to model the speaker led to better results. It was

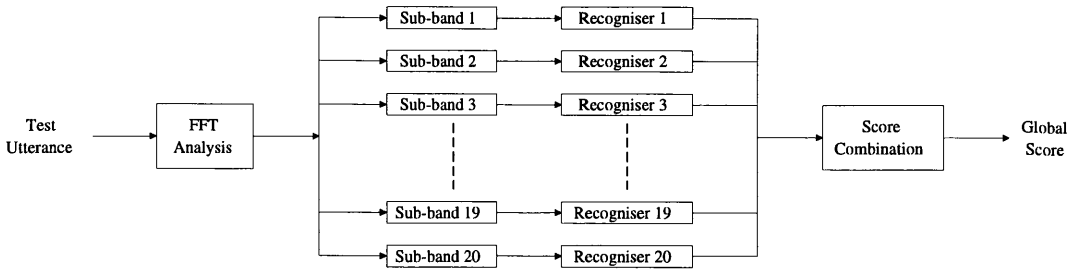

Figure 9.1: Block diagram of an FFT-based sub-band processing system using 20 sub-bands.

also found that the correlation between sub-bands was important when second-order statistical measures are used, and it was suggested that this might influence future recombination strategies.

However, our approach to sub-band processing, applied for the first time to speaker recognition, owes more to Bourlard and Dupont (1996a) than to Besacier and Bonastre. Bourlard and Dupont investigated sub-band processing for automatic speech recognition using a HMM system. Several parameters of the sub-band processing system were investigated, including the number and location of the sub-bands, the feature set used and the weighting scheme for recombination.

For the speech recognition tests, Bourlard and Dupont found that 4 to 5 sub-bands performed well, but that further investigation was warranted before any firm conclusions could be drawn. With regard to the feature set, it was found that modelling the filter bank outputs in terms of LP cepstral coefficients (with cepstral mean subtraction) was more successful than using critical band energies (0.5% error rate vs. 2.0% using 4 sub-bands).

Combination strategies included using the arithmetic mean of the scores, weighting the relative amount of recognition information in each sub-band, weighting on the basis of the signal-to-noise ratio in each sub-band and using the sub-bands scores as the input layer for an MLP network trained to recognise the words being tested. Of these the MLP gave the best results, though the other weightings performed well compared to the wide-band system.

Although sub-band processing gave better results than wide-band processing when narrow-band noise was present, it gave poorer recognition rates than a wide-band system using J-RASTA noise cancellation (Hermansky and Morgan, 1994) when wide-band noise is present. However, using J-RASTA first, followed by sub-band processing led to lower error rates than for the wide-band system (9.1%

Filter-bank		
Sub-band	Centre Frequency (Hz)	Critical Bandwidth (Hz)
1	83	101
2	176	102
3	280	106
4	396	111
5	526	119
6	671	130
7	833	144
8	1015	164
9	1218	188
10	1446	218
11	1700	254
12	1985	298
13	2303	351
14	2659	415
15	3057	490
16	3502	580

Table 9.1: Centre frequencies and bandwidths for the 16 sub-bands.

vs. 12.1%).

9.3 Implementation of sub-band processing

The system implemented for these experiments had 16 sub-bands positioned according to the mel-scale, which is a non-linear scale based on the human auditory system (Zwicker and Terhardt, 1980). Such a scale was used by Besacier and Bonastre (1997), but not by Bourlard and Dupont (1996a) who used fewer sub-bands. The choice of 16 sub-bands may not be optimal, but is in keeping with the auditory basis of sub-band processing. The centre frequencies and bandwidths of the 16 sub-band filters are given in Table 9.1.

The filters implemented were second-order infinite impulse response (IIR) filters. They were designed using the bilinear transform method described in Owens (1993). The advantage of IIR filters is their ease of design and implementation. The disadvantage of IIR filters, though, is that they have a non-linear phase response. It was thought, however, that this would not have a

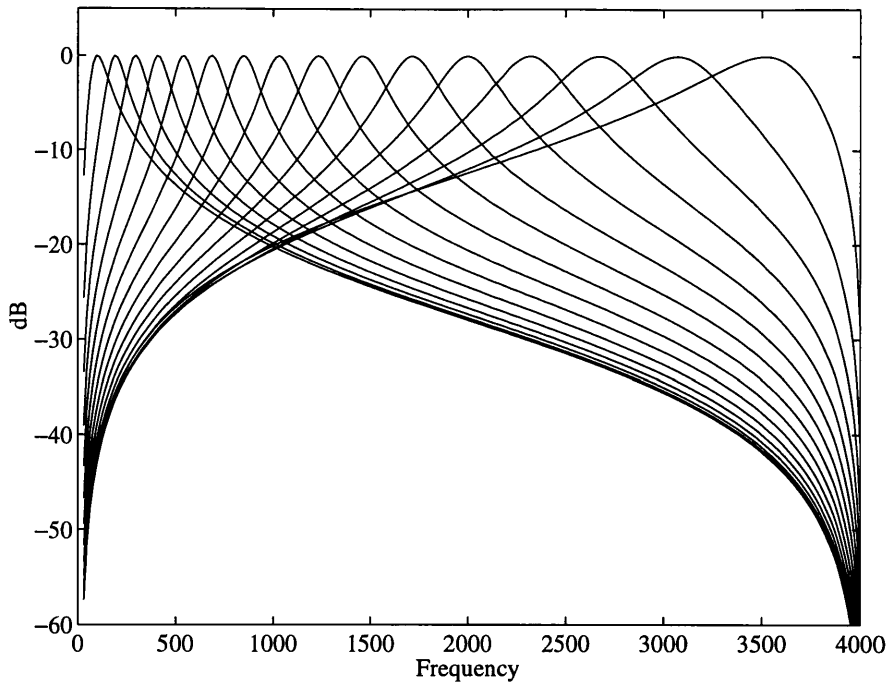

Figure 9.2: Filter characteristics of the filter bank.

significant effect on the results of the final recognition system.

The filter characteristics are depicted in Figure 9.2. As the filters are only second-order, there is considerable overlap between pass-bands. However, the higher the order of the filter, the more poles of the LP analysis that are required to model the filter and not the speech.

The time-domain waveforms created by the filtering were parametrised using linear prediction cepstral coefficients. The standard analysis frame for the experiments (c.f. section 2.5) of 20 ms with a Hamming window and overlapping by 50%, was retained. The LPCC parameters were modelled using vector quantisation codebooks of size 32 (c.f. section 6.2.1). This meant that there were 16 separate recognition systems working in parallel on different frequency ranges.

A block diagram of the system is given in Figure 9.3. Here the first step is is to split the time-domain waveform into 16 band-limited waveforms using the filter-bank. These waveforms are then modelled using LP cepstra and presented to the independent sub-band recognisers. The scores from these recognisers are then combined to give the global score.

The small database of 12 speakers saying the word *seven* (c.f. section 8.3)

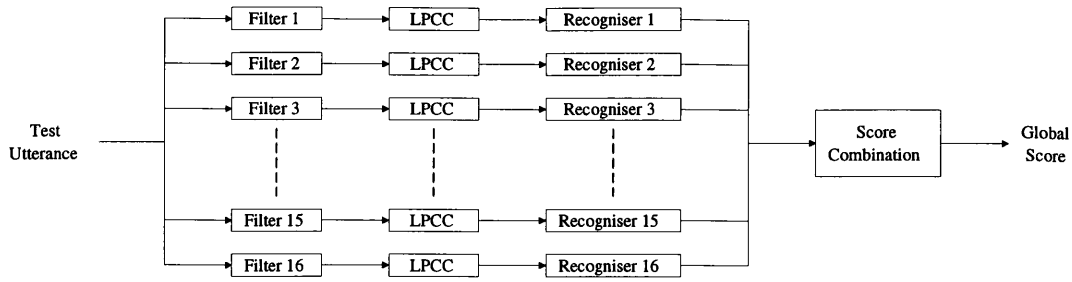

Figure 9.3: Block diagram of the LPCC-based sub-band processing system.

was used for these experiments. In keeping with previous experiments, the first 10 utterances were used for training and the last 15 for testing.

As the database contains quite a few hand-picked problem speakers, any improvement in results is likely to be more pronounced than if the larger set of 31 speakers had been used. There are a lot of good speakers in the larger set who already have zero error rates, and hence, would not benefit from any improvements, making the average improvement smaller than for the problematic speaker set.

The system implemented for the following experiments is quite different from that used by Besacier and Bonastre (1997). Their system used 24 mel-scale triangular-filter bank coefficients to represent each frame of speech. The sub-bands were generated by taking the filter coefficients in groups of 4, giving a total of 20 sub-bands. Our system filtered the speech into separate band-limited waveforms and then analysed each sub-band separately to generate LPCC. Their system used the second-order statistical measures favoured by Bimbot and Mathan (1994). Our system used VQ codebooks to model each sub-band. Finally, they found most of the speaker-dependent information to be above 4 kHz, whereas the database used for these experiments is band-limited to 4 kHz. With regard to the experiments conducted, their tests were text-independent and ours were text-dependent.

9.4 Effect of sub-band processing

This section shows how sub-band processing affects the position of the LP poles and therefore the cepstral representation of the speech for each sub-band.

Figure 9.4 shows a typical frame of voiced speech from the word *seven* (used

in the reduced database). (A similar set of figures to the following may be found for a representative frame of unvoiced speech in Appendix C.) The upper figure shows the 20 ms frame of speech and the lower diagram the FFT (in decibels) of the frame as well as the smoothed log-spectrum, generated from the impulse response of a filter created using the LP coefficients. The smoothed log-spectrum shows four peaks at approximately 600 Hz, 1600 Hz, 2500 Hz and 3500 Hz.

Figure 9.5 shows the LP coefficients generated by each sub-band for the same frame of speech. The x-axis represents the 12 LP coefficients, the y-axis the sub-band (where 1 is the lowest frequency sub-band and 16 the highest-frequency sub-band) and the vertical z-axis the coefficient value. It is clear that the coefficients change as the sub-bands vary in range, though it is hard to determine from using the LP coefficients alone what is happening in terms of the spectra.

Figure 9.6 shows the smoothed log-spectra generated by the impulse responses of filters created using the LP coefficients. In this case the x-axis represents the frequency in hertz, the y-axis the sub-band and the z-axis the spectral magnitude in decibels. Now the results of the variation in LP coefficients are readily visible. The poles of the filter (represented by the peaks in the smoothed spectrum) vary from sub-band to sub-band.

Each sub-band emphasises a different frequency range, thus allowing the poles of the LP filter to focus on particular areas of the spectrum. Starting at the low-frequency end of the spectrum, we can see how the first sub-band has two prominent poles (indicated by the peaks in the spectrum) located below 800 Hz. As the centre frequency of the sub-bands is increased, these two poles are brought closer together, until finally they are modelled as a single pole. As the centre frequency increases further, the influence of this low-frequency pole is reduced.

A similar effect is seen in the other frequency ranges as the sub-bands emphasise or de-emphasise the peaks in the spectra. In particular, sub-bands 9–14 (centre frequencies 1218–2659 Hz) locate many poles in the middle-frequency range which were absent from the low-frequency sub-bands and also from the smoothed spectrum of the wide-band analysis in Figure 9.4.

It is clear from the figure that emphasising different frequency ranges using sub-banding causes significant pole migration across the bands. The effect of this on the cepstrum is demonstrated in Figure 9.7, where the LP cepstral coefficients for each sub-band are shown. This variation in the make-up of each sub-band's

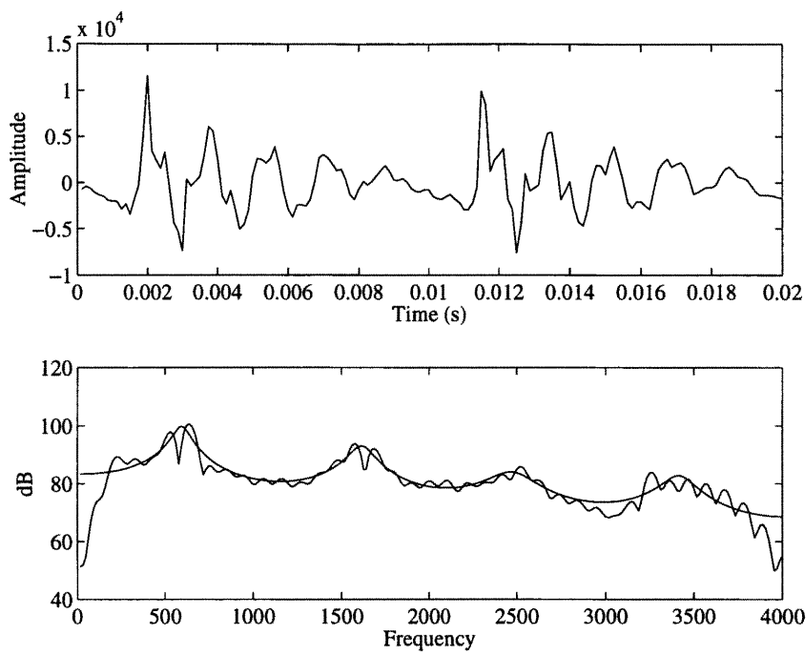

Figure 9.4: Representative frame of voiced speech, its FFT and the smoothed spectrum of the LP analysis.

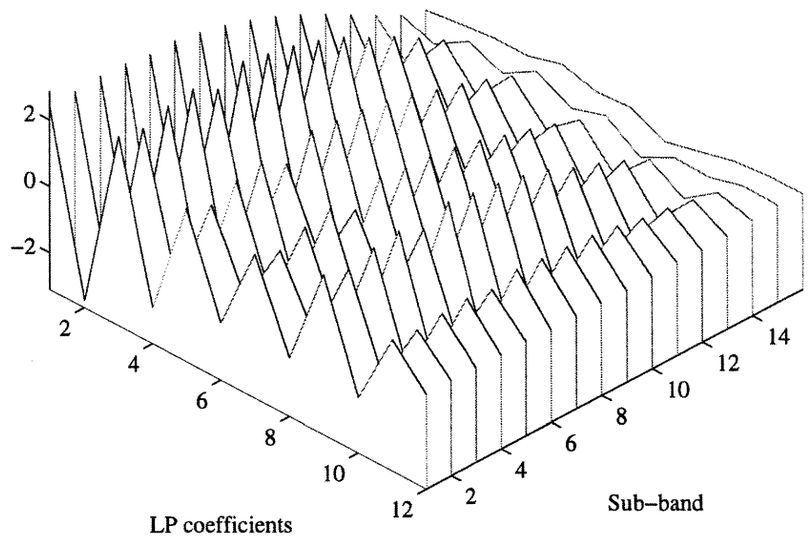

Figure 9.5: LP coefficients for the voiced frame of speech.

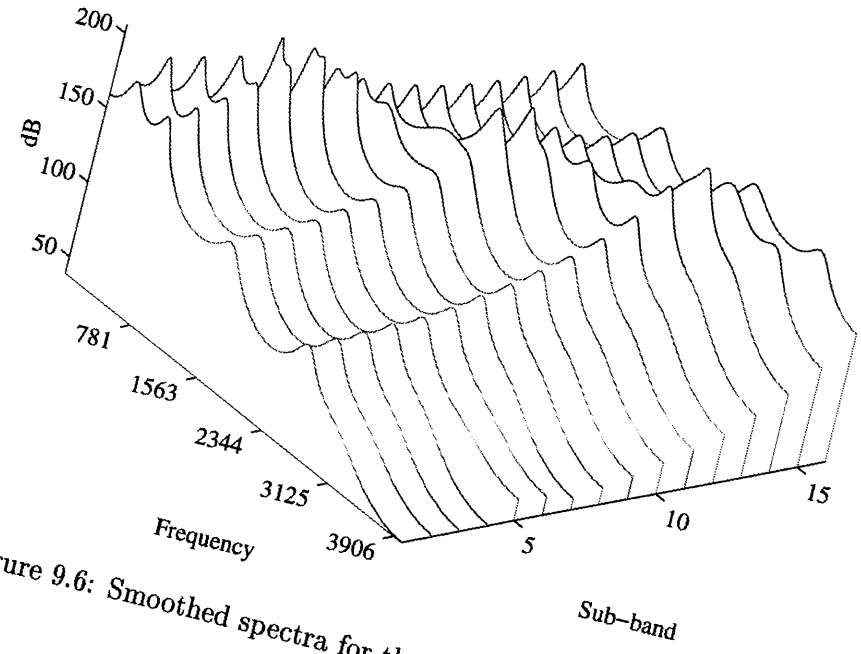

Figure 9.6: Smoothed spectra for the voiced frame of speech.

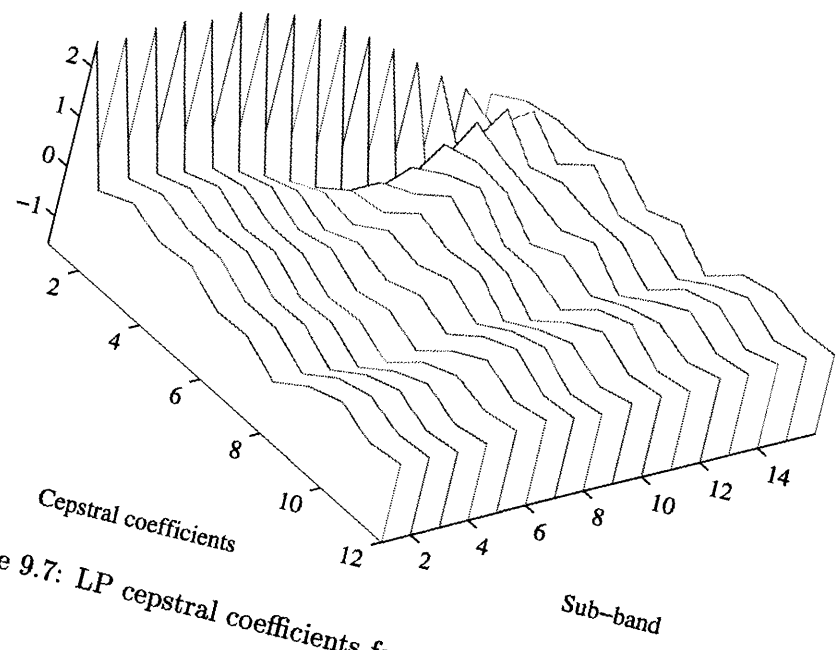

Figure 9.7: LP cepstral coefficients for the voiced frame of speech.

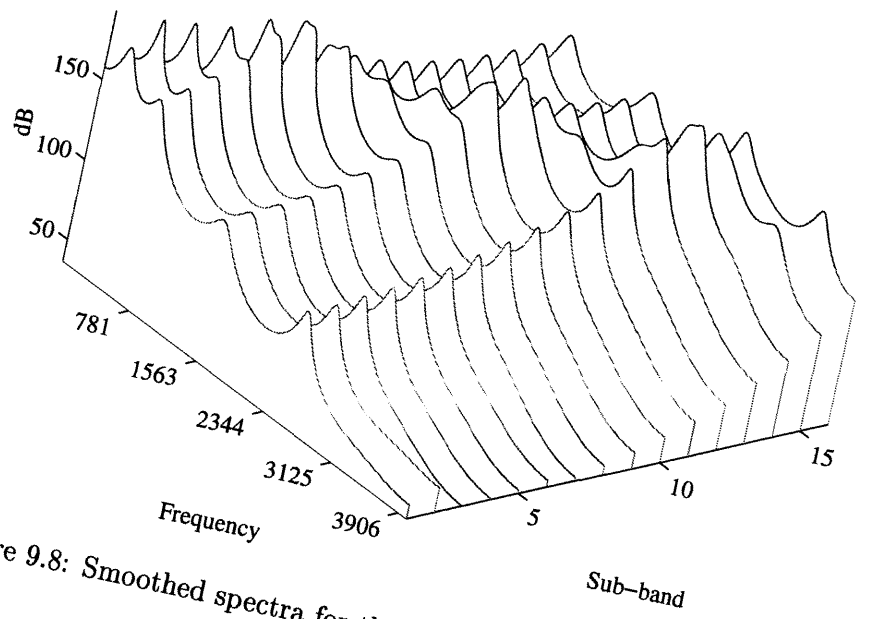

Figure 9.8: Smoothed spectra for the next frame of voiced speech.

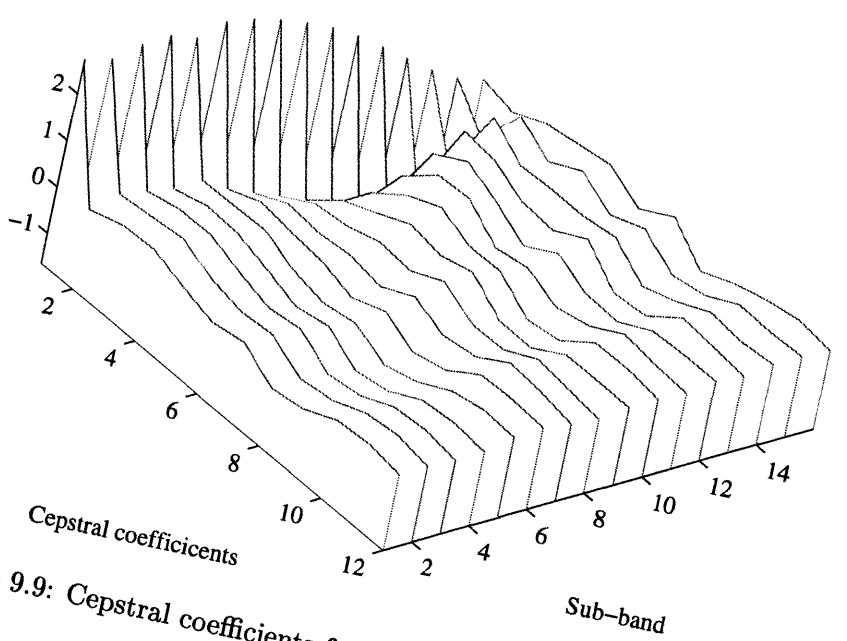

Figure 9.9: Cepstral coefficients for the next frame of voiced speech.

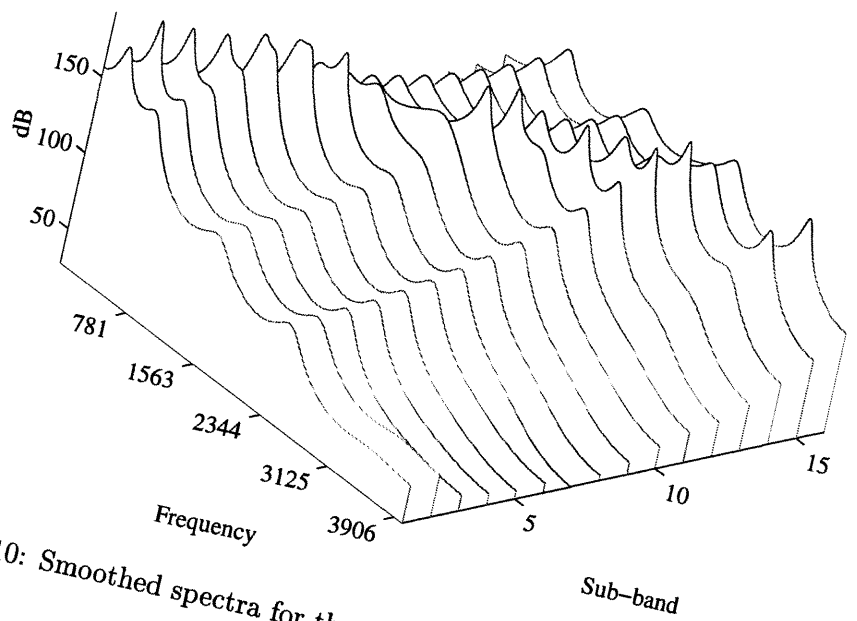

Figure 9.10: Smoothed spectra for the second next frame of voiced speech.

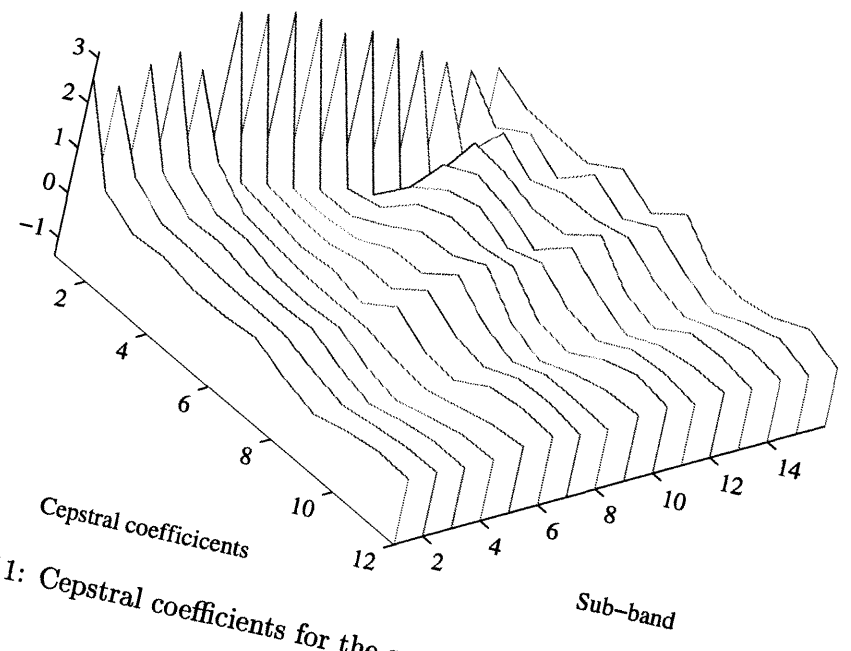

Figure 9.11: Cepstral coefficients for the second next frame of voiced speech.

cepstral coefficients is vital to sub-band processing, because it is the cepstral coefficients that are used as the feature set for the recognition systems. If the spectral variation visible in Figure 9.6 failed to make its presence felt in the content of the cepstral coefficients then there would be little or no difference between the sub-bands, and hence, no gains to be made form using sub-band processing.

Figures 9.8 to 9.11 show the the variation in the smoothed spectra and the cepstral coefficients across the sub-bands for the next two frames. They confirm that the findings for the frame just analysed are not unique to that frame and apply to voiced frames in general. Furthermore, the variation between frames is also visible, as the smoothed spectra and cepstral coefficients in each sub-band are different for each frame.

Each sub-band focuses the attention of the LP analysis on a different frequency range. So using sub-bands to model a speaker means that they are modelled in more detail than using a wide-band approach, where the LP analysis must cover the whole frequency range at once and cannot focus on individual sub-bands. This improvement in modelling the speaker should lead to a reduction in the identification error rate, as the speaker model should now be more accurate than before. This localised modelling cannot be achieved by simply increasing the order of the linear predictor of the wide-band system. Reynolds (1994) found that increasing the spectral resolution in this way degrades system performance by modelling spurious events or introducing too many parameters to be trained.

9.5 Summary

This chapter has described the sub-band processing implementation used for this work and how it differs from previous work. It has also shown how sub-band processing focuses on different frequency ranges to produce a more detailed speaker model. The next chapter cover the experimental work carried out using this approach.

Chapter 10

Sub-band processing experiments

10.1 Introduction

Having covered the implementation and effect of sub-band processing in the previous chapter, this chapter looks at the experimental work and how it compares to the more common wide-band approach. The experiments also examine the effect of varying the weighting used to combine the sub-band scores when calculating the overall score.

10.2 Experiments

As the location of each sub-band was fixed by the mel-scale, the experiments concentrated on the combination of the sub-band scores to get the final score. The most straightforward method is just to add the results of the sub-bands together without any weighting. This was the only combination strategy used by Besacier and Bonastre (1997) and one of several tested by Bourlard and Dupont (1996a).

The second approach to recombining the scores was to use some measure of each sub-band's speaker-dependent information content to weight the sub-band. This is similar to Bourlard and Dupont's weighting of the sub-bands on the basis of their speech recognition accuracy. Three measurements of speaker recognition performance were investigated as a means of weighting the sub-bands: the d', the identification error and the average equal error rate. The larger the d' for a sub-band the greater its weighting. The lower the identification error or equal error rate, the greater the weighting. Each set of weights was normalised so that

they summed to 1.

In each case the parameter used for the weighting could be determined in 3 ways, one of them *a priori*, the other two *a posteriori*. In the *a priori* case only the results based on the training data may be used. This works for d' and the average EER, but the training data identification error rate is zero, so it could not be used. The two *a posteriori* methods involve the test data alone and the test data combined with the training data respectively.

Finally, a genetic algorithm was used to weight the sub-bands. The GA would try to find whether particular sub-bands contained more speaker-dependent information than others. The weighting would emphasise these sub-bands at the expense of sub-bands with less speaker-specific information. d' was used as the means of evaluating the benefit of the weightings. The GA tests were also done using training, test and combined results. Because of the random nature of genetic algorithms (c.f. section 3.6), the GA was run 15 times with a population of 40 genotypes for 100 generations for each set of results.

10.3 Results

First of all, the average score and the average standard deviation of the scores for each sub-band are given in Figure 10.1. As there is little difference between the scores or their standard deviations, adding the scores together without weighting is not going to favour any band or group of bands over any other. The results for the case of just adding the results together are shown in Table 10.1. When compared with the results of the wide-band analysis (part of Table 10.1), it is clear that there are significant advantages to using sub-band processing. This is particularly true in terms of the identification error, which falls from 3.3% to 0.6% (in effect a decrease from 6 misclassified utterances to 1 misclassified utterance). As the identification error is also important to the score normalisation, the ICN average equal error rate results reflect this with a significant drop from 3.7% for the wide-band case to 1.4% for the sub-band case.

Figure 10.2 shows the average d', identification error and average equal error rate for the 16 sub-bands using the three possible evaluation methods: training set only, test set only and combination of training and test sets. These measures of speaker recognition accuracy were then used to generate weights (normalised

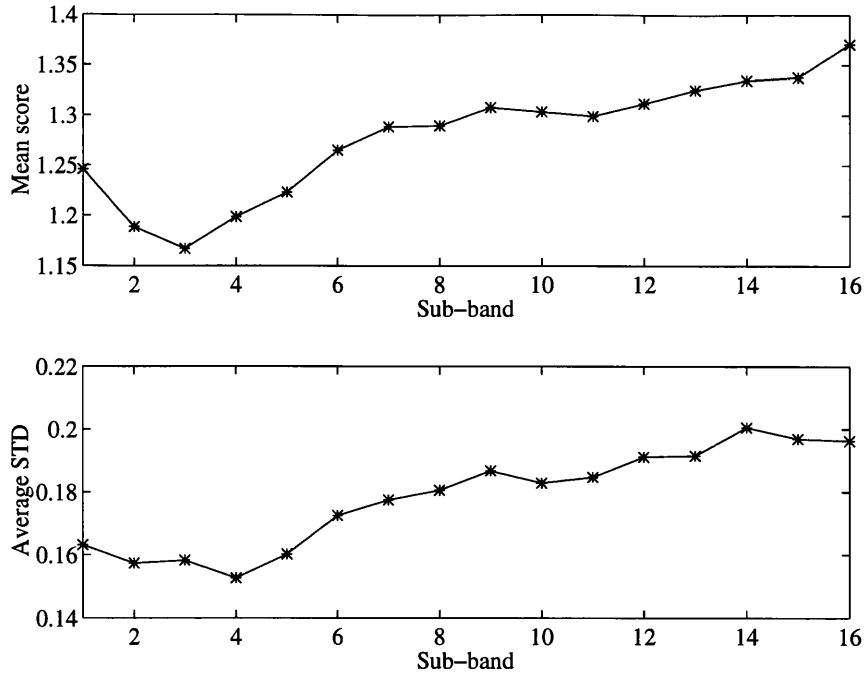

Figure 10.1: Mean scores and the standard deviations of the scores for each sub-band.

	Unnormalised					ICN		
	d'	IE	FR	FA	EER	FR	FA	EER
Wide-band	3.1	3.3	3.4	10.1	10.0	1.7	3.8	3.7
Sub-band	3.2	0.6	4.0	5.3	5.2	0.0	1.5	1.4

Table 10.1: Comparison of the wide-band and sub-band processing results (IE = identification error, FR = average false rejection rate, FA = average false acceptance, EER = average equal error rate).

to sum to 1), which are presented in Figure 10.3. The weights based on d' vary very little, while those based on the identification and verification error rates show some variation across the sub-bands.

The results of applying the d', identification error and average equal error rate weightings to the sub-bands are presented in Tables 10.2 to 10.4. The effect of the weightings may best be summarised by saying that they lead to no significant improvement over the unweighted results. This may not be completely surprising as the weights didn't differ very much from each other. However, this would also seem to indicate that there is a certain amount of robustness in the system. Small variations in channel scores don't seem to affect the overall score. The final

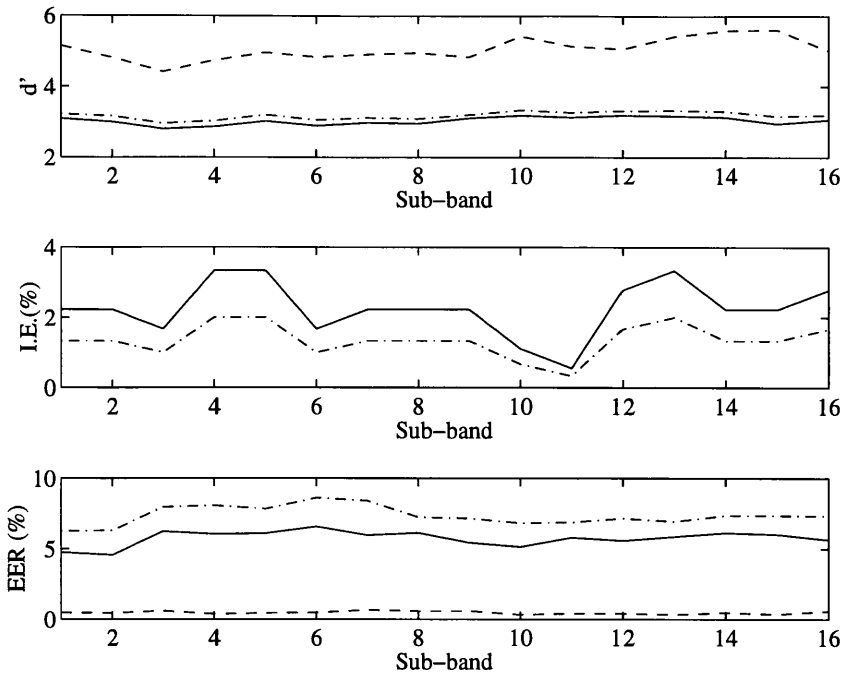

Figure 10.2: The average d', identification error and equal error rate for each sub-band using test only ('–'), training only ('– –') and both together ('– · –').

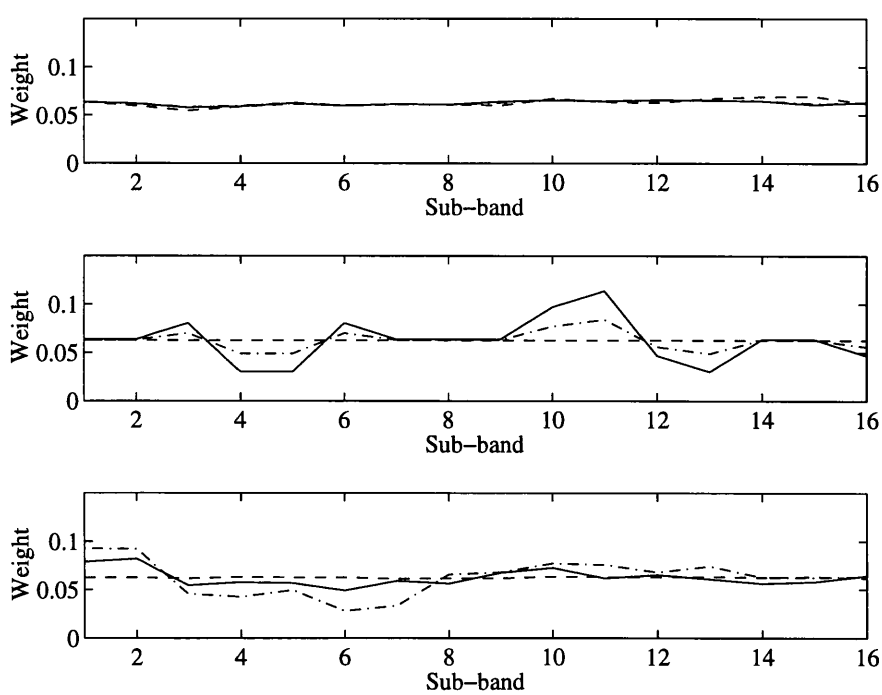

Figure 10.3: The weights (normalised to sum to 1) based on the average d', identification error and equal error rate for each sub-band using test only ('–'), training only ('– –') and both together ('– · –').

	Unnormalised					ICN		
	d'	IE	FR	FA	EER	FR	FA	EER
Test	3.2	0.6	4.0	5.3	5.2	0.0	1.5	1.4
Training	3.2	0.6	4.0	5.4	5.3	0.0	1.5	1.4
Combined	3.2	0.6	4.0	5.3	5.2	0.0	1.5	1.4

Table 10.2: Results for the d' weighting (IE = identification error, FR = average false rejection rate, FA = average false acceptance, EER = average equal error rate).

	Unnormalised					ICN		
	d'	IE	FR	FA	EER	FR	FA	EER
Training	3.2	0.6	4.0	5.4	5.3	0.0	1.5	1.4
Combined	3.2	0.6	4.0	5.4	5.3	0.0	1.5	1.4

Table 10.3: Results for the identification error weighting (IE = identification error, FR = average false rejection rate, FA = average false acceptance, EER = average equal error rate).

ICN average equal error rates for all the weights are virtually identical, despite variations in the original weights.

Even the application of the genetic algorithm to generate the weights, the results for which are presented in Table 10.5, made little difference. The average EER of around 1.4% is the same as that achieved by the other weightings.

The average weights generated by the genetic algorithm are shown in Figure 10.4. These averages were calculated from the 15 sets of weights generated using the test data, training data and both combined. The three approaches give very similar results. The first two sub-bands are quite favoured, but the

	Unnormalised					ICN		
	d'	IE	FR	FA	EER	FR	FA	EER
Test	3.2	0.6	4.0	5.2	5.1	0.0	1.5	1.4
Training	3.2	0.6	4.0	5.3	5.2	0.0	1.5	1.4
Combined	3.3	0.6	4.0	5.2	5.1	0.0	1.5	1.4

Table 10.4: Results for the average equal error rate weighting (IE = identification error, FR = average false rejection rate, FA = average false acceptance, EER = average equal error rate).

	Unnormalised					ICN		
	d'	IE	FR	FA	EER	FR	FA	EER
Test	3.3	0.6	4.0	4.9	4.8	0.0	1.6	1.5
Training	3.3	0.6	4.3	5.3	5.3	0.0	1.5	1.4
Combined	3.3	0.6	4.0	5.0	4.9	0.0	1.6	1.4

Table 10.5: Results for the genetic algorithm weighting (IE = identification error, FR = average false rejection rate, FA = average false acceptance, EER = average equal error rate).

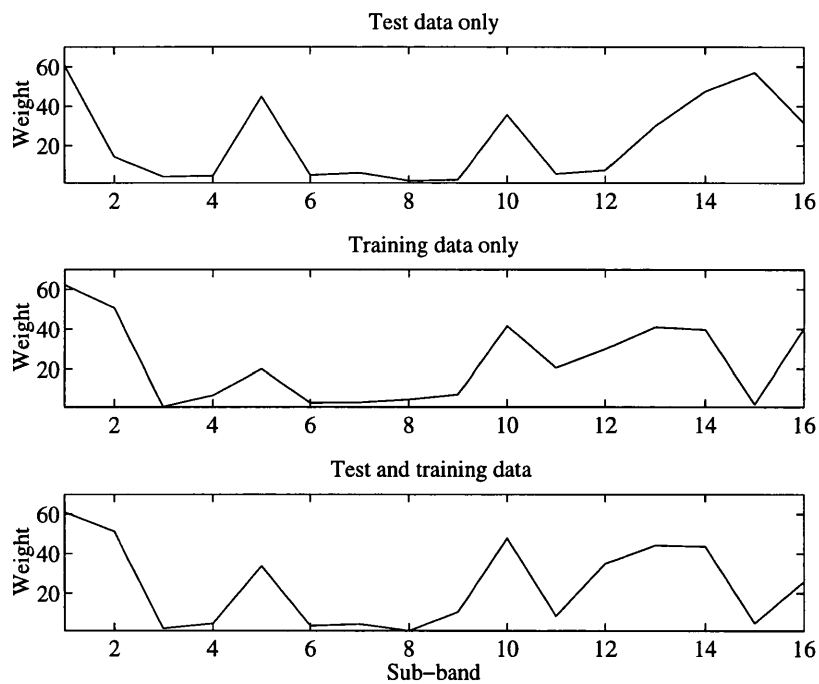

Figure 10.4: Average weights generated by the genetic algorithm.

sub-bands from 3 to 9 (centre frequencies 280Hz to 1218Hz) are rarely chosen. Although these sub-bands have been largely cut from the combination of the scores, the net advantage seems to be minimal, as the genetic algorithm results were no better than the unweighted case.

10.4 Discussion

The most dramatic effect of sub-band processing was to reduce the identification error rate compared to that of the wide-band system. Examples of this are given in Figures 10.5 and 10.6. The figures show the identification scores for two

utterances using both wide-band and sub-band processing. In both cases, the utterances were presented to all 12 speaker models. As it was an identification test, the model with the lowest score determined the speaker. In the wide-band case, the utterance from Speaker 1 is attributed to Speaker 6, and in the second example the utterance from Speaker 10 is attributed to Speaker 8. However, using an arithmetic mean combination strategy, the sub-band processing system correctly attributes the utterances to the correct speakers. In both cases the utterance still scores best against the same impostor model, but the score against the genuine speaker model is much lower. This would appear to indicate that sub-banding produces a better genuine speaker model.

The fact that the various weights, based on the speaker recognition capability of each sub-band, made no improvement on the arithmetic mean is in keeping with Bourlard and Dupont's findings. They got the same error rates when they used the arithmetic mean and a weighting based on the speech recognition accuracy of each sub-band. Better results were only obtained through the non-linear MLP approach. However, they also found that in narrow-band noise conditions, weightings based on speech recognition accuracy and the signal-to-noise ratio of the sub-band gave better error rates than the arithmetic mean.

The genetic algorithm weights, though failing to improve on the arithmetic mean combination, showed how similar results could be obtained even when several sub-bands were removed from the combination completely. This fact, and the way the speaker recognition weights failed to improve on the arithmetic mean, would seem to indicate that there is some inherent robustness to noise (in this case in the form of sub-band weighting) in the system.

Although the results here have proved to be a significant improvement on wide-band processing, the configuration used may not be optimal. The 16 sub-bands were chosen as they fitted with an established mel-scale filter-bank. It would be better if fewer sub-bands could be used to produce comparable results.

Another aspect that could be looked at is the feature set. In these experiments, each utterance was divided into 16 new band-limited utterances and then analysed. In the approach taken by Besacier and Bonastre (1997), filter-bank coefficients were grouped together and the actual recognition done using second-order statistical measures.

Overall, the results of the sub-band processing look very promising. The

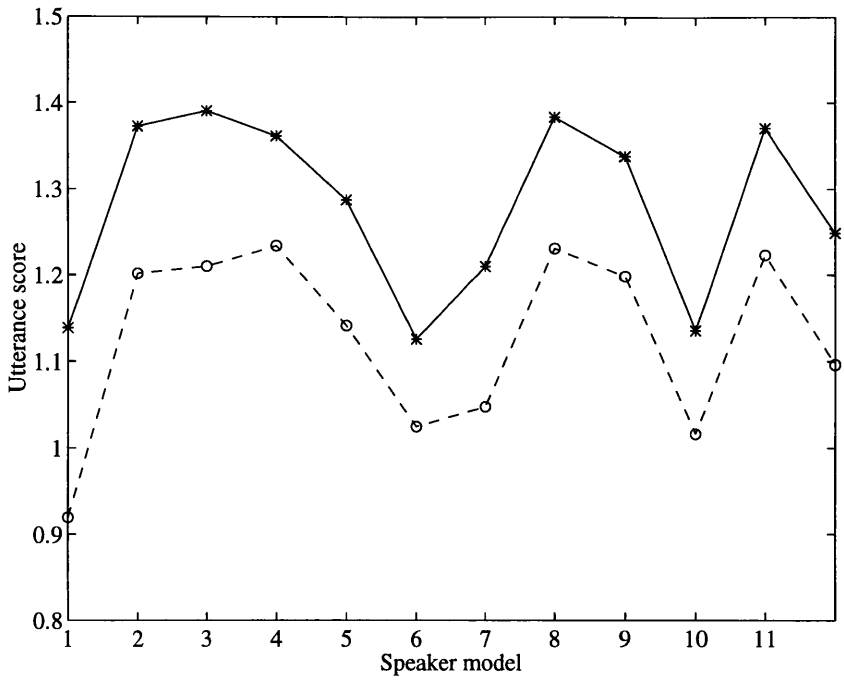

Figure 10.5: An example of how sub-band processing improves Speaker 1's self-test score compared to the wide-band system ('*' = wide-band scores, 'o' = average sub-band score).

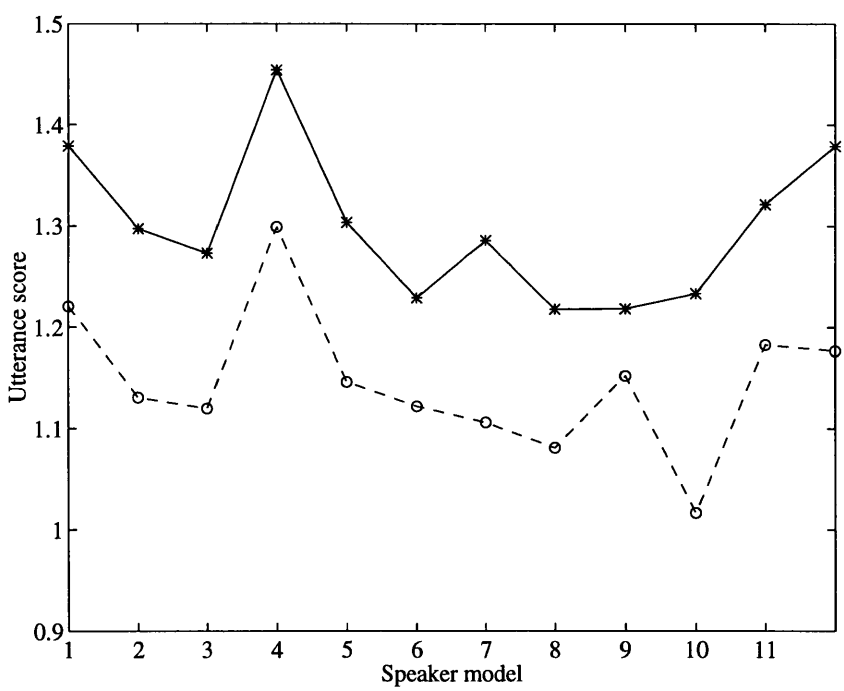

Figure 10.6: Another example of how sub-band processing improves Speaker 10's self-test score compared to the wide-band system ('*' = wide-band scores, 'o' = average sub-band score).

improved speaker models have resulted in lower identification and verification error rates than those of the wide-band approach. Further improvements should be possible through refinement of the sub-bands' width and location, and the recombination strategy.

10.5 Summary

This chapter has presented the results of applying a previously untried sub-band processing system to automatic speaker recognition. It has significantly improved on the wide-band approach both in terms of identification and verification error rates. Furthermore, no weighting was required to optimise the combination of sub-band scores. It has offered a credible alternative to the wide-band approach and merits more attention.

CHAPTER 11

CONCLUSION

11.1 Review of the experimental results

This section reviews the results of the experimental work carried out, while the following section discusses their implications for the field of automatic speaker recognition.

The experiments of Chapter 4 investigated the application of classifying neural networks to automatic speaker recognition and, in particular, the importance of the training set in determining the success of the system. The results indicated that classifying artificial neural networks, as implemented for these experiments, may have problems with a large speaker database. Their success was highly dependent on the specific data used to train the network. If some impostors were left out of the training set then the results deteriorated accordingly. As it would be impossible to include information in the training set for every impostor in a real-world database, this may prove to be a drawback. Attempts using VQ speaker models failed to determine *a priori* which speakers would be most likely to impersonate each other. If this had been successful, it may have been possible to reduce the impostors in the training set to those most likely to impersonate the genuine speaker.

Before moving to predictive neural networks, which don't use discriminative training, Chapter 5 considers score normalisation for distance models and a new means of ranking a speaker's impostors is presented. Rather than testing all the impostor training utterances against the genuine speaker model, only the impostor's VQ model was used. The impostor rankings generated using this approach were then used to create impostor cohorts for a score normalisation

method (ICN). Initial results indicated that the normalisation improved the separation between the genuine speaker and impostor score distributions.

The results of more comprehensive tests, presented in Chapter 6, confirmed that both the *model vs. model* impostor ranking and the cohort normalisation improved the verification error rate for both text-dependent and text-independent conditions.

Impostor cohort normalisation was then applied to PNN, and the details of the text-dependent and text-independent tests are given in Chapter 7. Despite using score normalisation and various frame-rejection strategies, the basic predictive neural network results could not match the results of the VQ system. However, work by other researchers, using predictive neural networks, indicated that the use of a multi-state ergodic model using Viterbi training would bring the results into line with those of the VQ system.

As different systems were producing results of a similar magnitude, Chapter 8 sought to decrease error rates through using the LP residual to complement the LPCC. Combination of the results from LPCC-based and LP residual-based recognisers led to a decrease in the identification error rate. The idea of using recognisers which focus on different areas of the speech signal was expanded on in Chapter 9. In an approach known as sub-band processing, the speech signal was divided up into 16 band-limited channels, with a separate recognition system for each channel. This approach to speaker recognition had not previously been tried and it proved to be very successful. Combining the results of the 16 sub-bands together produced a significant increase in performance over the wide-band processing normally implemented in conventional speaker recognition systems. No special weighting seemed to be required to optimise the combination strategy, with even genetic algorithms failing to improve on the unweighted case.

11.2 Discussion

The results described above have several implications for the field of automatic speaker recognition:

- Results from the classifying neural network experiments indicted that care must be taken with the make-up of the training data for discriminative training. In particular, it is important to include sufficient inter-speaker

variation, otherwise this may lead to an increase in the false acceptance error rates.

- The use of *model vs. model* scores proved to be an efficient way of ranking impostors. It is faster than the traditional method of testing the genuine speaker model with all the impostor training utterances, and provided a convenient way of selecting the closest impostors for cohort normalisation. It is possible that the rankings may also be applicable to other systems such as HMM and GMM, though this would require further testing.

- Impostor cohort normalisation (ICN), with cohorts determined from the *model vs. model* rankings, proved to be a successful way of normalising the scores as well as decreasing the verification error rate.

- Despite the fact that VQ is one of the oldest methods, when combined with a good normalisation method it gives a very good speaker recognition system. One of the main drawbacks appeared to be the use of cepstral vectors rather than the VQ method itself. That is not to say that there aren't gains to be made from using hidden Markov models or Gaussian mixture models. The benefits to be had from using more sophisticated systems may be limited, however, if all these systems use the LP cepstrum or mel-scaled cepstrum as their feature set. Furui (1997) raised this dependence on the cepstrum in his review of automatic speaker recognition, and it must be questioned whether this might be creating a performance ceiling for automatic speaker recognition.

- The novel implementation of sub-band processing for automatic speaker recognition led to significant decreases in identification and verification error rates. Each of the sub-bands emphasised a different frequency range, compared to wide-band processing which operates on the whole band-width at once. The emphasis on different frequency ranges created different cepstra across the bands and therefore created a more detailed model of the speaker than the wide-band approach.

- Sub-band processing should also be applicable to systems other than VQ. Although VQ is one of the easiest systems to implement, and the

optimisation of a sub-band processing system might be done using it, HMM and GMM should also benefit from this approach.

- Finally, one of the biggest problems encountered throughout the experiments was intra-speaker variation. This was highlighted in the temporal variation experiments, where inclusion of utterances from all recording sessions greatly reduced the error rates. Unless some means of predicting how a speaker's voice may vary temporally is determined, the problem of insufficient information for a complete speaker model will remain.

11.3 Future work

The following points indicate suitable areas of research to continue the direction of the work done to date:

- Although frame score normalisation helped to reduce the predictive neural network error rates, it was never used in conjunction with the VQ system. So, using frame score normalisation together with sub-band processing may further improve overall performance.

- Although various means of weighting the sub-bands were investigated, the structure of the system itself was not looked into. The number of channels chosen was not necessarily optimal, and fewer channels may do equally well. It may be possible to select the centre frequencies and bandwidths of the channels using genetic algorithms. This could even be done on a speaker-specific basis, if it turned out that some channels are better than others for certain speakers.

- Another issue to be addressed with sub-band processing for speaker recognition is the point at which the scores should be combined. In this work the scores were combined at the end of the utterance, though this might have been done on a frame, state (for HMM), phoneme or syllable basis.

- A comparison with the approach used by Besacier and Bonastre (1997) should also be undertaken. Their approach was quite different from that implemented for this work, using 24 mel-scale triangular-filter bank coefficients as a feature set and second-order statistical measures for

recognition. An evaluation of the two implementations in terms of efficiency and recognition rates may then be made.

- In keeping with the last point, if the sub-band processing implemented for this work compares well to the Besacier and Bonastre approach, then it should also be tested for HMM and GMM systems.

- The sub-band experiments carried out for this work used a small database of problematic speakers. To confirm the benefits of sub-band processing, more exact testing of both the text-dependent and text-independent databases should be done. Furthermore, tests should be carried out with added noise, to determine if sub-banding creates a more robust speaker recognition system than wide-band processing.

- A much harder task would be to search for a more speaker-specific feature set. Speech synthesis is one area which may be able to help with this. Work is ongoing into how to vary the synthesised voice so that it sounds like a specific speaker. To do this requires determining important speaker-specific characteristics which might be applicable to speaker recognition. One important aspect seems to be the glottal wave, which is ignored in most speaker recognition systems. Furui (1997) has alluded to the fact that new features, both macro-transitional and prosodic, may be necessary to enhance current recognition rates.

- Finally, it would be useful to be able to predict how a speaker's voice might vary over time. Although it is known how spectral averages and variances change with time, this knowledge is hard to incorporate in the speaker model. Ideally, new time-domain waveforms should be generated from the training data, which represent how the speaker might sound during testing. If this were possible, then models based on a single session could be modified to take account of how the speaker's voice might change in future tests.

11.4 Conclusion

This final chapter has summarised the research findings and their implication for the field of automatic speaker recognition, as well as indicating suitable areas for future research.

The thesis has put forward a new means of ranking impostors, which, when used for a cohort-based score normalisation, reduced verification error rates for both text-dependent and text-independent cases. Furthermore, a previously untried sub-band processing approach yielded marked improvements on the wide-band processing results for both speaker identification and verification tests. It improved the speaker model by emphasising different frequency sub-bands and dedicating a separate recogniser to each one. This provided more speaker-dependent information than the wide-band approach, which creates a single model of the speaker to cover the complete frequency range. The benefits of sub-band processing have yet to be explored to the full and their realisation should lead to further improvements in speaker recognition rates.

APPENDIX A

LP-DERIVED CEPSTRAL COEFFICIENTS

If the linear prediction filter is stable (which is guaranteed for the autocorrelation analysis used in this work), the logarithm of the inverse filter may be expressed as a power series in z^{-1} as follows:

$$\begin{aligned} C_{LP}(z) &= \textstyle\sum_{i=0}^{N_c} c_{LP}(i) z^{-i} \\ &= \log H(z) \\ &= \log \frac{G_{LP}}{\sum_{j=0}^{N_{LP}} a_{LP}(j) z^{-j}} \end{aligned}$$

where N_c is the number of cepstral coefficients, N_{LP} is the number of LP coefficients, c_{LP} are the cepstral coefficients, a_{LP} are the LP filter coefficients and G_{LP} is the gain of the LP filter $H(z)$.

We can solve for the cepstral coefficients by differentiating both sides of the expression with respect to z^{-1} and equating the coefficients of the resulting polynomials. This results in the recursion used by Atal (1974) to generate the cepstral coefficients from the linear prediction coefficients is as follows:

$c_{LP}(1) = -a_{LP}(1)$
for $2 \leq i \leq N_c$ {
$\quad c_{LP} = -a_{LP} - \sum_{j=1}^{i-1}(1 - \frac{j}{i}) a_{LP}(j) c_{LP}(i-j)$ }

Normally $c_{LP}(0)$ is defined as the log of the power of the LP error and is treated as a separate parameter (Picone, 1993). The frequency scale used to generate cepstral coefficients in this manner is linear, in contrast to the non-linear frequency scale of the mel-frequency cepstral coefficients.

Appendix B

Speaker sets

The following sets of speakers were used for the experiments described in this thesis.

BT Millar – 31 speaker set

The 31 male speakers from a single age-group: 1, 3, 4, 8, 9, 10, 11, 12, 14, 20, 21, 22, 24, 25, 27, 29, 30, 31, 33, 35, 36, 37, 38, 39, 41, 45, 46, 47, 49, 54 and 56.

BT Millar – 12 speaker set

The reduced set of 12 speakers: 1, 3, 4, 11, 14, 21, 25, 30, 33, 45, 46 and 56.

TIMIT 38 speaker set

The 38 male and female speakers from the same region: mcpm0, mdac0, mdpk0, medr0, mgrl0, mjeb1, mjwt0, mkls0, mklw0, mmgg0, mmrp0, mpgh0, mpgr0, mpsw0, mrai0, mrcg0, mrdd0, mrso0, mrws0, mtjs0, mtpf0, mtrr0, mwad0, mwar0, fcjf0, fdaw0, fdml0, fecd0, fetb0, fjsp0, fkfb0, fmem0, fsah0, fsjk1, fsma0, ftbr0, fvfb0 and fvmh0.

Appendix C

Sub-band processing of unvoiced speech

The following figures illustrate the effects of sub-bands processing on a representative frame of unvoiced speech (an explanation of the figures is given in section 9.4). Again we see that the different sub-bands emphasise different frequency ranges, causing the location of the LP poles to change from band to band. In fact, as the unvoiced spectrum is markedly flatter than the voiced spectrum, it is easier to see in Figure C.3 how the sub-bands focus on certain frequency ranges, as the peaks of the spectra move quite clearly from low to high frequencies across the sub-bands.

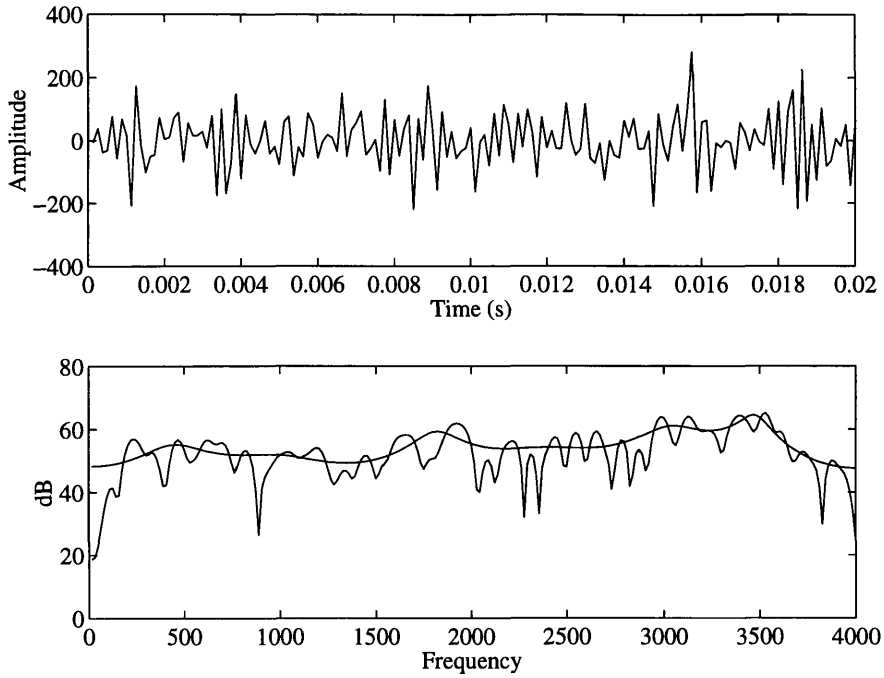

Figure C.1: Representative frame of unvoiced speech.

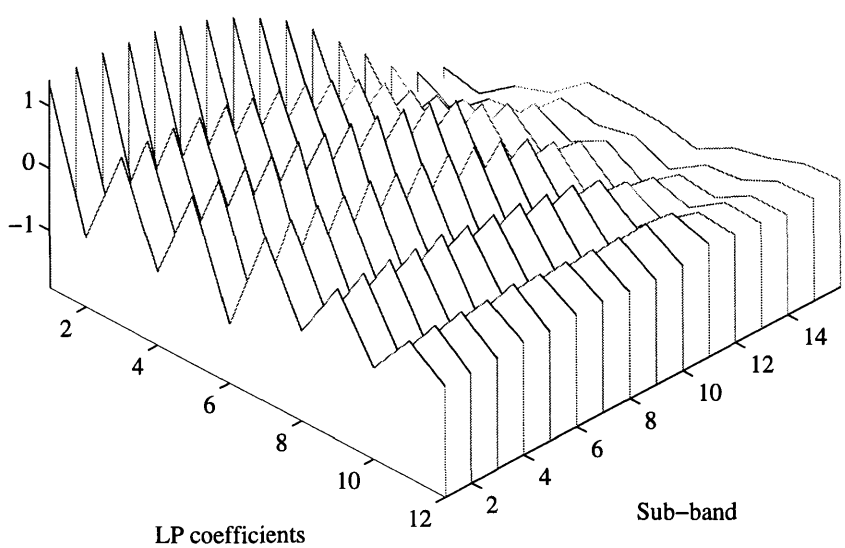

Figure C.2: LP coefficients for the unvoiced frame of speech.

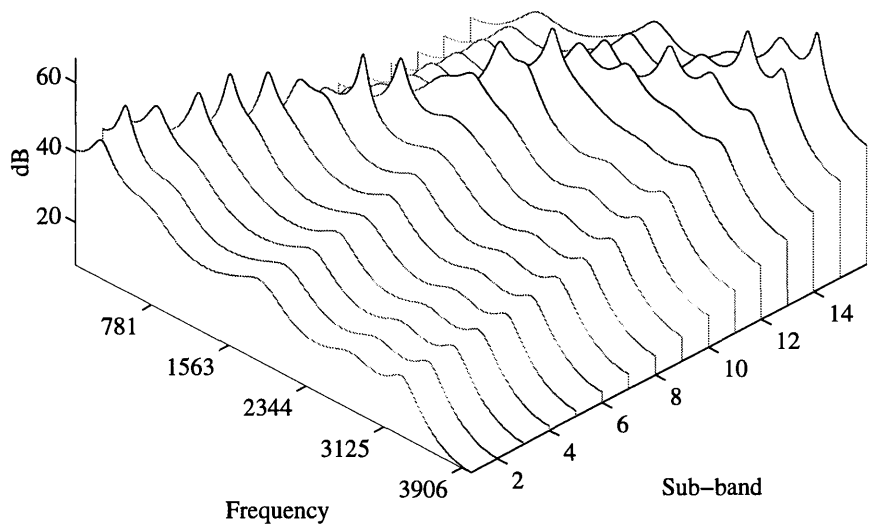

Figure C.3: Smoothed spectra for the unvoiced frame of speech.

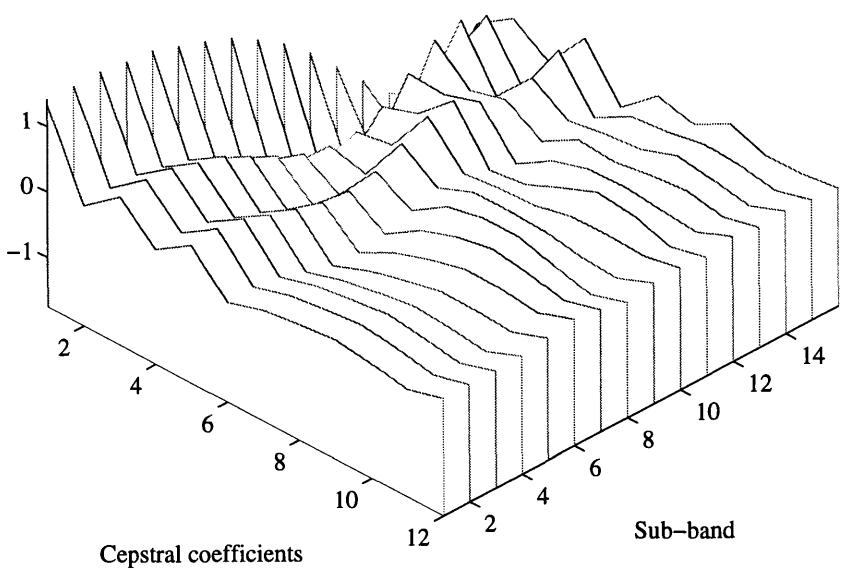

Figure C.4: LP cepstral coefficients for the unvoiced frame of speech.

Appendix D

Glossary of acronyms

ANN Artifical Neural Networks.

ATM Automatic Teller Machine.

BP Back-Propagation.

EER Equal Error Rate.

FA False Acceptances.

FFT Fast Fourier Transform.

FR False Rejections.

FSN Frame Score Normalisation.

GMM Gaussian Mixture Models.

HMM Hidden Markov Models.

ICN Impostor Cohort Normalisation.

IE Identification Error

LP Linear Prediction.

LPCC Linear Prediction Cepstral Coefficients.

MFCC Mel-Frequency Cepstral Coefficients.

MLP Multi-Layer Perceptron.

PNN Predictive Neural Networks.

RBF Radial Basis Function.

TD Text-Dependent.

TI Text-Independent.

TV Temporal Variation.

SLIMS Subtracting the Lowest Impostor Model Score.

SMIC Subtracting the Mean of the Impostor Cohort.

VQ Vector Quantisation.

REFERENCES

Allen J.B., 1994. "How do humans process and recognize speech?", *IEEE Trans. Speech Audio Processing*, vol. 2, no. 4, pp. 567–577.

Ambikairajah E., Keane M., Kelly A., Kilamartin L. and Tatterstall G., 1993. "Predictive models for speaker verification", *Speech Communication*, vol. 13, pp. 417–425.

Artières T. and Gallinari P., 1993. "Neural models for extracting speaker characteristics in speech modelisation systems", *Proc. ESCA Eurospeech '93*, Berlin, Germany, vol. 3, pp. 2263–2266.

Artières T. and Gallinari P., 1994. "Adequacy of neural predictors for speaker identification", *World Congress on Neural Networks*, San Diego, USA, vol. 4, pp. 601–606.

Artières T., 1995a. "Predictive systems for speaker identification: heuristics for model selection", *Int. Conf. Artificial Neural Networks*, Paris, France, pp. 241–246.

Artières T. and Gallinari P., 1995b. "Multi-state predictive neural networks for text-independent speaker recognition", *Proc. ESCA Eurospeech '95*, Madrid, Spain, pp. 633–636.

Atal B. and Hanauer S., 1971. "Speech analysis and synthesis by linear prediction of the speech wave", *Journal of the Acoustical Society of America*, vol. 50, pp. 637–655.

Atal B., 1974. "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification", *Journal of the Acoustical Society of America*, vol. 55, pp. 1304–1312.

Atal B., 1976. "Automatic recognition of speakers from their voices", *Proc. IEEE*, vol. 64, no. 4, pp. 460–475.

Bennani Y. and Gallinari P., 1994. "Connectionist approaches for automatic speaker recognition", *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, Martigny, Switzerland, pp. 95–102.

Besacier L. and Bonastre J-F., 1997. "Subband approach for automatic speaker recognition: optimal division of the frequency domain", *Proc. Audio- and Visual-Based Biometric Person Authentication*, Crans-Montana, Switzerland, pp. 195–202.

Bimbot F. and Mathan L., 1994. "Second-order statistical measures for text-independent speaker recognition", *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, Martigny, Switzerland, pp. 51–54.

Booth I., Barlow M. and Watson B., 1993. "Enhancements to DTW and VQ decision algorithms for speaker recognition", *Speech Communication*, vol. 13, pp. 427–433.

Bourlard H. and Dupont S., 1996. "A new ASR approach based on independent processing and recombination of partial frequency bands", *Proc. Int. Conf. on Spoken Language Processing*, Philadelphia, USA, pp. 426–429.

Bourlard H., Hermansky H. and Morgan N., 1996. "Towards increasing speech recognition error rates", *Speech Communication*, vol. 18, pp. 205–231.

Carey, M.J. and Parris E.S., 1992. "Speaker verification using connected words", *Proc. of the Institute of Acoustics*, vol. 14, no. 6, pp. 95–100.

Charvet D. and Jouvet D., 1997. "Optimizing festure set for speaker verification", *Proc. Audio- and Visual-Based Biometric Person Authentication*, Crans-Montana, Switzerland, pp. 203–210.

Cheng Y.M. and O'Shaughnessy D., 1989. "Automatic and reliable estimation of glottal closure instant", *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 37, no. 12, pp. 1805–1815.

Davis S.B. and Mermelstein P., 1980. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 28, no. 7, pp. 357–366.

de Veth J., Gallopyn G. and Bourlard H., 1993. "Limited parameter hidden Markov models for connected digit speaker verification over telephone channels", *Proc. ICASSP '93*, vol. 2, pp.247–250.

Deller J.R, Proakis J.G. and Hansen J.H., 1994. *Discrete-Time Processing of Speech Signals*, Macmillan, NY, USA.

Doddington G., 1985. "Speaker recognition – Identifying people by their voices", *Proc. IEEE*, vol. 73, no. 11, pp. 1651–1664.

Dubreucq V. and Vloeberghs C., 1994. "The use of the pitch to improve an HMM based speaker recognition method", *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, Martigny, Switzerland, pp. 15–17.

Finan R.A., Sapeluk A.T. and Damper R.I., 1996. "Comparison of multilayer and radial basis function neural networks for text-dependent speaker recognition", *Proc. ICNN '96*, Washington, USA, vol. 4, pp. 1992–1997.

Finan R.A., Sapeluk A.T. and Damper R.I., 1997a. "VQ score normalisation for text-dependent and text-independent speaker recognition", *Proc. Audio- and Visual-Based Biometric Person Authentication*, Crans-Montana, Switzerland, pp. 211–218.

Finan R.A., Sapeluk A.T. and Damper R.I., 1997b. "Predictive neural networks for text-independent speaker recognition", *Proc. 5th Int. Conf. on Artificial Neural Networks*, Cambridge, England, pp. 274–279.

Finan R.A., Damper R.I. and Sapeluk A.T., 1997c. "Impostor cohort selection for score normalisation in speaker verification", *Pattern Recognition Letters*, vol. 18, pp. 881–888.

Fredrickson S.E. and Tarassenko L., 1995. "Text-independent speaker recognition using neural network techniques", *Proc. 4th Int. Conf. on Artificial Neural Networks"*, Cambridge, England, pp. 13–18.

Furui, S., 1974. "An analysis of long-term variation of feature parameters of speech and its application to talker recognition", *Electronic Communications*, vol. 57-A, pp. 34–42.

Furui S., 1981. "Cepstral analysis techniques for automatic speaker verification", *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 29, pp. 254–272.

Furui S., 1986. "Research on individuality features in speech waves and automatic speaker recognition techniques", *Speech Communication*, vol. 5, pp. 183–197.

Furui, S., 1994. "An overview of speaker recognition technology", *Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, Martigny, Switzerland, pp. 1–9.

Furui, S., 1995. "Toward the ultimate synthesis/recognition system", *Proceedings of the National Academy of Sciences USA*, vol. 92, pp. 10040–10045.

Furui S., 1997. "Recent advances in speaker recognition", *Proc. Audio- and Visual-Based Biometric Person Authentication*, Crans-Montana, Switzerland, pp. 237–252.

Goldberg D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, MA, USA.

Green D. and Swets J., 1966. *Signal Detection Theory and Psychophysics*, John Wiley, New York.

Hannah M., Sapeluk A., Damper R. and Roger I., 1993a. "The effect of utterance length and content on speaker-verifier performance", *Proc. of ESCA Eurospeech '93*, Berlin, Germany, vol. 3, pp. 2299–2302.

Hannah M., Sapeluk A., Damper R. and Roger I., 1993b. "Using genetic algorithms to improve speaker-verifier performance", *Proc. IEE/IEEE Workshop on Natural Algorithms in Signal Processing*, 24/1–24/9, Chelmsford.

Hannah M., 1997. *Prospects for applying speaker verification to unattended secure banking*, PhD Thesis, University of Abertay Dundee.

Hattori H., 1992. "Text-independent speaker recognition using neural networks", *Proc. ICASSP '92*, San Francisco, USA, vol. 2, pp. 153–156.

Hattori H., 1993. "Text-independent speaker recongition using neural networks", *IEICE Trans. Inf. & Syst.*, vol. E76-D, no. 3, pp. 345–351.

Hattori H., 1994. "Text-independent speaker verification using neural networks", *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, Martigny, pp. 103–106.

Hayakawa S., Takeda K. and Itakura F., 1997. "Speaker identification using harmonic structure of LP-residual spectrum", *Proc. Audio- and Visual-Based Biometric Person Authentication*, Crans-Montana, Switzerland, pp. 253–260.

Haykin S., 1994. *Neural Networks - A Comprehensive Foundation*, McMillan, NY, USA.

Hermansky H. and Morgan N., 1994. "RASTA processing of speech", *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589.

Higgins A., Bahler L. and Porter J., 1991. "Speaker verification using randomized prompting", *Digital Signal Processing*, vol. 1, pp. 89–106.

Li K-P. and Porter J.E., 1988. "Normalizations and selection of speech segments for speaker recognition scoring", *Proc. ICASSP'88*, New York, USA, pp. 595–598.

Linde J., Buzo A. and Gray R.M., 1980."An algorithm for vector quantizer design", *IEEE Trans. Communications*, vol. 28, no. 1, pp. 84–95.

Mahanalobis, P.C., 1936. "On the generalised distance in statistics", *Proc. National Institute Science (India)*, vol. 12, pp. 49–55.

Mak M., Allen W. and Sexton G., 1994."Speaker identification using multilayer perceptrons and radial basis function network", *Neurocomputing*, vol. 6, pp. 99–117.

Matsui T. and Furui S., 1990. "Speaker recognition using vocal tract and pitch information", *Proc. Int. Conf. on Spoken Language Processing*, pp. 603–606.

Matsui T. and Furui S., 1992. "Speaker recognition using concatenated phoneme models", *Proc. Int. Conf. on Spoken Language Processing*, Banff, Canada, pp. 603–606.

Matsui T. and Furui S., 1995. "Likelihood normalization for speaker verification using phoneme- and speaker-indpendent model", *Speech Communication*, vol. 17, pp. 109–116.

Moody J. and Darken C., 1989. "Fast learning of networks of locally tuned processing units", *Neural Computation*, vol. 1, pp. 281–294.

Oglesby J. and Mason J., 1991. "Radial basis function neural networks for speaker recognition", *Proc. ICASSP '91*, Toronto, Canada, pp. 393–396.

Oglesby J., 1995. "What's in a number? Moving beyond the equal error rate", *Speech Communication*, vol. 17, pp. 193–208.

O'Shaughnessy D., 1987. *Speech Communication: Human and Machine*, Addison-Wesley, MA, USA.

Owens F.J., 1993. *Signal Processing of Speech*, Macmillan Press, Basingstoke, England.

Picone J., 1993. "Signal modeling techniques in speech recognition", *Proc. IEEE*, vol. 81, no. 9, pp. 1215–1247.

Rabiner L.R. and Sambur M.R., 1975. "An algorithm for detecting the endpoints of isolated utterances", *Bell System Technical Journal*, vol. 54, no. 2, pp. 297–315.

Rabiner L.R. and Schafer R.W., 1978. *Digital Signal Processing of Speech Signals*, Prentice Hall, NJ, USA.

Rabiner L.R., 1989. "A tutorial on hidden Markov models and selected applications in speech recognition", *Proc. IEEE*, vol. 77, no. 2, pp. 257–286.

Reynolds D.A., 1994. "Experimental evaluation of features for robust speaker identification", *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 4, pp. 639–643.

Rosenberg A.E. and Soong F.K., 1987. "Evaluation of a vector quantization talker recognition system in text dependent and text independent modes", *Computer Speech and Language*, vol. 22, pp. 143–157.

Rosenberg A. and Soong F., 1992a. "Recent research in automatic speaker recognition", *Advances in Speech Signal Processing*, eds. Furui and Sondhi, Marcel and Decker, pp. 701–738.

Rosenberg A.E., DeLong J., Lee C-H., Juang B-H and Soong F.K., 1992b. "The use of cohort normalized scores for speaker verification", *Proc. Int. Conf. Spoken Language Processing*, Banff, Canada, pp. 599–602.

Rumelhart D., Hinton G. and Williams R., 1986. "Learning internal representations by error propagation", *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, eds. Rumelhart D. and McClelland J., vol. 1, MIT Press, MA, USA.

Sambur M., 1974. "Selection of acoustic features for speaker identification", *IEEE Trans. Acoustics, Speech and Signal Proessing*, vol. 23, no. 2, pp. 176–182.

Soong F.K. and Rosenberg A.E., 1988. "On the use of instantaneous and transitional spectral information in speaker recognition", *IEEE Trans. Acoustics, Speech and Signal Proessing*, vol. 36, no. 6, pp. 871–879.

Thévenaz P. and Hügli H., 1995. "Usefulness of the LPC-residual in text-independent speaker verification", *Speech Communication*, vol. 17, pp. 145–157.

Thompson J. and Mason J.S., 1994. "The pre-detection of error-prone class members at the enrollment stage of speaker recognition systems", *Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, Martigny, Switzerland, pp. 127–130.

Tohkura Y., 1986. "A weighted cepstral distance measure for speech recognition", *Proc. IEEE ICASSP '86*, Tokyo, Japan, vol. 1, pp. 761–764.

Yu, K., Mason, J. and Oglesby J., 1995. "Speaker recognition using hidden Markov models, dynamic time warping and vector quantisation", *IEE Proc. Vision, Image and Signal Processing*, vol. 142, no. 5, pp. 313–318.

Zwicker E. and Terhardt E., 1980. "Analytical expressions for critical-band rate and critical bandwidth as a function of frequency", *Journal of the Acoustical Society of America*, vol. 68, no. 5, pp. 1523–1525.