Learning Models of Speaker Variation

Michael John Witbrock

July 12th 1996
CMU-CS-96-135

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Thesis Committee:

Scott E. Fahlman, Chair
John Bridle, Dragon Systems UK.
Alexander H. Waibel
Sheryl Young.

This research was partially supported by the National Science Foundation under grant IRI-9214873. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. government.

ACM Computing Reviews Keywords: talker variability, speaker variability, speech synthesis, speaker adaptation, FFT, LPC, log area ratio, voice transformation, neural networks, canonical discriminant analysis, principal components, speaker modelling, non-linear discrimination, non-linear compression, second order networks, classification figure of merit.

Carnegie Mellon

School of Computer Science

DOCTORAL THESIS
in the field of
Computer Science

Learning Models of Speaker Variation

MICHAEL JOHN WITBROCK

Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

ACCEPTED:

Scott E Fahlman — 6 May 1996
THESIS COMMITTEE CHAIR — DATE

[illegible] — 6 May 1996
THESIS COMMITTEE CHAIR — DATE

[illegible] — 15 May 1996
DEPARTMENT HEAD — DATE

APPROVED:

[illegible] — 5/17/96
DEAN — DATE

Abstract

Speaker based variability is an important component of the speech signal, whether it is regarded as a nuisance, impeding speech recognition, or a goal, improving speech synthesis. Although many speech recognisers attempt to avoid errors caused by speaker variation, and a few synthesisers attempt to produce a wide range of voices, these efforts tend to be narrowly focused on the task at hand, rather than based on a general model of the variation. What work has been done on modelling variability itself, on the other hand, has mainly aimed at understanding specific linguistic events, rather than at providing an implementation that is practical.

This thesis attempts to bridge the gap between these two approaches, by using statistical and connectionist techniques to separate out, and to model, the speaker variability component of the speech signal. A number of these models are built and examined for speaker specificity and speed of convergence. Two applications for speaker models are studied with mixed results: speaker adaptation without parameter reestimation for recognition, and mimicry by transforming the voice personality of synthetic speech.

Table of Contents

Acknowledgments

The author would like express his gratitude to Raymond Watrous, for his valuable efforts in recovering the Peterson-Barney vowel data and making it publicly available, to Scott Fahlman, Alex Waibel, John Mount and the late Dr. Sheryl Young at CMU, to John Bridle at Dragon Systems, and to Gerald Cheang at Yale for helpful discussions, and to Warren Sarle or the SAS institute, for his many informative posts to "connectionists" on statistical matters. Special thanks are due to Vidya Dinamani for rescuing the results of many of the experiments from a particularly voracious file-system at the PSC, and to Arthur McNair and Ricky Houghton for attempting to satiate my inexhaustible appetite for disk space and machine time. Thanks are also due to Russell Leighton, from whose Aspirin/Migraines connectionist simulator the CDA code used for these experiments was obtained, and to Richard Stern and Wayne Ward, for lending me papers, and general helpfulness. The author is indebted to the members of the Informedia digital library research group at CMU for their forbearance during the completion of this document, and especially to Dr. Alex Hauptmann for his generosity in taking on some onerous duties during the final months of writing. Especial thanks are due to the National Science Foundation for supporting this research under grant IRI-9214873 .

Finally a debt that cannot be repaid is owed to Antoine Brusseau, for his support and patience, and for remaining in Pittsburgh and our household as this work extended itself almost indefinitely.

Chapter 1. Introduction to Speaker Variation

In computer science, most of the effort spent studying speech has been directed towards the problem of speech recognition, where the goal is to reduce the speech signal to a finite, invariable symbol set and to discard any components of the signal that do not serve to distinguish these symbols. Much research has been directed at supporting this goal by controlling and decreasing speech variability. The work described by this thesis seeks to do just the opposite - discarding the symbols that carry the meaning of the speech, and instead studying the variability in the way those symbols are produced. In particular, this thesis studies the variability associated with speaker identity.

While this effort to understand the variability of speech is interesting in its own right, it also has practical import for speech-based technologies. In the speech recognition domain, some forms of speaker variation, which are noise with respect to the problem, are not *a priori* recognisable as such, and must be identified by a model of variability before they can be eliminated. Unfortunately, there is presently a lack of such models at any but a very gross level, and speech recognition systems must be trained to treat most such sources of variability simply as noise. In speech synthesis, it is clear that the effort to produce a variety of natural sounding voices would be greatly aided by a study of the variability that underlies that variety.

Many previous speech recognition applications have addressed speaker variability by adapting their parameters over time to each new speaker. Others have sought to identify new speakers with particular members of a small set of speakers on whose speech the system has already been trained. The work described in this thesis takes a novel approach: an explicit model of the dimensions along which speakers can differ from each other is built, and, faced with a new voice, the system identifies where the voice fits within the model. This position in "speaker space" constitutes a speaker code, which, if the model is adequate, represents salient features of voice quality. Applications that need to deal with a variety of new speakers are trained to use this code as a source of information with which they can make speaker-specific adjustments to their processing. By designing the model of variability so that it generalises over speakers, producing reasonable codes for speakers outside the training set, it becomes possible to adjust speech applications to new speakers by simply identifying the position of each speaker's voice in the space of the voice model.

1.1. What the thesis does

This thesis applies neural network and statistical techniques to this task of characterising speaker variability. Models are formed of the variability found in speech segments, and these models are combined to form an overall model of speaker variability. Speaker models formed this way are applied to two tasks:

- improving a speaker independent speech recognition system by allowing it to better handle a variety of speakers' voices, and

- modifying a synthesised voice to more closely match that of a target speaker.

The neural net based recognisers, when applied to a realistic speech recognition task, proved stubbornly unable to use information about speaker identity, whether presented in the form of a voice model or more explicitly. Applying this speaker information to a simplified recognition task from the speaker adaptation literature was more successful. Although experiments aimed at accounting for this performance gap gave some insight into what the important differences were, it remains unclear how sources of speaker information can be made useful to "speaker independent" recognisers working in realistic domains. It is possible that investigating the question using the better-understood variety of speech recognition system based on hidden Markov models will permit this question to be answered in future, but the resources to do so were not available during this thesis.

The second application, to speech synthesis, was more successful at using the information contained in the speaker model, allowing the production of a variety of different voices from the same synthetic speech source. When coupled with improved speech synthesis system, and after refinement of both the speaker modelling technique and the voice transformation systems, this work should provide a solid foundation for future work aimed at improving the naturalness and accuracy of speech synthesis for multi-speaker applications.

1.2. The nature of speaker variability

This introduction will review work on speaker variability to provide a general context for work in speaker modelling. Discussion of the literature that is more closely related to the two applications of speaker modelling to speech recognition and synthesis will be deferred to the chapters concentrating on those applications.

First, the review will cover the sources of variability in the speech signal, distinguishing those that contribute to speaker characteristics from those that do not. Then previous attempts to model that variability will be discussed, sometimes with reference to the target applications. Finally, the studied approach to speaker modelling by positioning speakers in a speaker space will be outlined and contrasted with other approaches. This outline will be expanded in the following chapters.

1.3. Sources of variability in the speech signal

The central difficulty facing those who would do speech recognition or speech synthesis is the fact that the single speech signal carries a great many different kinds of information, and that this information is intermingled in a way that makes it very difficult, and sometimes impossible, to decompose. Roughly, these sources of information can be divided into three classes:

- *Linguistic:* information that conveys the speaker's intent to the listener.
- *Speaker identity:* information that conveys permanent characteristics of the speaker
- *State:* information that conveys transient states of the speaker or environment that

are not relevant to the meaning of the signal.

1.3.1. Linguistic information

Linguistic information is conveyed by the speaker's exerting control over the rate of airflow, the tension in the glottis, and the positions and rates of movement of the jaw, tongue and lips, to produce the periodic voicing or aperiodic fricative noise that gives power to the speech, and to shape the acoustic cavity that filters that sound into a signal representing the intended information.

In a naïve model of spoken language, one might view the speech signal resulting from these processes as representing a stream of discrete units — symbols such as words or phonemes — that convey the information in the signal. However, important information is also conveyed by prosodic effects, such as the rate at which and rhythm with which someone speaks, the relative pitch and amplitude of various parts of the utterance, and even by special effects, such as the use of devoiced (*i.e.* whispered) speech or the deliberate production of a mean fundamental frequency that is lower or higher than that of the speaker's normal voice.

Linguistic information is also conveyed by signal characteristics that might otherwise be interpreted as speaker differences. Abe [abe93] studied a single speaker of Japanese reading samples of text from a novel, an encyclopaedia entry and an paragraph of advertising copy. He found that the speech produced for the different genres of text showed effects on vowel formant frequencies similar to those distinguishing speakers. He also found a strong influence from text type on sentence duration, and on the relationship between fundamental frequency and power. These deliberate changes to speaking style are important not only because they bear information and must be accounted for in any complete speech understanding or synthesis system, but, more immediately, because they blur the boundaries between speech segments. A more detailed review of these effects can be found in Eskénazi's work [eskenazi93] which covers the influence speaking-style has on speech, and Péan's work [pean93] which discusses a database being constructed to investigate these effects further. While it is clear that much has been already been found out about speaking style, the similarity between changes in the speech signal due to speaker differences and those due to speaking style suggests that a fully explicated model of how text genre is conveyed by speaking style will ultimately depend on the construction of a good model of speaker differences.

1.3.2. Speaker Identity

There are several permanent characteristics of a speaker that affect voice quality. Characteristics of the glottis affect both the natural pitch of the voice and the shape of the glottal pulses that drive voiced speech. The range of dimensions that the vocal tract can adopt at the speaker's will, and the dimensions of the tract when relaxed, including the position and size of the tongue and lips, affect the range of harmonics that can be produced from a given driving signal, and the harmonic content that the vocal tract is more likely to produce.

As well as these anatomically derived characteristics of voice personality, there are long term preferences for the allophones a speaker uses to instantiate a given phoneme, or even which phoneme to use, in a given context. These differences may be as glaring as the differ-

ence between an English and an American pronunciation of the vowel in *"can't"*, or as subtle as the differences in voicing onset time for vowels following stop consonants for francophones and anglophones speaking English [flege94]. There are also permanent preferences for prosodic characteristics, such as segmental duration[1], and the degree of pitch, amplitude or duration stress used to mark semantic and syntactic events in an utterance.

Unfortunately, the effects of these speaker characteristics on the speech signal qualities are often indistinguishable, at least in the short term, from the effects of the language. What appears, over the short term, to be a high fundamental frequency, characteristic of a speaker with a high pitched voice, may turn out, in the long term, to have been a linguistic effect, such as *sotto voce* speech. And, even more obviously, an American speaker can choose to pronounce the word *"can't"* in the English way.

In general, many of the characteristics that make voices distinctive if they are observed from a speaker over the long term, can be produced for purposes of communication in the short term.

1.3.3. State

It is not just the communicative intent and long-term characteristics of a speaker, described above, that affect voice quality. External or internal events can change the environment in which speech is produced, or the internal state of a speaker, and affect voice quality. Vroomen *et al* [vroomen93] showed that a speaker's emotional state, or affect, was reflected by duration and pitch changes[2] in speech, and that these changes were sufficiently pronounced that even stylised versions were sufficient to convey emotion in synthetic speech. Emotion affects speech in a variety of ways; Murray and Arnott [murray93], in their extensive review of the literature on vocal emotion, identify seven aspects of voice quality, including rate, pitch, intensity and mode of articulation, affected by vocal emotion.

Speakers who are tired, or who are under stress, produce speech reflecting those states [arbe 1980 and Sulk 1977 in murray93], and speakers in very noisy environments produce a characteristic voice quality called "Lombard speech". Junqua *et al.* identified changes in twelve components of voice quality for this latter speech, many of which, like pitch, vowel duration and formant frequency, are also important to voice personality [junqua90].

These state based changes in voice quality may be especially problematic for attempts to rapidly model voice personality. They tend to occur over a reasonably long duration, and to affect many of those voice features, such as pitch and speaking rate, that are characteristic of speaker differences. That this confusion should be present is unsurprising if one accepts the notion of Brown *et al.* [1974 in murray93] that personality itself is simply "the characteristic emotional tone of a person over time". It is likely, then, that a model of speaker differences would be aided by research enabling modellers to make the effects of emotion and other state on speech explicit, and by the collection of speech corpora in which visible components of speaker state were labelled.

---

1. Further complicated for languages, such as Japanese or Maori, in which segmental duration is phonemic.
2. It should be noted that the emotive speech in this study was deliberately produced, and therefore could more properly viewed as linguistic. One hopes that it can be assumed that natural emotions have similar characteristics.

1.4. Models of variability

The foregoing discussion will have made it clear that the desire to separate variation due to linguistic phenomena from that due to speaker characteristics will not be easily satisfied. Sometimes the effects of these sources of variation have exactly the same form. The only consistent difference between the sources is that speaker characteristics are nearly permanent, and should be apparent in the long term statistics of the speech signal from a person, and that state and linguistic phenomena affecting voice quality occur over a shorter time course. The aim of this thesis work is to build a model of a speaker's voice that can, in a sense, be "subtracted" from the speech signal to yield a more consistent signal for speech recognition, or "added" to a synthetic speech to mimic a particular voice. Since it is desirable to characterise a new speaker's voice quickly, it will be assumed that the major factors determining the form of the final speech signal are:

- the phone string to be produced, and
- the talker's voice quality.

Medium term voice changes caused by mimicking speaker differences, or by the speaker's emotional or other state, will be treated as if they were produced by different speakers. It is to be hoped that this assumption will not be harmful to applications. Before describing the method for modelling speaker variation, some previous work on the topic of talker variability will be reviewed.

The literature on speaker variation falls into two main classes:

- explicit attempts, with primarily linguistic motivation, to characterise the variability in particular parts of the speech signal, and
- attempts to deal with variability in the pursuit of some particular task, such as speech recognition.

Although the latter research has not been aimed directly at speaker modelling, an implicit model of speaker variability is often apparent. In this section, both kinds of models will be reviewed.

1.4.1. Explicit Models

Since variability amongst speakers is interesting in its own right, quite apart from its possible application in speech recognition and production systems, there have been a number of studies that have attempted to study this variability. Roughly, these can be divided by the degree to which the variability is expressed in terms of phonological rules, or in terms of statistical variability in the speech signal. Since the system to be described in this thesis relies on a model of the latter kind, it is on these statistical models that the review will concentrate.

Phonological rules for variability

Kimura and Nara [Kimura87] view speaker variability in terms of the choice of a particular set of phonological rules that are applied to an orthographic (spelled) transcription of an

utterance to transform it into a string of acoustical templates corresponding to the speech. The rule set they used was developed in the context of a speech segmentation system where the task was division of Japanese speech into a string of symbols representing phoneme variants. During training, the rule set, covering such transformations as palatalisation of consonants, nasalisation of vowels, and so on, was expanded for successive speakers whenever segmentation failed. A final set of 317 rules was sufficient to segment almost all of the speech. Interestingly enough, the rate at which new rules had to be added to the rule set to cover new utterances dropped dramatically after five speakers had been covered, suggesting that the types of phonological variability in the speech signal are reasonably few. Each speaker required about 53% of the available rules to describe his or her speech.

The interesting result in this work is that explicit models of the sources of segmental variability can be built, and that they can attain useful coverage from rules derived from relatively few training speakers. While the fact that each speaker uses a subset of the possible rules is interesting, it is not clear how one could use this characterisation in terms of rules to build a representation of a speaker's voice that could be used in other applications; it might take rather a lot of speech to decide which rule subset should be selected for a new speaker, especially if one wished to obtain probabilities for the application of alternate rules. Although it is possible that use of certain rules entails or predicts the use of others, and that such predictions could be used to group speakers by rule sets they are likely to use, this possibility was not investigated in the paper.

Vieregge and Broeders [vieregge93] looked for similar variability in a much narrower domain. They investigated variability in the realisation of the phonological variable /r/ in Dutch, where /r/ can have a variety of realisations depending on context. They found some talker specificity in the choice of realisation across speakers in some contexts, and also found variability in the degree of intra-talker variation. Unfortunately, insufficient data was available to clearly answer the question of whether even the variability in this single speech sound was regular and predictable from other speaker characteristics.

As will become evident when it is described, the model of speaker variation adopted for this thesis assumes that it is possible to identify which speech segments — typically, which phones — a speaker has used when producing an utterance, or that it is possible to chose correctly which phones to synthesise for a speaker. Although the research to date has not produced good algorithms to guide these choices, these studies of variation at the segmental level, since they can help provide a basis for these decisions upon which the model depends, will be very important at improving performance in the future.

Statistical characterisations of variability.

Although the statistical models in the literature tend to have a narrower scope than one might like, often concentrating on a limited set of phones or speakers, they do have the advantage over a purely rule based characterisation of variability that they can be learned fully automatically. They are also, of course, more directly comparable to the work reported in this thesis. Most directly comparable, perhaps, are members of a set of neural network models of speaker variation, which will be described in the next subsection.

One of these narrowly focused statistical models is found in the work of Heuvel *et al.* [heuvel93], who investigated the sources of variability in steady state portions of the three

Dutch vowels /a,i,u/ in C-V-C-ə context (e.g. "tatə"). They performed a discriminant analysis on bark scale spectra for the steady state portions in 10 repetitions each, by 15 male speakers, of these three vowels in 8 consonant contexts (/p,t,k,d,s,m,n,r/). The aim was to discover where in the vowel spectra the speaker-distinguishing information lay. Their con-

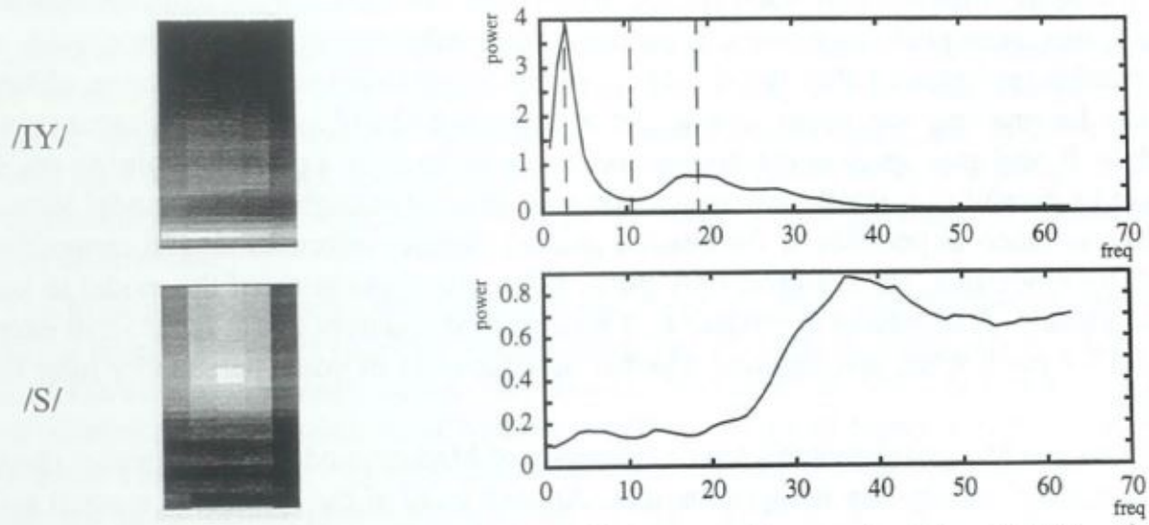


Figure 1: An example of the spectrum of two speech sounds, /S/ and /IY/ as in "see" (although the sounds displayed were not excerpted from an utterance of that word). The illustrations on the left have time on the x axis, frequency on the y access, and power represented as brightness. Each phone has been normalised to take up five time slices, or "frames". The plots on the right are the middle frame from each, with frequency on the horizontal and power on the vertical axis. For the vowel, /IY/, the middle three frames are nearly the same; this is the "steady state" portion of the phone. On the right, lines have been drawn to mark spectral peaks and troughs in this steady state portion. The frequency of the *n*th spectral peak is called the *n*th "formant".

clusion was that most of this information was to be found adjacent to the spectral peaks in the speech; that it was chiefly formant shape, rather than formant position or some quality of the spectral troughs, that distinguished speakers, and that, moreover, these distinctions were captured by around four discriminant functions. Principal components analysis required more functions to capture the same amount of variability. Although there seem to be some problems in the interpretation of these results based to the difficulty of distinguishing the effect of a small formant shift on the variability of the speech signal near the spectral peak from that of a change in spectral shape, these experiments are interesting for two reasons: The first is that the technique of using discriminant functions to highlight variability is one that will be used in building the model described in this thesis. The second is that the limited number of discriminant functions required to model the speaker variability in this admittedly limited domain gives some hope that a reasonably compact, and therefore easily estimated, statistical speaker model might be obtainable.

Ward and Gowdy [ward89] used even simpler acoustic measurements to distinguish speakers, in this case for an application to speaker verification. Pitch at three points in the vowel of the word "stop" was measured, along with the duration of voicing for that vowel. Even with this simple model, voices were somewhat separable: a 70% correct speaker iden-

tification rate was obtained when the acceptance threshold, based on Mahalanobis distance[3], was set at a point where the numbers of false acceptances and false rejections were equal. It is not surprising, of course, that pitch is an important source of information for distinguishing speakers. Perhaps more interestingly, for male but not for female speakers in this study, the timing information also aided speaker discrimination. While Ward and Gowdy do not believe that mean pitch, on its own, is sufficient for speaker discrimination, the speaker discrimination performance they did manage to obtain using rather simple measures, although too low for practical use, points to a danger in attempts to build more sophisticated speaker models: if one uses speaker discrimination as a goal to base a model's training on, one should be careful to make the discrimination task difficult enough that the model learns to capture as much as possible of the desired speaker identity information, and cannot "succeed" by modelling, for example, only pitch. When the application of the model to voice transformation is discussed in chapter 6, it will become apparent that it is far from easy to control for pitch when investigating whether other sources of voice personality have been modelled.

Mathan and Miclet [mathan90] built a hierarchy of Markov models to do speaker clustering on a small-vocabulary recognition task. At each level of the hierarchy, a small set of adaptation words was used to chose which of two subordinate trees of models to use. The models at the leaves corresponded to speaker clusters. While this clustering technique is quite sophisticated, allowing modelling of both acoustic and timing variability, the authors reported that results were disappointing. Performance was better than for a system using a single Markov model for all speakers, but only insignificantly better than a more sophisticated recogniser (a "bi-model" recogniser) using two Markov models run in parallel for each word. Moreover, this insignificant improvement was only possible after fifteen adaptation words had been uttered. They note that performance of the bi-model recogniser was improved if, during training, words from a particular speaker were used to train only one of the models. This paper reflects a general trend in the literature of disappointing recognition results using speaker clustering techniques, and, disappointingly in terms of efficiency, better results from the use of parallel recognition models. More work on speaker clustering will be reviewed in the next section on neural networks.

Tishby [tishby88] derived a mathematically sophisticated framework for describing the effects of known contributors to the speech signal in terms of a combination of a prior model, such as the state means and covariances used in a standard model, and a set of constraints describing observables, such as speaker identity, related to the new information to be modelled. The aim was to use these constraints to select the one probability distribution, out of the set of possible distributions satisfying the prior model, that also predicted the observables, such as speaker means, describing the new information and that had minimum cross entropy with the prior model. In setting the parameters to achieve this minimisation, a representation of the observations was formed that could be used for clustering. As a demonstration, the technique was used to extend a prior model that divided a set of speakers by sex for a set of acoustic states within digits into one that described speaker means for these states. Clustering in the parameter space used for the transformation distinguished speakers better than a system trained from scratch to do speaker identification. While the technique was applied to speaker identification in [tishby88], it is possible that with a large set of training

3. A measure of separation of distributions in a space that will be discussed later.

speakers, the parameter space formed could be used as a basis for a predictive speaker model. Once simpler models of speaker contributions to the speech signal of the sort described in this thesis have been successfully applied to a practical task, it may be worthwhile to investigate whether techniques such as Tishby's can be usefully applied to the general speaker modelling problem.

Lamel and Gauvain [lamel93] approached the problem of variability by training independent Markov models for each speaker, or in another case, for each sex. The problem of speaker variation was viewed as applying to the entire speech signal, and an entire Markov model used to model each speaker. In the final model, speakers were, in a sense, viewed as independent from each other, and no attempt was made to take advantage of regularities in voice differences.

The Markov models were run in parallel during recognition, and the speech labelled as coming from the speaker or group whose model had the highest probability of having produced the observed acoustic string. The problem of sparse training data was alleviated by first training a speaker independent Markov model and adapting copies of it to the individual speakers. The technique was very successful when applied to the tasks of speaker, sex and language identification, giving, for example, a text independent speaker identification rate of 98.3% after 2.5s of speech, for models adapted to TIMIT testing speakers starting from a seed model trained on the entire training set.

While this technique of running full Markov models in parallel is obviously worth considering if one wishes to do speaker ID from a known set of speakers, and while it clearly allows one to build a good model of each voice, it suffers from some deficits as a general speaker model. It is, of course, computationally expensive to run even a single Markov model recogniser. Running many in parallel compounds this cost. Moreover, the models must be pre-trained for speakers, and do not satisfy the criterion that speaker models should enable generalisation across speakers. Despite these difficulties, the success of this technique of running parallel specialised models makes finding generalisations that alleviate the problems an attractive prospect. Although doing so is outside the scope of the work discussed in this thesis, some approaches that might be taken will be discussed in the chapter on conclusions and future work (chapter 7).

A non-segmental model

The majority of models of speaker variation have attempted to characterise segmental variation, but this is not the only component of speaker difference. Itahashi and Tanaka [itahashi93] viewed the prosodic contour as an important component of variation due to dialect differences in Japanese. In particular, they examined f_0 contours for fourteen male speakers, each of whom represented a different Japanese dialect, reading a well known Japanese short story. These contours were approximated using a piecewise linear function. Eighteen aggregate statistics were calculated over parameters, such as starting F_0, slope and power, of the line segments. The resulting 18-element vectors were subjected to principal components analysis (PCA)[4]. The authors plotted the vectors for the fourteen dialects pro-

4. A technique for characterising variation that will be described in detail in later chapters.

jected onto pairs of the first five principal components and observed that in some of these projections the speakers appeared to group into linguistically plausible dialect clusters.

While one might have wished that the data had been gathered over a larger set of speakers, the paper serves the useful function of pointing out that prosodic contours are an important component of speaker variability, including variability between speaker classes. Explicitly modelling variation in the prosodic contour in this way would be well worth pursuing further, especially if a more sensitive model than aggregate statistics can be constructed.

Statistical models of variability such as those reviewed above reflect a useful approach to understanding speaker differences. However, they have generally been either too narrowly or two widely focused. A concentration on particular linguistic phenomena can be so narrow that it is of little obvious use in applications dealing with multiple speakers. Or a model can capture a great deal of the variability in a small set of speakers at the cost of a loss of computational tractability or the ability to generalise to new speakers. The aim of this thesis is, of course, to build a model that is sufficiently expressive to capture a useful amount of the available information about speaker variation, but simple enough to be practical. Statistical techniques, such as PCA, used in some of the previous work will be among the tools applied to this task.

Neural Network models of variability

More closely matching the initial intention of this work, there have also been a number of attempts to use neural network models to characterise variability. Although neural network techniques are, in some respects, very similar to the statistical models described above, there are important differences. The first of these is motivation: while the aim of building an explicitly statistical model is generally to understand the variability in the signal itself, neural net models are frequently construed as offering a model of how human information processing works. Neural net models may sometimes be built in the hope of understanding how people represent speaker variability, but it is not clear that this hope is warranted. The second main difference is that a neural network generally represents a larger class of possible modelling functions than a particular statistical technique. This may or may not be an advantage; in applying a particular statistical model to a set of data the modelling assumptions — the form one expects the data to take — are usually explicit, and the causes of failure or success of the modelling effort are generally understandable. The operation of neural network models tends to be more difficult to analyse. On the other hand, by representing a larger class of possible models, neural networks may have a greater chance of initial success.

Artières and Gallinari [artières93] looked to a neural network non-linear auto-regressive model, where speech frames f_{t-1} and f_{t-2} are used to predict frame f_t, to improve performance at speaker classification over that of similar linear auto-regressive models that had been previously used [bimbot92]. Speaker identification was done for fifteen speakers from the TIMIT database. Separate model networks were trained for each speaker.

Like the multiple Markov model system described above [lamel93], this system attempted to model each speaker's voice separately, in this case as a 1st order autoregressive process. No representation of the voice was formed, except in the weights of the network used to model it. Interestingly, from the point of view of building speaker models on the basis of

speaker discrimination, a more accurate model of the speech signal from each particular speaker did not necessarily translate into better speaker discrimination. They attempted to improve the extraction of speaker distinguishing information from the networks by learning to identify inputs that would generate correct speaker classifications. This boosted speaker recognition accuracy somewhat, but at a cost of requiring more input for a classification decision, since much of the information is thrown away. Nevertheless, the improvement is interesting, since it implies that appropriately chosen subsets of the speech signal can be used to improve the representation of speaker characteristics.

Konig and Morgan [konig93] constructed a rather simple model which viewed speakers as belonging to one of between two and five speaker clusters. The speaker code used was the long term average of cepstral parameters. These clusters were either supervised to distinguish men from women, or formed by an unsupervised *k*-means clustering in speaker code space. Neural nets were trained to classify incoming data into the clusters, and the binary decisions made by these networks used during training as an input to a phoneme classification network. In recognition, all cluster inputs were tried, and the one with the highest decoding probability was used for the whole utterance.

Results were disappointing, with not even the supervised clustering into males and females producing a significant performance improvement over the baseline. This disappointing result for speaker information added to the input of a single recogniser, as compared to schemes using separate recognisers for each group is consistent with the findings reported in chapter 5 of this thesis. The model explored by the Konig paper is one in which speaker differences divide speakers into acoustic clusters based upon long term spectral characteristics. Unfortunately, the assumption that the identity of these clusters can readily be used as additional information to neural net phonetic classifiers appears to be problematic, even when, as in this case, the system has the opportunity to try using all groups during recognition. Existing classifiers, at least, do not seem to be equipped to make good use of this type of speaker information.

Blackburn *et al.* [blackburn93] concentrated on speaker differences due to accent. They trained neural network classifiers to distinguish between Arabic-accented, Chinese-accented and unaccented Australian English when given features extracted from segmented phonemes as input. Separate networks were trained for stops, voiced and unvoiced phones, and energy dips, and their results combined over time to give an accent classification. Although classification error rates were not detailed in the paper, except by giving segment by segment accent confusion rates, the authors claimed that the system classified accents as rapidly as a trained phonetician. The model of variation implied by this work is, of course, that speakers fall into classes with transparent descriptions, and that these accent classes can be identified and used. If one is dealing with speakers with a variety of strong accents, it seems natural to assume that preclassification into these accent classes is likely to be useful, before attempting to form a more finely-grained speaker space, although, as the preceding paper showed, applying these classifications may still be problematic. What is even less clear is whether simple acoustic features such as those employed by Blackburn *et al.* are sufficient to describe more subtle forms of accent variation, such as those distinguishing speakers from different regions within the United States.

1.4.2. Implicit models

There have been few, if any, applications of explicit descriptive models of speaker variability to actual speech recognition or synthesis systems. Some systems, however, can be viewed as having a model of the nature of speaker variation that is implicit in their choice of a method for dealing with variability. Most of these systems are speaker-independent or multi-speaker speech recognisers, but such implicit models are also to be found in synthesis systems that allow user control of parameters meant to affect voice quality.

Most speaker adaptation schemes applied to speech recognition have involved partially retraining the system, before recognition, using a set of "adaptation" samples of the new speaker's voice. In essence, these systems have used a variety of methods to look for corresponding frames in the new and originally trained speakers' speech and to attempt to find a function that maps between them. When this mapping is found, it is either added to the system as a preprocessing stage used during recognition to relate codebook entries for the new speaker to those for the original speaker or speakers, or the training samples for the old speaker are converted into the voice of the new speaker via the mapping, and the recognition system is retrained with this new larger synthetic training set.

A number of recognition systems have used this adaptation by retraining scheme, with some variation in implementation. Furui [furui89] formed hierarchical trees, based on an inter-frame distance measure, for frames from both the reference and the new speaker, and then used distance measures computed between nodes of these tree structures to learn a transformation from position in the new speaker's tree into position in the reference speaker's. The system was designed to map between corresponding spectral clusters. Because this technique was based only on spectral structure, it could be performed on unlabelled, unprompted speech, and could be carried out during recognition. A disadvantage of the technique is that it required a large amount of training speech to estimate the cluster positions. In this model, voices were regarded as similar in structure, but different in realisation. While the acoustic frames emitted by a given speech state might differ, the relationship between states was regarded as consistent across speakers, enabling a correspondence to be found between trees.

Rigoll [rigoll89] adapted the IBM speech recognition system to a new speaker by having the speaker utter a subset (25% or five minutes) of the sentences that had been used initially to train the system. A mapping function was generated between speaker specific codebooks by time aligning the data from the old and new speaker, transforming the remaining 75% of the training data into the estimated templates of the new speaker, and retraining the system on the new synthetic and natural speech. Similarly, but using mapping during recognition, instead of during training, Nakamura and Shikano [nakamura89] had their system learn a mapping between a fuzzy labelling[5] of frames for the new speaker and a fuzzy labelling for the reference speaker in a standard hidden Markov model system. They defined a fuzzy labelling as a scheme where frames are represented by the set of probabilities that the frame was generated by each label.

Watrous [watrous91a] was one of the first to suggest the use of neural network models to separate out the effects of different sources of variation on the speech signal. He suggested

5. This system is called a fuzzy vector quantization in the paper.

that this variability was best modelled by regarding each phoneme as having a canonical form that is modified as each form of variation, such as loudness, context, or speaker, is introduced. He suggested, moreover, that these transformations were reversible. In experiments done using the Peterson and Barney database [peterson52,watrous91b], which will be described in detail later in the chapter on speaker adaptation in this thesis, Watrous showed that normalisation of formant frequencies, by specialised neural nets using multiplicative connections, significantly reduced classification errors. Inter-speaker variance was also reduced for phonemes from the TIMIT database, although the effect of this reduction on classification accuracy was not specified [watrous90,91a,93]. It was Watrous' early success with improving recognition of vowels from formant pairs by using speaker information, together with Cottrell's work on modelling variability in faces using compression networks [cottrell90], that initially motivated the work reported in this thesis. However, the technique used by Watrous required the use of labelled speech from the new speaker to train the input transformation, making it somewhat difficult to apply in many contexts where adaptation would be desirable, and required training of the transformation network for each new speaker, a relatively time consuming process. The aim of the work contained herein was to find adaptation techniques that did not share these faults. Unfortunately, the successes of Watrous' pilot experiments, and of the replications of them described in chapter five, were not matched when the same sort of normalisation techniques were applied to more ambitious tasks. It remains, therefore, a matter of controversy whether the effects of various kinds of variability on the speech signal are separately reversible.

Hernandez-Mendez and Figueras-Vidal [hernandez-mendez93], developed speaker models based on performing a "non-linear principal components analysis" of acoustic frames using neural networks specialised for each speaker. The speaker classifications produced by these models were used to combine the results of speaker-specific recognisers in a spoken digit recognition task for four male and two female speakers. Five repetitions for each of the ten digits were used for training, and five more repetitions, collected after a one month delay, were used in testing. High energy frames, representing voiced segments of the digits, were extracted and modelled by projecting them onto a lower dimensional space using self organising feature maps, radial basis functions, a variety of back-propagation networks, and principal components. Unlike the models described in [witbrock92] and in this thesis, models were not specialised for phonological units - frames for all digits were modelled in a single network.

Speaker identification performance using single digits from the test set ranged from 25% to 29% error for the network speaker model, and 35% error for the principal components model. Discriminant training, using null targets for frames from other speakers, improved speaker recognition performance by about 20%.

Results for applying the speaker discrimination from these speaker-identifying networks in a digit classifier were mixed. While choosing the best speaker on a digit by digit basis improved performance relative to choosing frame by frame or utterance by utterance, best recognition for training speakers was obtained by running all speaker dependent models in parallel. For novel speakers, merging speaker dependent recognisers using speaker models did not produce any additional gain in performance improvement when more than four speaker-specific recognisers were combined, and in general did not improve on the performance of a single multi-speaker recogniser.

In this paper, Hernandez-Mendez *et al.* modelled a speaker's voice quality by building specialised filters for their voices, and speaker similarity was judged by the amount of distortion introduced by these filters. A new speaker was modelled by using a mixture of these filters. Although some success from such filters might be expected in multi-speaker domains, even ones more realistic that the digit recognition task used, the models did not generalise well for new speakers. It is this failure to generalise that the explicitly speaker-independent voice models described in this thesis were intended to address.

Cox and Bridle [cox89,90, bridle91] described neural network based systems that, like Watrous' were based on the notion that there is a canonical form for speech that is affected by speaker dependent transformations to produce the final speech signal. Given a speech stream from a new speaker, their system simultaneously searched for the most probable sequence of recognised symbols and the most probably correct speaker dependent transformation function, according to these three ways of modelling the speaker effect:

- As a spectral bias - each speaker added a constant bias to each of twenty-seven spectral frequency channels.
- As a variable shift plus bias - each speaker added a fixed bias vector, plus a weighed combination of a three channel window to form a channel. The weights for the combination were fixed. This model represented a shift of up to one channel up or down.
- As a similar transformation, except the weights could vary across the spectrum, allowing variable but limited spectral shifts, expansions and compressions.

In these three models, the speaker differences were purely acoustic and uniform across time. There was no provision for varying the transformations depending on phone class, for example. On the other hand, since there were few parameters to be estimated, the small speaker adaptation effect available from these modelled speaker differences could be obtained with relatively little adaptation speech, and in relatively little time, event though back-propagation was used to do the search for speaker specific parameters.

Hampshire and Waibel's meta-pi network adapted a recogniser to new speakers by selecting amongst multiple speaker specific recognisers on a frame by frame basis, using a learned weighted average of the individual recognisers' outputs to make a final decision. This network was found to perform significantly better than a single recogniser trained on the multiple speakers [hampshire92]. This model differed from that of [hernandez-mendez93] in that the speaker model was the weighting between the recognisers, and was optimised on a case by case basis for each speaker, rather than being computed by an independent network. Speaker modelling was implicit in the way the recogniser operated, rather than being performed in a separate subsystem. The advantage of networks such as this, which are constructed to group parameter subsets by speaker, over a single multi-speaker recogniser, may lie in the ability of the speaker specific models to more sharply cover their input space, rather than modelling the between speaker variance. If this is the case, the model of speaker variation implied is one in which individual speakers have substantially less acoustic variability than multiple speakers, and where speakers cluster to a sufficient extent that using a particular speaker's models — or a weighted combination of a set of speakers' models — is closer to the truth about a new speaker than is an overall mean representation derived from all speakers' speech.

More conventional — non connectionist — recognition systems have done something with similar effect: by increasing the number of codebook entries[6] they use, they can divide the input space into output classes more accurately. This sort of technique, of course, is limited, in that it selects between codebook entries derived from different speaker classes on the basis of the single frame of input speech being processed. Each of these frames must still contain all the information needed for its classification. Such systems cannot use information from previous frames to improve performance, as human beings have been shown to do [mullenix89]. One could imagine a variation on this scheme in which the system is encouraged to use the same speaker model for contiguous frames, switching between reference speakers more slowly. In the systems reviewed this was done in one respect: Some systems, such as Sphinx [hwang94] have maintained both male and female models, and chosen the best one to analyse an entire utterance. While this does model a characteristic of the speaker, the technique is not easily extended to a wider set of speaker characteristics. The attempt has been made, of course. The system in [lamel93], described above, does the same thing in a much more dramatic way, maintaining parallel Markov models for all speakers.

All these systems share the characteristic that they effectively retrain the system, or at least a subset of the system, to be a speaker dependent recogniser for the new speaker. One can draw a distinction between this learning-based "adaptation" to a new speaker, and a system that is able to collect information about a speaker to make a transient adjustment in its recognition strategy. It is the latter strategy that is pursued in this thesis.

1.5. A space of speakers

The models of speaker variation evident in the work described above are limited in one respect or another. If they are explicit, they are designed to cover variation in a narrow phenomenon. This limits their applicability in two ways: it limits the amount of the speech signal that the model can make use of, and it may limit the applicability of the model to parts of the speech signal not explicitly modelled. For implicit models, one is limited to trying to build the complete model for a speaker only with the speech one has already heard from that speaker, or to regarding the speaker as identical to its nearest neighbour in some small set. The former limit makes it difficult to apply models to applications where limited speech is available from the target speaker. The latter limit makes it hard to model novel speakers with any fidelity.

1.5.1. What people can do

Human beings show a remarkable ability to handle speaker variation. Despite voice differences, the average person is able to understand utterances from a wide variety of speakers correctly and with little or no apparent effort. There is substantial experimental evidence showing that human listeners use information from earlier utterances to influence the processing of later input; the identification of vowels by human listeners is more accurate when the stimuli are drawn from a single speaker than when they are drawn from multiple speakers, for isolated vowels [assman82], and for consonants [Fourcin, 1968 in mullinix89]. Sim-

6. A codebook entry is a kind of "reference template" for part of the speech signal. The signal is modelled by comparing it to a sequence of these codebook entries.

ilar effects apply for recognition of monosyllables [creelman57]. These effects of talker change on human speech recognition accuracy apply strongly and consistently to whole word recognition tasks as well [mullenix89, mullenix90]. Since the effects include changes in processing latencies for words, the existence of a system for low level mandatory processing of speaker differences is implied. On the other hand, there is some dispute over how important these effects are in everyday performance [e.g. creelman57] and even whether they are important at all [verbrugge76]. More research is warranted in this area to fully understand the nature and role of speaker adaptation in human performance. Nevertheless, since, currently, speech recognition systems trained speaker-independently do suffer performance decreases relative to speaker independent systems, it is desirable for them to be able to improve their performance by adapting to the voice of the speaker, as humans do in at least some cases.

1.5.2. The model

What this thesis proposes is that there are underlying regularities in the way speakers' voices differ, and that these regularities can be used to amplify the usefulness of a small sample of a speaker's voice. Using data from a large number of training speakers, models are constructed for the variability of in each of a number of speech segments.[7] Using knowledge of which speaker said each segment, these segmental models are used to construct an overall model across speakers of the relationships between segmental pronunciation within a speaker. This overall voice variation model can be used with information, extracted from the appropriate segmental model, about the pronunciation of a given segment to make predictions about the pronunciation of unheard segments from the same speaker. Once it is trained, the model can be run in an entirely feed forward manner, allowing it to be applied in a straightforward manner to speech tasks that might benefit from speaker specific information. The feed-forward mode of operation also ensures that use of the model imposes a very limited computational demand.

1.5.3. Comparison with others

Earlier, speaker models were divided into two classes for review: explicit models that attempted to form a representation of vocal personality as such, and implicit models, that used some representation of voice personality formed in the course of trying to perform a task.

Since the model developed here contains an explicit representation of a speaker space, and can place speakers within it independent of any particular task, it is an explicit model. However, it shares, in some respects, more of the qualities of some of the implicit models. In particular, it is closely related, in application, to the models of Cox and Bridle, and of Watrous. The difference is that the speaker representations in their models are formed as the result of parameter optimisation in the course of a speech representation task, the current model is trained to produce a voice personality representation for each of a large set of training speakers in a manner designed to permit generalisation to new speakers. Beyond that needed

7. In fact, the models built in this thesis have always covered the variability in phonemes, but that is for practical, rather than theoretical reasons.

to form the model from the initial training set, no further parameter optimisation should be needed to handle new speakers.

1.5.4. Applications

The main aim of this thesis has been the construction of statistical and neural network models of speaker variation. Ideally, one would evaluate such models by measuring their ability to capture salient differences between speakers. Unfortunately a well articulated set of parameters for describing voice differences remains to be discovered. This makes the evaluation task more complicated. Although, for the sake of comparison, two measures of speaker discriminability are applied to the models to approximate a measure of speaker model quality, it must be noted that increasing the system's ability to distinguish speakers is not necessarily the same thing as increasing its utility in describing their voices.

Since a direct measure of model quality is lacking, the model has had to be evaluated in the context of specific tasks for which voice differences are important, at the cost of not inconsiderable extra effort. If the model can provide the information about speaker voice needed by some realistic applications where speaker differences matter, it will have shown its virtue as a representation of a speaker's voice quality.

Recognition

One of the motivations for this research was the reported and evident ability of human beings to use rapid adaptation to new speakers to improve speech recognition accuracy. This, together with the extensive literature showing improvements in recognition accuracy for speech recognisers adapted to new speakers led to the belief that a connectionist speech recogniser, given an adequate description of a speaker's voice, would be able to use the information to adjust its classification surfaces, leading to improved recognition performance. Regrettably, it turned out to be the case that even providing a perfect speaker description, in the form of speaker ID, to such nets, had little effect on recognition accuracy. Although it was possible to get some information about why this may have been the case, the fact remained that this task was not serving well as way to measure model quality. In future work, it is hoped that there will be opportunity to apply speaker models similar to those described here to a hidden Markov Model (HMM) based speech recognition system. The explicit statistical representation of the spectral characteristics of the input signal used by these models, by allowing an understanding of the relationship between the position of a speaker in speaker space and the acoustics of the speech signal they produce, may allow an explanation of why the neural net based recognisers were unable to make use of this information. It is to be hoped that this understanding will permit the design of recognisers that are able to use speaker identity information, in a single recogniser, without retraining.

Voice transformation

Since speech synthesis based on the model would permit direct perception of the effects of different speaker descriptions, voice mimicry was selected as an alternative application domain. Using networks trained to convert SoftTalk[8] speech into the voice of the speaker described by the speaker voice description, it is possible to produce speech that, although

not of terribly high quality, is similar in sound to that of the modelled speaker. The low quality of the converted speech is at least in part due to the use of synthetic speech as the source signal that is converted into the voice of the target speaker. Further quality deficits were attributable to the fact that it was necessary to work entirely with speech represented in a reasonably low quality linear predictive coefficient (LPC) encoding. The performance of the model and of the voice conversion system was evaluated on the basis of perceptual experiments using human subjects.

1.6. Outline of chapters

The next chapter gives an outline of the speaker model, and the speaker voice descriptions it yields. Chapters 3 and 4 describe the model in more detail: the first of these describes the phone pronunciation models from which the overall model is built, and compares a variety of techniques that were tried for forming these phone models; chapter 4 describes the overall speaker model, again comparing a number of methods for obtaining it. Chapter 5 describes the experiments with speech recognition, including some reasonably extensive work exploring the circumstances under which a recogniser would use a speaker model. Chapter 6 describes the application of the speaker model to mimicry synthesis by voice transformation, both comparing competing voice models, and various methods of achieving the voice transform for quality. Finally, chapter 7 draws overall conclusions from the presented experimental work, and suggests future directions to take, both with voice modelling, and with mimicry synthesis.

8. A commercial speech synthesis system produced by Digital Equipment Corporation.

Chapter 2. Measuring Voice Characteristics

2.1. Introduction - Speaker Models

The most important quality of a useful model of speaker variation is productivity. The model must produce a representation of previously heard speech that allows it to make predictions about future speech from the same speaker. For speech synthesis, the aim is to alter synthesis parameters and produce novel utterances in the speaker's voice. For speech recognition, the aim is to alter a system's expectations about such things as phoneme boundaries in acoustic space and the timing of forthcoming speech, so as to improve its recognition performance.

As a degenerate case, a set of labels uniquely identifying all speakers is a suitable speaker model, when used with a application system that has been trained on the same speakers. A recognition system using such a label model must have heard enough speech from a given speaker to estimate the speaker specific parameters it uses to set classification boundaries. A similar synthesis system must have learned a suitable set of synthesis parameters for each speaker. In this degenerate scheme of using labels as a model, the modelling system would have the task of identifying the speaker and assigning the appropriate label.

In the more interesting case that speech from novel speakers is to be recognised or mimicked, or where a reliable method of identifying speakers is not available, it is necessary to look elsewhere for a model. Since there is available no *a priori* knowledge about whom the speech comes from, a model must describe a means of extracting a set of features, describing the speech personality of any particular speaker, from the speech signal itself. Henceforth, such a set of features extracted for a particular speaker will be called a Speaker Voice Code (SVC). Since it is desirable to form this SVC as rapidly as possible, and to have the SVC easily useable by application systems, one criterion for choosing the model's features is that they should be as stable as possible within a speaker. Of course, the features must also successfully distinguish what *is* distinct about the speech of different speakers, preferably in an application independent manner.

If the parameters are chosen appropriately, so that speaker class characteristics can be exploited, this sort of general speaker model should allow better estimation of adaptation or synthesis parameters, where limited training data is available for each speaker, than is available from speaker identity alone, since parameter estimates can be smoothed by those of other similar speakers.

2.2. Elements of a model

For speech technology applications, at least, the most important aspect of the speech signal is its symbolic content: a string describing *what* is said. In speech recognition, the aim is to transcribe or otherwise identify this string, or at least some useful part of it. In synthesis, the aim is to produce speech corresponding to the string. Although the assumption is not

completely warranted, for the current purposes it can be supposed that the content of the speech is independent of who is speaking. In building the speaker voice model, it is clearly not desirable to capture variation due to what is being said. In fact, the goal is to avoid doing so, by separating this variation from the rest of the signal, and only modelling what is left. How each of the audible symbols (lexemes, for example, or smaller "segments" such as syllables or phonemes) making up the meaning of an utterance is represented in the speaker's voice is the essence of speaker variation. The elementary modelling technique used in this thesis is that of holding symbolic content constant, modelling the variation *within* acoustic symbols, and then combining the representations of this variation into an overall model of the speaker's voice. The choice of symbols will be discussed more thoroughly later, but generally a subset of phonemes will be used.

It is, of course, the case that the meaning of an utterance is also partially conveyed by prosody. Pitch, amplitude, rate, presence of voicing and other components of the speech signal can be varied at both the segmental and suprasegmental level to change the meaning of the string. Moreover, differences in the way these prosodic effects are applied are an important part of voice personality - distinguishing English from American voices, for example. Lacking an adequate method for separating prosody from other components of the signal, or the personality related aspects of prosody from the semantic, there is little choice but to treat prosody as simply a source of noise in the speaker model.

The technique, then, is to segment phones from the speech signal using a phoneme labeller,[1] to model the variation between examples of the same phone uttered by different speakers, and to combine these models into an overall speaker model. Inconveniently enough however, acoustic symbols vary greatly in duration, whereas, in general, the available techniques for modelling variation require that their input be a set of fixed length vectors. As a first step, therefore, it is necessary to produce fixed length vectors from the phoneme segments in the speech signal. This will be done by simply applying a linear transformation to stretch or shrink the phoneme segments extracted from utterances to constant duration. Other methods of fixing the length of speech symbols will, however, be discussed in a later section (3.5.).

The application of these speaker models, or rather of the SVCs produced by them, depends on it being possible to train the target system (a synthesiser or recogniser, for example) with the SVCs from some limited set of training speakers, and on the system being able to generalise to using SVCs from novel speakers to aid it in its task. For this hope to be justified, the space of SVCs has to be reasonably well populated by the training speakers, so the target application can learn a speaker adaptation that reasonably interpolates between training speakers. Two conditions must be met: 1) the number of training speakers must be reasonably large, and 2) the parameters making up the SVC must be relatively few, limiting the dimension of the space that must be filled.

Introspection about the experience of listening to voices suggests that it should be possible to find a relatively small number of parameters that represent voice differences with adequate fidelity. Since representation vectors derived from speech units by signal processing tend to be rather long, statistical and neural net techniques will be used to find a lower dimensional subspace onto which these phone based vectors can be projected, while still

1. Although for all the experiments reported here, an oracle in the form of a pre-labelled database was used.

retaining the important voice quality information. First, descriptions for speakers vocal quality when producing individual speech units (the phoneme /IY/ for example) will be found, and then these phone descriptions will be combined into an overall speaker model. For consistency, these reduced segmental representations will be called Phone Pronunciation Codes, or PPCs. The next chapter (3) will describe the experiments that were done to explore and evaluate different kinds of PPC.

The final stage in building the speaker model is to combine the models of pronunciation of individual symbols into an overall voice model. This model is intended to capture the regularities that exist in relationships between the pronunciations of different phones by a single speaker, for example, the way the pronunciation of the phone /ix/ varies with the pronunciation of /ah/. This model will be relied upon to predict the sounds of unheard speech units from the sounds of previous heard units from the same speaker. Like the symbol codes (the PPCs), this speaker model should produce SVCs with as low a dimension as possible. Its derivation and use should also be robust in the face of missing inputs, since it is not possible to rely on having a complete speech symbol inventory for a speaker during training, let alone from the short segments of speech the speaker modelling system should be able to make use of when applied. Again connectionist and neural net techniques will be used to build a variety of candidate models. These experiments are described in Chapter 4.

2.3. A small example model

The requirements described above define a class of voice models. Before the discussion continues, in later chapters, to cover a comparison of some of the members of that class, the next few pages will be used to describe a particular instance from this class. This should serve to illustrate each of the steps involved, and provide a framework for the comparison between models in later chapters. In this illustrative example, attention will be paid to the degree to which the model satisfies the criteria for the utility of models that have been have set above, mainly with an eye to comparison with later models.

As is the case with nearly all of the work described in this thesis, the model was trained and tested using the TIMIT acoustic phonetic corpus [fisher86, lamel86]. This corpus is described in some detail near the beginning of the next chapter (§3.1), but for this illustration it should be sufficient to note only that it contains about thirty seconds of phonetically labelled speech from each of 630 speakers from eight "regions" of the United States. For this model speech from only regions 1, 2 and 3 (New England, Northern, North Midland) was used. For training, the speakers from the "train" subset of the database were used, and for these speakers, only speech from the 5 "sx" (phonetically compact[2] sentences per speaker was used. There were 190 speakers in the training subset used, and a total of 950 utterances.

For the sake of speed in training this model, use was only made speech from the following ten phonemes,[3] which occurred most frequently in the studied section of the database: /ix/, /s/, /n/, /iy/, /tcl/, /r/, /l/, /kcl/, /dcl/, /k/.

2. These sentences were designed to provide good coverage of pairs of phones, and to include extra occurrences of phonetic contexts thought to be difficult or of particular interest by the corpus designers.

3. A guide to the representation used in this document for phones and phoneme is contained in Appendix A.

2.3.1. Signal Processing

Using the phoneme level labels available for all sentences in the TIMIT database, each example of any of the selected phones was excerpted from the speech recordings in the database. The digital recording of the excerpted phone was zero padded to ensure that it was a multiple of 128 samples long, and an FFT power spectrum computed on non-overlapping 128-sample frames, yielding 64 power values per frame. These variable length collections of frames were used to build phoneme models. Further details on these signal processing steps can be found in §3.3 to §3.5, to which a reader unfamiliar with speech processing techniques may wish to refer.

2.3.2. Phoneme models

All FFT spectral frames for each phone were warped, using a linear distortion (§3.5), to a constant five frame duration. Each of the phones (which will be indexed with i) was, at this stage, represented by a vector $\vec{p}_i$ of 320 spectral coefficients. This fixed-length real-number representation is suitable for further processing by connectionist networks or multivariate statistical analysis.

As an indication of the amount of training data used for building these phone models, the phoneme /kcl/, one of the less frequent of those selected, was represented by 1058 of these 320 element vectors, or an average of 5.6 occurrences per training speaker.

As noted above, it is desirable for phone models to be compact, consistent within a speaker, and distinct across speakers (who are indexed here by j). The technique from multivariate statistics that is generally used to achieve these goals is Canonical Discriminant Analysis (CDA) [james85, dennis91]. Although it will be described more thoroughly in the next chapter, briefly, CDA involves computing the eigenvectors of the ratio of the between groups[4] and within groups[5] covariance matrices. The data is then projected onto the eigenvectors corresponding to the largest eigenvalues of the ratio matrix. Putting aside the technical description, the important thing to note is that this projection turns out to be the linear transformation of the original data that maximises the discriminability of the groups, while resulting in vectors having the desired dimension. This analysis was applied separately to each of the ten phonemes. In this application, since the aim is to distinguish pronunciations of a phoneme from different speakers, the goal of discrimination is to separate phone instances into groups by speaker identity.

A modified[6] version of the Aspirin CDA software [dennis91] was used to compute canonical discriminants for speakers over the training vectors for each phone, and the data were then projected onto the first eight of these discriminants. After this projection, each phone pronunciation was represented by those eight linear combinations of the original 320 components in the $\vec{p}_i$ vectors that maximally distinguished between speakers. The choice to

4. The within groups covariance matrix is the covariance matrix of sample vectors after group (speaker) mean vectors have been subtracted, and measures the variability of samples of a phone within a single speaker.
5. The between groups covariance matrix is the covariance matrix of group (speaker) mean vectors, and measures the amount of dispersion of speakers with respect to a given phone.
6. The modifications were simply made to increase the size of the data that could be handled, and to provide more flexibility in the way results were output. No substantial algorithmic changes were made.

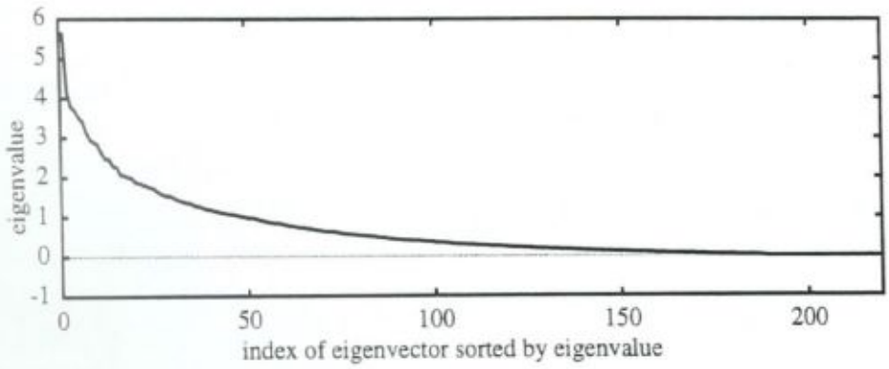


Figure 2: Eigenvalues for the phoneme /iy/, sorted by size. Since they fall off smoothly, there is no obvious choice of the dimensionality at which the important variation in the phone has been accounted for. A projection onto eight eigenvectors was chosen for the phoneme models, to keep the representation reasonably compact.

make the dimension of these PPC vectors eight was taken rather arbitrarily; there are 189 eigenvectors[7] resulting from the analysis and any set of the first n of them, ranked by eigenvalue, could have been chosen. As Figure 2 (for the phoneme /iy/) shows, the eigenvalues fall off quite smoothly, offering no strong guide to where the useful variation in the phone has been accounted for in a projection. The chief reason for the decision to use only eight dimensions to represent each phoneme was a desire to limit the size of the input to the speaker modelling phase. Improvements in the time warping used could reduce a source of irrelevant variation in the phone vectors, and make the eigenvalues fall off more sharply.

Characteristics of the Phoneme Codes

To show what the PPCs look like, and to demonstrate that they are at least somewhat consistent with a speaker, PPCs were extracted for the phone /iy/ for the three male and two female training set speakers who used the phone at least six times.[8] Since these speakers used the phone different numbers of times, the number m of PPCs for each speaker differed. These PPCs were then divided into two groups for each training speaker, derived from the first $m/2$ utterances of /iy/ from the speaker, and the second $m/2$, respectively. The plots in Figure 3 are designed to give a graphical representation of first and second groups of PPCs from each speaker. Since these are two dimensional plots, only the first two components of the PPC are used. There is considerable variation amongst the PPCs from a given speaker, as might be expected, considering the number of effects on the PPC that could not be controlled for. Nevertheless, relationships among the speakers and position are largely preserved. Despite the variation in the pronunciation of the phone within each speaker, maximising the discriminant function has located the speakers within a subspace of speaker space for the phone, and it is this information that will be combined with similar information about other phones to form the speaker model. Plots for the other phones have similar characteristics.

7. The number of groups (i.e. speakers) less one.

8. The fact that there were only this many speakers with enough data to compute variances for halves of the phone instances is surely a sign that the TIMIT database is less than ideal for this sort of research.

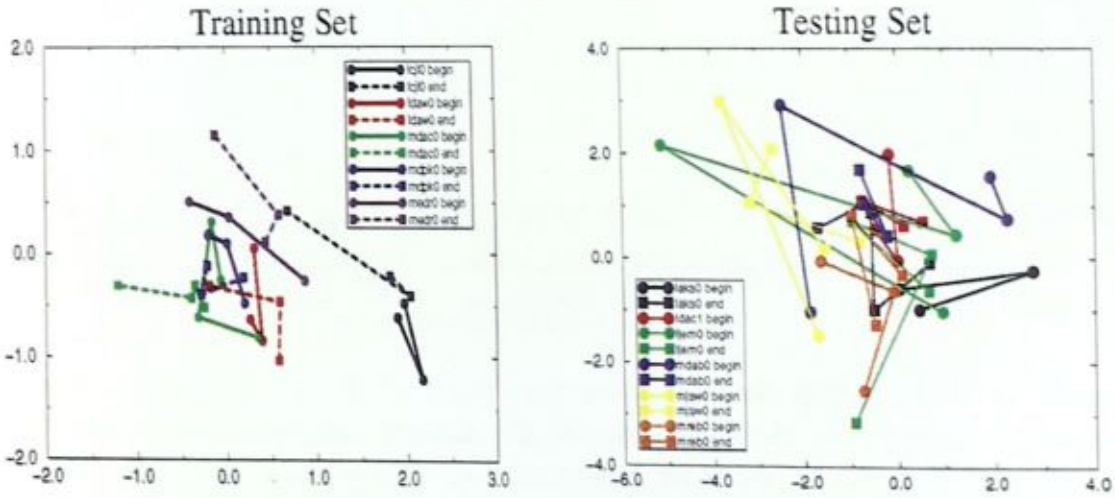


Figure 3: The first two components of the Models of /ix/ pronunciation for first and second half of the set /iy/ pronunciations available from each of a subset of speakers. While the speakers could not easily be identified on the basis of these phone codes alone, the codes from a given speaker are clearly located in nearby regions of the speaker space for /iy/. This position will be combined with evidence from other phones to form the speaker code. The within speaker variation of the speakers from the testing set is even greater, but there is still visible clustering of points by speakers.

Another measure of code quality.

Table 1 gives the values of a discriminant measure designed to measure the relative dis-

Phone	/dcl/	/ix/	/iy/	/k/	/kcl/	/l/	/n/	/r/	/s/	/tcl/
J	8.85	3.55	5.50	5.92	6.12	5.26	4.49	6.14	4.36	4.31

Table 1: Discriminant measure on PPCs for small example phone models. A description of the measure is given in the accompanying text; it is based on the relative separation of PPCs for different speakers when the ratio of this separation to that of PPCs within a speaker has been maximised.

persion of groups — in this case of speakers — in the transformed space that the CDA produces and in which the PPCs lie. This measure, the square root of the trace[9] of the ratio matrix referred to above, will be discussed in more detail in the next chapter;[10] for now it will suffice to note that larger values of the measure indicate better clustering of PPCs within a speaker. The PPC for the phone /iy/, for which phone models were shown in Fig. 3, is in about the middle of the range, the other phones provide a similar amount of information to distinguish speakers. It is also interesting and somewhat surprising to note that, as far as this model is concerned, consonants are as good for discriminating speakers as vowels are.

9. the trace of a matrix, the sum of its diagonal elements, is equal to the sum of its eigenvalues, its square root is similar to the diameter of a sphere with a similar volume to the space containing the projected vectors.
10. In section 3.7 on page 53.

2.3.3. Speaker model

Correlations between phone models

Now that the PPCs for the example have been found, it is possible to proceed to combining them into a speaker model. Before doing so, though, it is worth verifying that they have the qualities that are needed. The previous paragraph showed that the phone models locate a speaker within their particular space with some exploitable stability. If these PPCs are to be useful in a speaker model, however, they must bear relationships to each other that can be exploited to make predictions about unheard phones from heard ones. If it is possible to predict the PPC for one phone from that of another in a pairwise fashion, then it is reasonable to expect that an underlying variable, the SVC, can be found that enable one to make all such predictions simultaneously. To demonstrate this prediction, mean PPCs for each phoneme were calculated for each speaker,[11] and canonical correlation analysis [becker88,manley86] applied between pairs of these means, across speakers. This analysis finds a set of pairs of

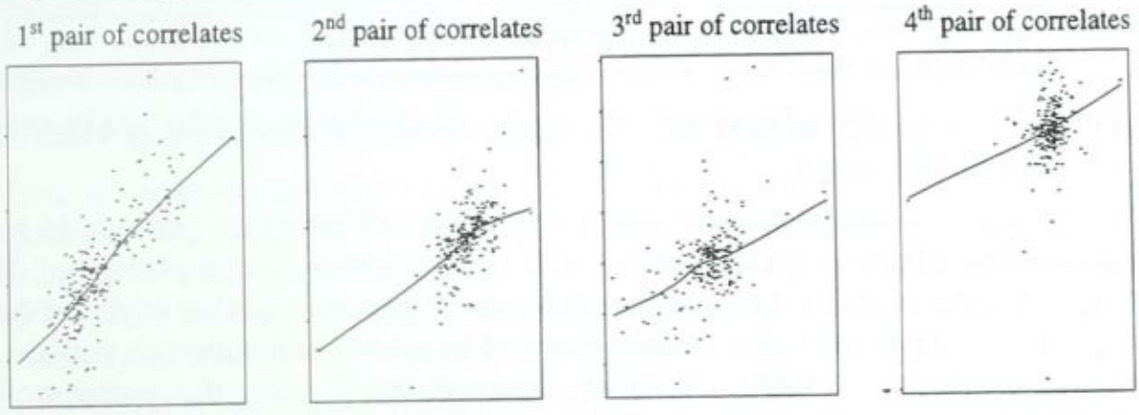


Figure 4: Scatter plots of the first four canonical correlates between /ix/ and /iy/ for the training speech. These correlates are pairs of projections of the PPCs that have maximal correlation, such that successive such pairs are orthogonal. The projection for /ix/ is given on the x-axis, and for /iy/ on the y-axis. The lines through the data are locally linear fits using the loess method, and are included to give an impression of the degree of correlation and how linear the fit is.

linear combinations of the components of the PPCs, such that the first pair has the highest possible correlation, the second has the highest correlation among variables uncorrelated with the first pair, and so on. Figure 4 shows the first four of these eight pairs, for the phones /ix/ and /iy/ from the training set plotted with the appropriate projection for /ix/ on the ordinate, and for /iy/ on the abscissa[12]. The values of the correlation coefficients, r, for these four pairs are r_1=0.798, r_2=0.636, r_3=0.486 and r_4=0.287 respectively. Using the Bartlett test given below, from [manley86], it is possible to calculate the probability that each of these measured correlations r_j is greater than or equal to its given value, under the null hypothesis that there is no correlation between the /ix/ and /iy/ vectors (n is 190, the number of speakers, and p, q, and r, the width of the /ix/, /iy/ and correlate vectors, respectively, are equal to

11. overall means for the phoneme were used when the speaker did not utter a phone.

12. The graph was produced using the SPlus scatter.smooth function. Interested readers are referred to the documentation for that program, and to [chambers93].

$$\left(\phi_j^2 = -\left(n - \frac{1}{2}(p+q+1)\right)\sum_{i=j}^{r} \ln(1-r_i^2)\right) \sim \chi^2(p-j, q-j)$$

eight). The results of this analysis are laid out in Table 2; the first three correlates are highly

Table 2: Significance tests for the first four canonical correlates between the /ix/ and /iy/ vectors across speakers using test due to Bartlett. The first three correlations are highly significant, and are almost certainly not due to chance.

index (j)	correlation	ϕ_j^2	$pr(\chi^2(p-j, q-j) \geq \phi_j^2)$
1	0.798	363.45	<< 0.0001
2	0.636	179.37	<< 0.0001
3	0.486	85.42	<< 0.0001
4	0.287	36.497	> 0.06

significant, suggesting that there are at least three orthogonal dimensions in which the PPCs for /ix/ and /iy/ are related.

The assumption behind speaker modelling on basis of segmental pronunciation — the approach taken here — is that there are similar relations, although perhaps of different strength or dimensionality, between the other pairs of phones a speaker might produce, and that, moreover, these correlations can be captured by a projection (although, perhaps, a non-linear projection) onto a single underlying vector variable. Building the speaker model consists of finding this projection of the phone pronunciation codes described above onto the single underlying variable, or Speaker Voice Code (SVC).

Training the speaker model

In this example of speaker modelling, instead of having the system explicitly model the correlations by trying to produce predictions of the values of PPCs from a subset of them,[13] it is assumed that the predictions made by the PPCs are useful only in as much as that they serve to distinguish speakers. The SVC will be generated using a non-linear projection of the PPCs onto a single vector. The projection will be found by optimisation performed with the goal of learning this speaker distinction. The model of speaker variation, in this case, consists of the neural net shown in Figure 5. This network attempts to do a non-linear discrimination between speakers, using the information available from a subset of the ten chosen phones.

During training, phones codes were randomly chosen from those available for the target training speaker so that, on average, 5.75 of the ten sets of eight-component phone inputs had PPC data on them. Inputs for which a PPC had not been chosen were set the overall mean of the appropriate PPC. A variety of widths was used for the bottleneck layer in which the speaker model was formed, producing a set of speaker space models of differing dimen-

13. Although that too will be done, in a later chapter.

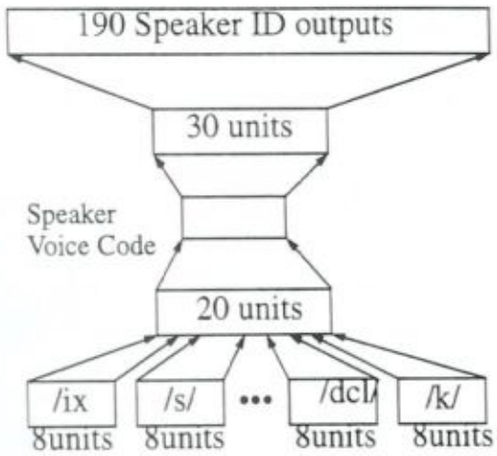


Figure 5: A connectionist discriminator that can be trained to form a low-dimensional speaker model. Phone codes from a speaker are placed on the appropriate sets of input units to the network, and the network is trained to output his or her identity. In doing so, the network produces a vector of hidden-unit activities in a bottleneck layer, and these activities can be used a speaker code. The purpose of the network is to form the SVC in the bottleneck layer.

sion. Training was carried out with a learning rate of 0.001 and momentum of 0.5 for a thousand epochs. At the completion of training, for the network with a two unit bottleneck, approximately 3% of the generated training input patterns resulted in correct speaker identification. It should be noted that this is an extremely difficult task, there is a great deal of noise in the input patterns, and to make it easy to plot, the network is trying to encode the speaker information into a very narrow, two unit bottleneck. Despite this, its classification performance is almost six times greater than the rate that would be expected by chance.

After training, the SVCs were generated (again for the training speakers in this example) by using input in the order that it had appeared in the original speech. For each speaker, all of the banks of eight phone code units were initialised with zeros. As phone codes were read from the file, they were placed on the appropriate input bank, replacing whichever value had previously been there.

The hidden unit activities, constituting the SVC are plotted in Figure 6 for the two unit bottleneck. SVCs for four female and four male speakers are shown at four points in time: after each of the first 15 phones had been placed on the input, after phones 11-25, phones 21 to 35, and after phones 31-45. Over time, the speaker codes move towards final position in the speaker space, and increase their within speaker clustering and between speaker separation. Not all speakers have easily identified positions; speaker *fjcf0*, for example, appears to vacillate between two positions. A more detailed analysis of the speaker codes, including their degree of clustering, and the time course of their formation, is deferred to chapter 4.

2.4. Summary

The example illustrates that, given an adequate division and classification of speech into segments, it is possible to construct models of the segments that are consistent within speakers, that differ between speakers, and that predict each other's values. To take advantage of

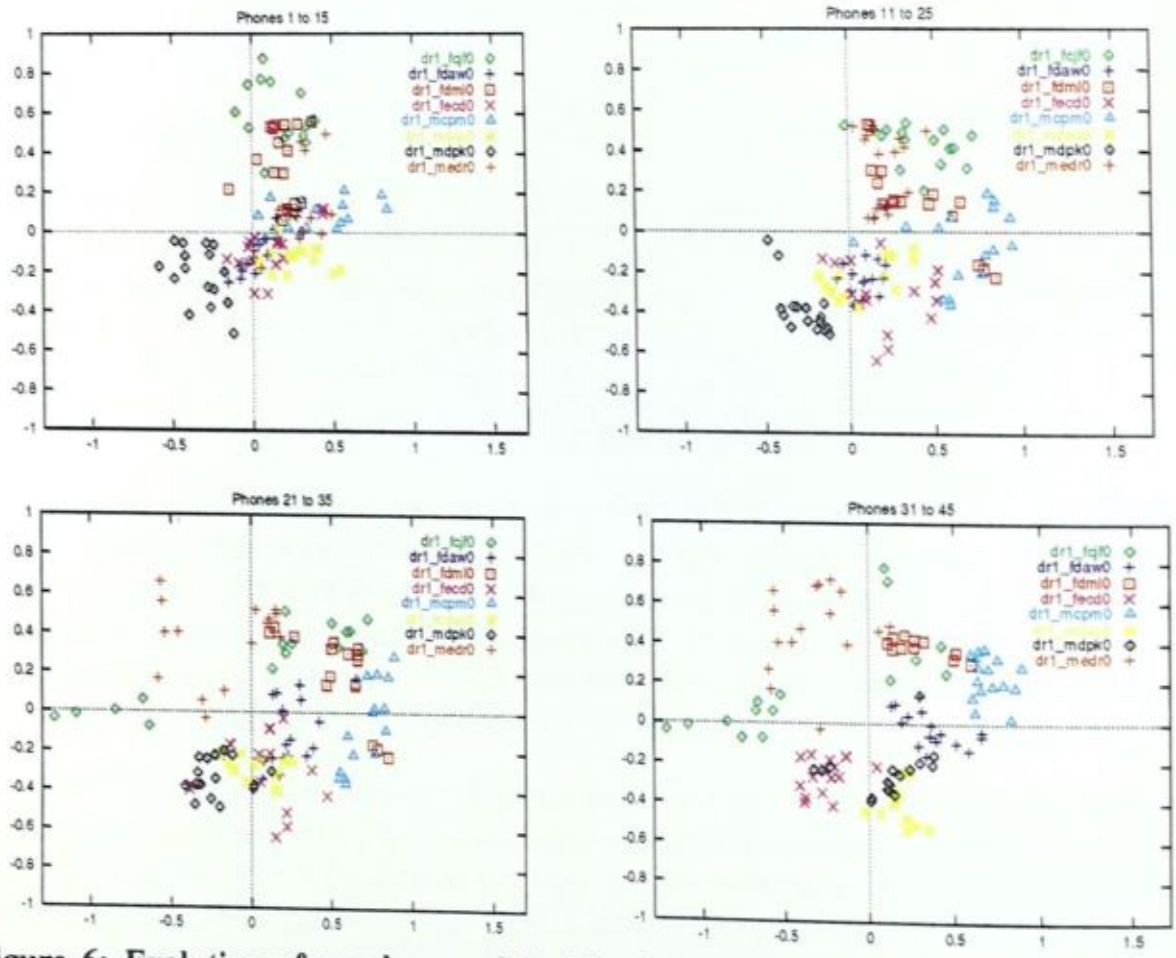


Figure 6: Evolution of speaker models. The four graphs represent speaker codes formed after the addition of phones 1-15, 11-25, 21-35 and 31-45 of those spoken by the first four female and male training speakers. While there is considerable variation within speakers, and overlap between speakers, there is a clear distinction by sex, and clear differentiation between speakers. Over time, the codes become more separated between speakers and more tightly clustered within speakers.

this ability to predict the PPC of one segment from that of another, speaker models can be built, designed to capture the relations between segmental pronunciations. The representation that one of these speaker models forms of the set of segments it has heard from a speaker, at some given time, constitutes at least a partial characterisation of the speaker's voice — a Speaker Voice Code.

Of course, there is more than one way to derive a PPC from a speech signal, and more than one way to combine these PPCs in an SVC. The following two chapters will explore these alternatives for the PPC and SVC respectively, after which two speech tasks will be used to examine the utility of this partial characterisation of voice quality.

Chapter 3. Variation within Phones

The previous chapter contained an overview of the form the speaker models would take, and illustrated that form with a particular instance of such a model. The next two chapters will describe possible variants of that general form in detail, and describe the experimental work that was done to select among them. The current chapter will concentrate on the construction of models of the variation in individual speech units, and the following chapter will cover the combination of the outputs of these segmental models into overall representations of a speaker's voice.

3.1. Database

Good data for speaker modelling work is not easy to obtain. Since it is desirable to attain reasonably good coverage of the space of speakers, speech samples are needed from a large number of people. Since speaker models are being built up out of models of the variability within phones, it is necessary to estimate both how pronunciation of each phone varies between speakers, and how it varies within a speaker. To accurately estimate the latter, within speaker variability, it would be useful to have available an amount of speech from each speaker sufficient to contain several examples of each of the phones composing the model. Unfortunately, it was not practical to gather such a huge, specialised database solely for the purpose of supporting this thesis work. Moreover, even with the considerable computational resources available to the CMU Neural Network Speech Group, many of the experiments described here would have been computationally infeasible if done on a larger database.

With these constraints in mind, a subset of the data in the TIMIT (Texas Instruments/Massachusetts Institute of Technology) database [fisher86,lamel86] was chosen as the training and test set for the speaker and phone models. This database contains recordings of 6 300 sentences, ten sentences uttered by each of 630 speakers from eight dialect regions in the United States. Because of computational and storage constraints, the experiments in this thesis used data only from speakers who had been raised in the first three of these regions, labelled dr_1 (New England), dr_2 (Northern) and dr_3 (North Midland) in the database. These groups contained, respectively, 49, 102 and 102 speakers, of whom 18, 31 and 23 were women. Of the ten sentences spoken by each speaker, two were "dialect sentences" designed to highlight dialectical variation, these sentences, sa_1 and sa_2 were identical for all speakers. Three of the sentences were "diverse" sentences, selected from two existing corpora with the aim of maximising the range of "allophonic" contexts of the phonemes used. These sentences (si_n) were different for every speaker. The remaining five "compact" sentences (sx_n) from each speaker were each spoken by a total of seven speakers, and were designed to give good diphone coverage, with a concentration on contexts thought to be difficult, or of particular interest, by the database designers.

The material on the database CDROM is divided into training and test directories, and this division was used in the reported experiments,[1] rather that the division suggested in the doc-

1. This was done merely for convenience; the recommended division should be used for further experiments.

umentation contained on the disc. During training, the $sa_{1,2}$ sentences were also excluded, so that they could be used as reference material in a comparison of the results of voice conversions performed using speaker models from different speakers applied to the same text.

3.2. Symbol set

If every speaker could be constrained to utter the same utterance, over the same duration, the voice modelling task would be relatively straightforward. A fixed set of these utterances could be collected from a great many speakers, and a model of the variation in them, of chosen dimension, could be estimated by a technique such as principal components analysis or compression in a bottleneck network.

In fact, for rapid, natural adaptation, no constraints can be placed on the speech uttered. Instead, a set of speech units must be chosen within which voice variation can be modelled, and into which the speech can be divided for analysis. To the extent possible, the realisation of these units should be constant within a speaker but vary between speakers. Additionally, the units must occur sufficiently frequently in speech to make them useful for modelling; learning to extract information from a unit of speech is of little use to a system if that unit is almost never used by speakers. This choice of a set of symbols for modelling depends on satisfying two, mutually antagonistic goals:

1. It is desirable to minimise the number of symbols used to describe the meaning of the speech signal, so that there will be enough samples of each symbol available to provide a reasonably dense coverage of the space in which it varies. This is essential to producing a model that makes useful predictions, since otherwise it will not be possible to obtain reliable estimates for the parameters of a phone model
2. Since the speech associated with each of the symbols is to be used only to model speaker differences, it is desirable to minimise the amount of the variation in the instances of each symbol that is unrelated to speaker characteristics. In particular, it is necessary to minimise the influence of phonetic and contextual variation.

A natural unit to choose as a symbol is the phoneme, since it provides a balance between frequency and consistency within speakers. It is also attractive, since it is generally the unit of recognition or synthesis. Information extracted by modelling differences in its pronunciation is likely to be useful to a phoneme based recognition or synthesis task, and the phoneme units required should be reasonably easy to extract from utterances, since the identification of these units is often a component of the target application.

Of course, there are disadvantages in the choice of phonemes as a basis of modelling — chiefly the amount of acoustic variation in phones due to context. Since much of this variation is due to immediate phonetic context, a large source of non-speaker-related variation could be eliminated if it were possible to use triphones[2] as the modelling unit. Unfortunately the number of triphones is so large, and the available resources of training data, computation and storage space so limited, that this sort of modelling is not tractable. Although the technique was not applied to the main body of experimental work reported here, an initial

2. Triphones, also called PICs (phones in context) are units consisting of the realisation of a phoneme in the context of a particular preceding and succeeding phoneme.

approach to the problem to reducing allophone variation, without modelling triphones separately, was made, and will be described in §3.11.

In most of the experiments reported, the only source of phonetic variation that was controlled for was the difference between phones, and this control was achieved by separately modelling variation within each of a set of phones. Since uncommon phones are, in general, unlikely to be particularly productive in predicting future speech, since they are not usually available, and since this scarcity also makes it hard to get reliable estimates for the parameters on which they vary, infrequent phones were not modelled. The choice of which phones to use and which to omit was made with the assistance of the data shown in Figure 7, a graph of the frequencies of phones used by the speakers in the first three geographic regions (dr1, dr2 and dr3) of the timit training set, sorted by frequency. The great majority of the

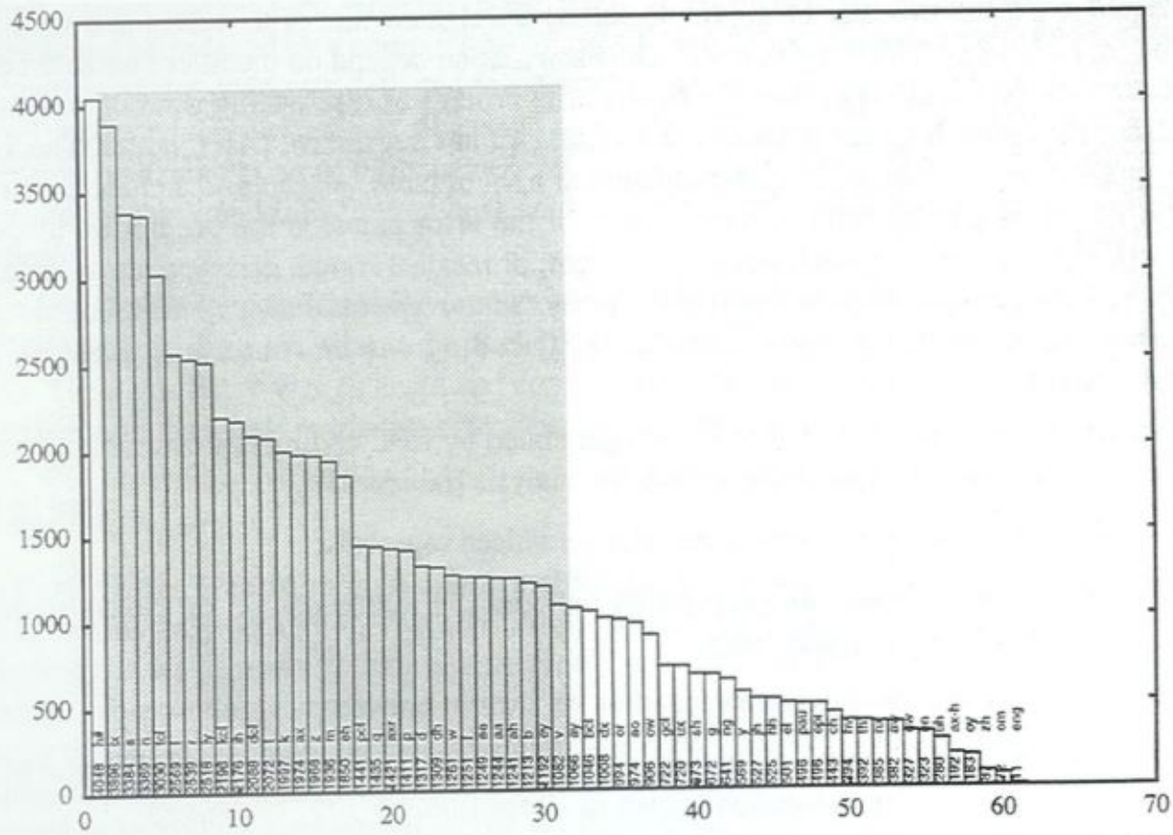


Figure 7: Frequency distribution for phones in speech from speakers frm the TIMIT $dr_{1,2,3}$ regions, excluding *sa* sentences, for all speakers. The shaded region indicates the thirty frequent phones used for the main experiments. These account for 78.3% of the phone occurrences.

phones in the database are covered by the first thirty of these phones, and it is these phones that were used to build models.

3.3. Analysis method

Once the set of symbols had been chosen, a representation of the raw speech signal was selected for use as a starting point for building the phone models. The two candidate representations were suggested by the target applications. Speech recognition systems typically

use a spectral representation of the speech signal, mimicking the signal analysis done by the basilar membrane in the human ear, so it was natural to consider building the model with FFT filterbank coefficients. Voice Transformation is performed, in the system described later in this thesis, in a representation consisting mainly of LPC log area ratio coefficients, which are related to a speaker's vocal tract dimensions, so it was also natural to consider using that representation for building speaker models. Both these representations are briefly described below, followed by an experiment that was done to compare their suitability for voice modelling.

3.3.1. LPC log area ratios

LPC (linear prediction coefficient) coding is derived from the observation that the speech signal at a given time can be approximated by a weighted sum of its values at a small number of past times. The weights used in this summation depend on the filter characteristics of the vocal tract, and vary relatively slowly. The process of discovering a set of weights that describe a speech signal is known as building an autoregressive (AR) model. The speech signal can be represented by a combination of a set of these weights and a crude approximation, such as a pulse train or white noise, of the error between the prediction of the AR model and the actual speech signal. This error, or residual signal, corresponds to the excitation signal generated by the vocal cords or by the movement of air past obstructions in the throat and mouth. Far more detail on LPC coding can be found in [rabiner93] and [markel76].

The compact representation of speech generated by LPC coding has desirable properties as a representation of speech for speech for analysis [rabiner93]:

- It models the speech well, especially for voiced segments.
- It provides a reasonable separation between the representation of glottal source and vocal tract characteristics.
- It is computationally tractable, and
- It has tended to work well in recognition applications, usually after conversion to a quasi-spectral representation known as a cepstrum.

The LPC representation also has the very desirable property, from the point of view of speech synthesis, that it is very straightforward to reconstruct a good quality speech signal from the LPC representation

The raw LPC representation of speech is not ideal for speaker modelling purposes. These models require learning to place speech from different speakers at different points within a space. During training, the modelled position and the desired position of the speech in this space must be compared using a distance metric. With raw LPC "reflection" coefficients it is not appropriate to use the sort of Euclidean distance measures appropriate for neural network training [rabiner93]. Fortunately, it is straightforward to convert the reflection coefficients generated by LPC analysis into a more appropriate representation.

The vocal tract can be viewed, approximately, as a concatenation of p fixed length cylindrical sections of cross-sectional areas $A_i, (i = 1, 2, ..., p)$. Starting from reflection coeffi-

cients, one can calculate coefficients that are each the log of the ratio between the areas of successive cylindrical sections, starting from the lips, i.e. $l_i = \log(A_{i+1}/A_i)$. These Log Area Ratio (LAR) coefficients have two desirable qualities from the point of view of speaker modelling: sensible Euclidean distances can be calculated on them [rabiner93], and they represent directly a fairly good [kuc87] approximation to the vocal tract shape differences between speakers that the models are trying to capture.

While linear predictive encoding is conceptually straightforward, producing a reliable LPC encoder/decoder is not an entirely simple matter. Despite some misgivings about its quality, a decision was made to use the freely distributable version of the government standard LPC-10 coder [tremain82]. This coder represents speech as a series of 22.5ms frames, each of which consists of a pitch value, a RMS power value, two boolean voicing decisions for half frames, and ten LPC reflection coefficients. In normal operation, bandwidth requirements would be reduced by a complicated bit encoding scheme detailed in the reference, but this section of the code was defeated for the experiments described in this thesis, and the frames produced were exactly as described above. For the majority of the experiments the reflection coefficients were transformed into the Log Area Ratio (LAR) representation before further use, and transformed back into reflection coefficients for the purposes of re-synthesis.

Readers intending to use the LPC encoder should note that it introduces a two frame delay in the speech stream, and that it drops the last two frames, making the delay difficult to detect. Until this delay was detected and compensated for[3], aligning the frames the encoder produced with the labels in the timit database had not been successful.

3.3.2. FFT

The Fast Fourier Transform is an efficient algorithm for computing the Fourier decomposition of a time varying signal. The algorithm, which is explained in detail in many places including the well known "Numerical Recipes" [press88], takes an array of floating point numbers representing a signal and returns an array of complex numbers representing that specify the phase and amplitude of a set of sine waves. These sine waves, at equally spaced frequencies between 0 Hz and half the sampling frequency of the original signal, can be summed to reproduce the original signal, or, more importantly for the present purpose, the amplitude at each of these frequencies can be extracted to provide the power spectrum of the signal. This representation, which is computed in approximation both by the basilar membrane of the inner ear, and, it seems, by the auditory cortex, displays useful information about a speech signal. Notably, this information includes the spectral peaks, or formants, in the signal resulting from vocal tract resonances controlled by the position of articulators during vowel production, and the shape of the filter applied to the noise of turbulent airflow during consonant production.

Since in speech processing, one is interested in the time course of the speech signal, and not just its overall spectral characteristics, the FFT is computed on fixed duration "windows" on the speech signal. The duration of the windows is chosen to be short enough to

3. By finding a minimum in the alignment error between encoded and unencoded speech, a process that the author feels compelled to mention not because it is particularly interesting, but simply because of the effort it involved.

capture important changes in the speech signal while still providing enough frequency resolution too reveal important spectral peaks and troughs. Typically, and in the work reported here, the frames contain 128 samples from a signal sampled at 16kHz, giving a time resolution of 8ms and a frequency resolution of 125Hz. The resulting series of power spectrum vectors are referred to as "frames" of speech.

It is common practice to smooth the signal by using overlapping windows, applying a windowing function to the signal to reduce boundary effects, and by reducing the dimension of the frames by combining contiguous frequencies into a smaller set of logarithmically spaced bins. In the work reported here, however, the power spectrum of the raw signal was typically used directly.

Figure 11 shows an example of an spectrogram computed from the utterance sa_1, *"She had your dark suit in greasy wash-water all year."*, spoken by a male speaker (mpgh0) from the New England (dr_1) region of the TIMIT data-set.

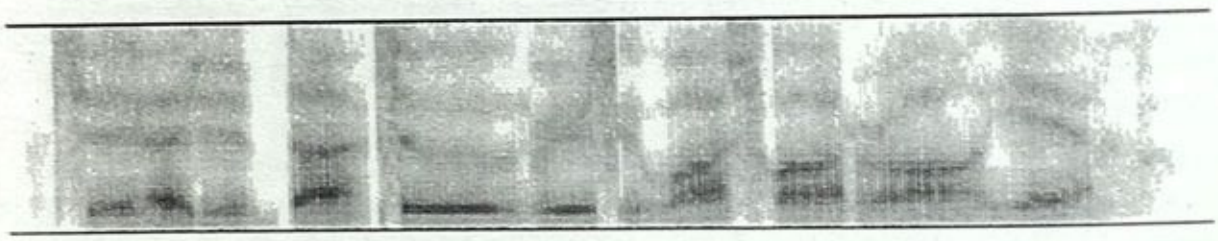


Figure 8: An example spectrogram generated using the Fast Fourier Transform (FFT). The sentence *"She had your dark suit in greasy wash-water all year."* was spoken by a male speaker.

3.4. Experiment: Choice of Analysis Method

Although both the LPC and FFT methods of extracting spectral information had been widely used in speech research, it was not known which, if either, was a better representation to use for modelling speaker differences. In this experiment, phone models built using both were compared for speaker discriminability.

3.4.1. Materials

The models compared were produced from speech from the ten phones /dcl/, /ix/, /iy/, /k/, /kcl/, /l/, /n/, /r/, /s/ and /tcl/, represented using both LPC LAR coefficients and FFT filterbank coefficients. In each case, the model was "trained" on the '*sx*' sentences for "training" speakers from dr_1, dr_2 and dr_3, and "tested" on the '*si*' train, '*si*' test, and '*sx*' test sentences for speakers from the same region. This testing data included speech from the same speakers used to develop the phone model, and from different speakers, to permit measurement of the degree of overfitting, if any, in the model.

In the case of LPC coefficients, the phone boundaries available in the TIMIT database were used to excerpt frames directly from precomputed LPC-LAR versions of the speech files. The LAR coefficients were used directly, and the two voicing decisions were multiplied by 1.0, and the pitch and gain by 0.01 and 0.005 respectively, to convert them to float-

ing point numbers. The frames making up each instance of a phone were then warped to a fixed length of ten frames using the linear warp described below, yielding a 140 element vector per phone.

If the case of FFT coefficients, the raw speech from the TIMIT database was read in, and converted to floating point values in the range [-1:1]. The speech for the target phone was excerpted, and FFTs calculated on non-overlapping 128 sample segments. The final segment was zero padded to 128 samples, if necessary. The FFT analysis yielded frames of 64 power values for each segment. The frames for each phone instance were linearly warped to a fixed set of five frames, yielding a 320 element vector per phone.

3.4.2. Procedure

All vectors from both encodings and for the four data-sets (sx_train, si_train, sx_test, si_test) for each phone were projected on to the first eight principal components for the training data, yielding eight unit phone codes. The discriminability measure J, described in section 3.7, was calculated for each phone and for each of the eight conditions.

3.4.3. Results

Table 3 shows the mean value across phones, and the variance, of the discriminant measure for the two representations, for both training and test speakers. This measure describes how closely clustered phone codes within a speaker are, compared to the spread of the phone codes between speakers. In all conditions the LPC-LAR representation produced slightly more tightly clustered phone models than the FFT representation, but the difference was clearly not statistically significant, since the differences between means are close to a single standard deviation.

Table 3: Discrimination measure J calculated for two candidate input representations to phone modelling. The phone models in question are generated by PCA. The shaded entries are calculated from phone instances used in training.

Representation	Speakers	sx	si
		mean / s.d.	mean / s.d.
LPC-LAR	train	2.05 / 0.43	2.36 / 0.40
	test	2.06 / 0.35	2.35 / 0.35
FFT	train	1.76 / 0.27	2.17 / 0.39
	test	1.79 / 0.35	2.28 / 0.46

3.4.4. Conclusion

There was no strong reason to choose one representation over another, although there was a nonsignificant tendency for the LPC to perform better. In the end, the FFT representation was chosen for use in further phone modelling experiments, in part because this representa-

tion is a more familiar one to workers in speech, and because the spectral representations it produces are more readable.

3.5. Fixed Length Phonemes

Phonemes present a problem as far as modelling is concerned. They occupy a variable duration. Moreover, the relative starting times within the phoneme of the acoustic states that make them up vary between different instances if the same phoneme. If the aim were simply to try to model vocal tract characteristics, this temporal variability would need to be eliminated, as far as possible, from the modelling process. For the most part, in this thesis, it is acceptable to accommodate this variability, since the dynamic aspects of speech production are also important to voice quality. In this latter case, the linear warp described below suffices to put the speech into a form suitable for model building. However, in recognition of the fact that for some potential applications of speaker modelling, particularly applications to recognition, reducing this variation will help highlight relevant speaker differences, an algorithm for doing so will also be described.

3.5.1. Linear warping

The most straightforward method of fixing the length of FFT analysed phonemes for further analysis is the linear time warp. This technique is rather straightforward; if the length of

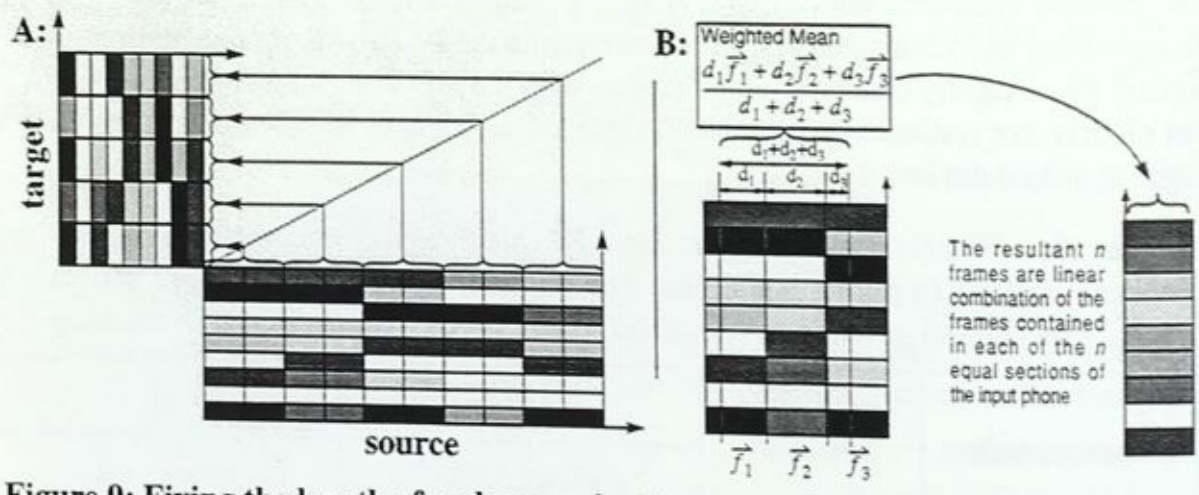


Figure 9: Fixing the length of a phoneme by linear warping. Part A of the diagramme shows the general scheme: Multiple frames in the source spectrum are copied onto single frames in the target in the case where the source length is a multiple of the target. Where this is not the case, target frames are linear combinations of source frames, as shown in part B of the diagramme. If the source spectrogram is shorter than the target duration, its frames are replicated until it is not, and the process shown above is applied.

the sound is *s* and the target length *t*, then the phone is divided into *t* sections *s/t* frames long. One can either take a "representative" frame from the middle of each section, in an attempt to reduce spectral smear, or produce the new frame by calculating the mean vector across the section, with frames from section borders being appropriately weighted. This process is described in figure 9. In cases where the input sound sample is shorter than the target dura-

tion, the length is doubled by frame replication until this is no longer the case, and then the linear warp is done. If the target length is sufficiently long, the speech can be recovered with reasonable fidelity after inverting the time warp, providing an invertible encoding, such as LPC, has been used to produce the starting vectors. In the case of FFT power spectra, this inversion is more difficult, but can still be done in approximation [Alex Waibel, personal communication].

This linear warping technique results in fixed length vectors, but retains information about timing within the phones. This information may not be particularly useful if used to construct phone models used, for example, to adapt a recogniser based on frame labelling.

3.5.2. Iterative alignment

Although this technique was not used in later experiments, in part because it is much more time consuming than linear warping, some pilot work was done to develop an algorithm to time align the excerpted phones used as input to the phone models. If an accurate recogniser were available, of course, one could eliminate temporal distortions by using the states from the alignment path generated during recognition. Speech information from a given recognition state would be inserted into the appropriate frame of the fixed length input vector.

In the absence of such a recogniser, an alternative method of identifying acoustic states within excerpted phonemes needs to be used. One such method is to reduce temporal distortions using a variation on the *k-means* clustering algorithm (described, for example, in [rabiner93]). For each speech sound, a fixed number of templates are used. After initialisation, involving assigning every *k*th instance of a phone to the kth template, an iterative procedure is used. Each example of the speech sound is aligned by dynamic time warping (DTW) alignment [ney84] against each template, and is assigned to the template with the best alignment score. After all samples have been treated in this manner, the frames in the templates are replaced by the mean of the frames in the samples of speech that aligned to them. This procedure is applied repeatedly until the total Euclidean distance between the input phones and the templates they align to reaches a stable minimum.

Table 4: These are the five, three frame long, template vectors for two vowel phones /IY/ and/EY/ and two consonants, /S/ and /K/ for the RMSpell database, at the end of the iterative time alignment procedure has been performed.

Vowels: /IY/ and /EY/		
Consonants /S/ and /K/		

3.5.3. Warping Chosen

The end result of either of these alignment procedures is a set of fixed length vectors capturing the acoustical features of the speech unit. These fixed length vectors can be used to build models of variation in pronunciation of the units they represent.

To accentuate the variation amongst instances of the same unit, the overall mean vector for the unit can be subtracted from each sample, and the resulting vector can be divided by the standard deviation, yielding vectors with zero mean and unit variance. This makes no difference to the statistical modelling techniques described later, but this sort of normalisation is widely believed to speed neural net learning, since there is no need to learn an offset for the output units.

Although the iterative warping method probably produces cleaner spectral estimates for sections of a phone, if the phone is thought of as consisting of a sequence of spectral states, it is not clear that this is the most desirable thing to do when building general models of voice variation. An important part of voice quality may be contained in the relative durations of these states within a phone, and the alignment would lose this information[4]. For this reason, and for practical reasons of computational load, a linear warping to five frames was used for the major experiments that will be described in the rest of this document. After this linear warping had been done, each phone was represented by a 320 element vector.

3.6. Capturing the Variation

After the length of the phone exemplar has been fixed, and any methods designed to correct for context effects have been applied, the next step is to build a representation of the variation in phone exemplars that is as parsimonious as possible. It is this representation that will be used as input to the system that combines the descriptions of variation in individual phonemes into an overall model of speaker variation. Four such dimensionality reduction techniques, two statistical and two using neural-networks, were investigated, with the aim being to retain as information as possible from the phone, in a representation of the lowest possible dimensionality. Figure 10 outlines the techniques.

	Variational	Discriminant
Linear	Principal Components Analysis	Canonical Discriminant Analysis
Non-Linear	Bottleneck Neural Net Encoder	Bottleneck Neural Net Discriminator

Figure 10: Methods for reducing the dimensionality of data. Variational methods select the directions of maximum variation in the input data. Discriminant methods choose directions of maximum variation relevant to a classification task.

The "variational techniques" involve forming a reduced dimension representation of a set of data that aims simply to retain as much information as possible about the variation in the

4. It might, however, have been the best choice or the actual test applications built, since in neither case was timing important to the system. However, improved voice transformers, for example, should adjust timing, and the model of voice should not discard this information without good reason.

data, whatever that variation might be. The linear variational technique, Principal Components Analysis (PCA) [james85, duda73, dennis91], finds a linear projection of the data onto a fixed, smaller, vector, that if inverted, most closely matches the original data set. Experiments with PCA are described in § 3.6.1. In the non-linear version of PCA [sarle94], a network with a bottleneck layer of the desired dimension is trained to match its outputs to its inputs. While it is not clear that there is any way to be certain that the encoding found is optimal, such networks can, as shown in § , exhibit impressive power when compared with linear methods. § 3.6.2 describes the application of these methods to the speaker modelling task.

Unlike the variational methods, which make no assumptions about the meaning of the data, discriminant methods assume that the data can be divided into relevant classes, and that, in fact, these classes are known at training time. These methods try to retain only variation that distinguishes the classes, and to discard that which distinguishes the members within each class. In the case of the experiments described here, the groups of observations to be distinguished are those coming from different speakers. The linear technique, Canonical Discriminant Analysis [james85, duda73, dennis91] (CDA)[5], does this by maximising the ratio of variation between groups to that within groups following the projection. Experiments using this technique are described in §3.6.3. The non-linear technique, again, uses a bottleneck network, but this time it is trained to label the speakers. Experiments with such a network are described in §3.6.4.

3.6.1. Statistical Dimensionality Reduction

If one is unbiased about which dimensions of variation in the original signal are important, principal components analysis gives the optimal linear projection of the original phone space onto a subspace of limited dimension. It retains as much of the information about the variability of the original space as can be retained in a linear subspace of the chosen dimension.

Since the phones are all same length at this stage, it is possible to use this standard technique to find a subspace describing their variation. This is done projecting the set of vectors on to an appropriately sized subset of the eigenvectors of their covariance matrix. By restricting this projection to the m eigenvectors with the largest eigenvalues, one chooses a set of m dimensions along which the original phone vectors vary the most. This is almost exactly what is wanted. This projection has the additional advantages of not being terribly expensive to compute and of being invertible. If the original analysis technique is invertible, as is the case with LPC coding, it is even possible to recover the speech, more or less accurately, by inverting the projection, reversing the time warp, and performing, for example, LPC synthesis on the resulting frames of estimated LPC coefficients.

3.6.2. Connectionist Dimensionality Reduction

One of the widely touted advantages of neural networks trained with backpropagation over other statistical methods is that they can learn non-linear transformations of their inputs. An example of such a transformation is the one implemented by the network in Fig-

5. Sometimes also known as Linear Discriminant Analysis (LDA).

ure 11. In principle, this ability gives neural nets far greater power than linear models

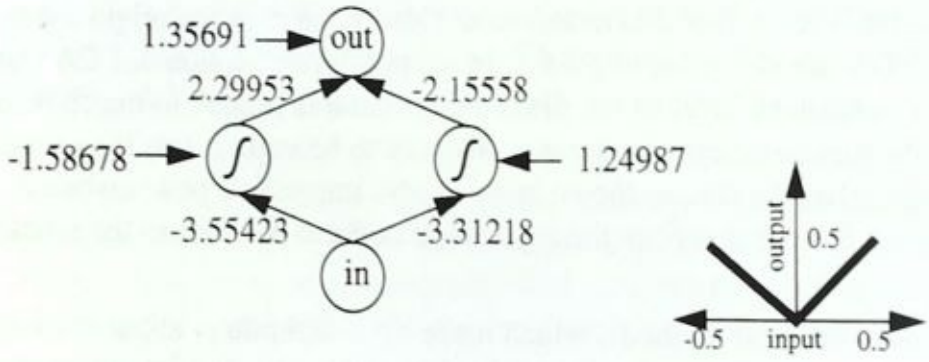


Figure 11: A trivial example of a nonlinear function learned by a neural network, $out = |in|$, or, more precisely {(-0.5,0.5), (0,0), (0.5,0.5)}. Weights are shown on the links connecting nodes, biases are the values next to each non-input node. The input-output fuction computed by this network is plotted at the right.

[minsky88], both as function approximators and as classifiers, if the appropriate weight settings can be learned. What these networks are doing, in effect, is multivariate multiple non-linear regression [sarle94]. It is the increased complexity in the regression function that the nonlinearity provides, that gives these networks the potential to more closely match the target values.

The Power of Bottleneck Networks

The idea of non-linear dimensionality reduction using neural networks is to use a network like that of Figure 11 to produce a non linear encoding of the input, and a second, similar network, concatenated with the first, to invert the encoding. The layer at which the encoding and decoding networks coincide is called a bottleneck[6], and the hidden unit activations (or outputs, if preferred) in this bottleneck layer constitute a reduced dimensionality representation of the inputs. Cottrell [cottrell90] used networks with a 64x64 grid of input and output units, representing greyscale pixels, and 80 (or fewer) hidden units in a single hidden layer to form reduced representations of face images. He then used these images as input to networks designed to extract features such as sex. The use of such networks is, however, somewhat controversial; in a very instructive paper, Boulard and Kamp [boulard88] showed that for "standard" three layer networks, using their one hidden layer as a bottleneck, the representations learned are at best equivalent to a subset of the principal components of the input. In fact, they showed the more general result that no matter how many layers the network has, *if* the bottle neck layer is the penultimate layer, then the representation learned can be no better that the optimal linear subspace found by principal components analysis[7].

There is a danger that this result will be seen as being more discouraging than it should be. One widely used connectionist text [hertz91], for example, seems to imply, although it does not state, that the Boulard and Kamp paper showed non-linear compression to be a hopeless

6. The first use of the term "bottleneck", or "goulot d'étranglement" in French, has been ascribed to Yann le Cun, but a suitable reference could not be located.

7. It is noted that this optimal linear subspace is, in itself, a rather good reduced representation for faces, and one that has been widely used in the literature [valentin94].

prospect for all networks trained with backpropagation. This is not the case; while the number of layers required may seem daunting, especially if one is not used to using networks with bypass (or short-cut) connections [lang88], networks with a layer (or layers) between the bottleneck and the output units can learn non-linear encodings of considerable complexity. In fact, DeMers and Cottrell [demers93] repeated the experiments in the earlier paper [cottrell90], compressing the first fifty principal components of the images to five dimensions using such a five layer autoassociative net[8].

To drive home the point that these networks can achieve better-than-linear performance, the two networks shown in Figures 12 and 13, each with a hidden layer following the bottle-

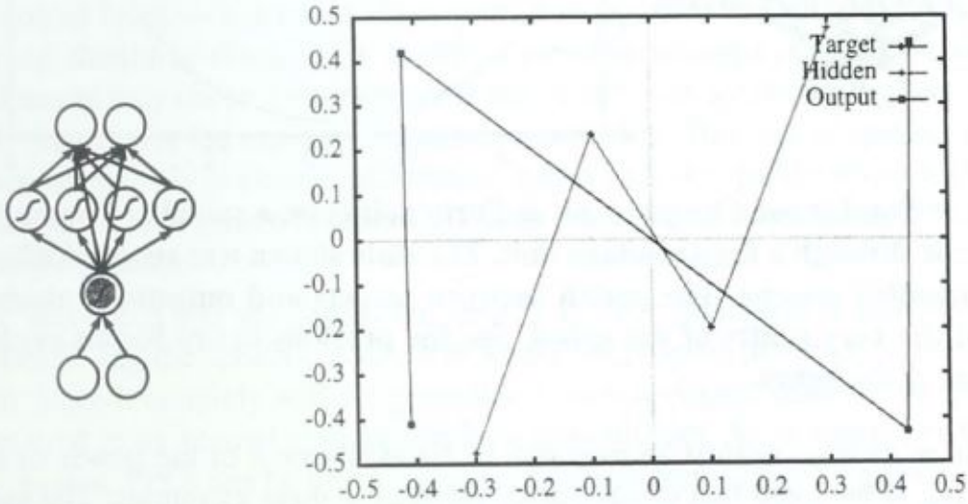


Figure 12: Non Linear Compression 1: four points on a square are passed without significant error through a single hidden unit. The state shown was reached after three hundred thousand training epochs. The outputs match the targets so well that their positions are indistinguishable on the graph. For the input and output activations, the dimensions of the graph correspond to the two input and output units, respectively. For the hidden unit activations, successive exampes of the four input patterns are spaced evenly across the *x* axis, and the y axis is the unit's output.

neck, were trained, using backpropagation with momentum, to encode increasingly complicated non-linear functions in a single dimension, by reproducing the input function on the output of a network with a single unit in its bottleneck layer.

The first figure (12) shows a net that was trained to implement one of the simplest non linear compressions possible. After training, four input points on the corners of a square in 2-space are encoded as four distinct points on a line by the single unit in the network's bottleneck. The four input tuples are reproduced almost exactly on the network's outputs after decoding in the second hidden layer.

The second figure (13) shows an example of a non-linear problem familiar from the connectionist literature [lang88] adapted to this context. A spiral, which goes through nearly 2 complete revolutions, is encoded, again, by a single hidden unit. It is perfectly evident, in this case, that thanks to the hidden layer that follows the bottleneck, it has been possible for

8. They did not, regrettably, compare the distortion of images compressed this way with those compressed by projection onto five principal components.

the representation in the bottleneck to be something much more complicated than a projection onto the first principal component of the training set.

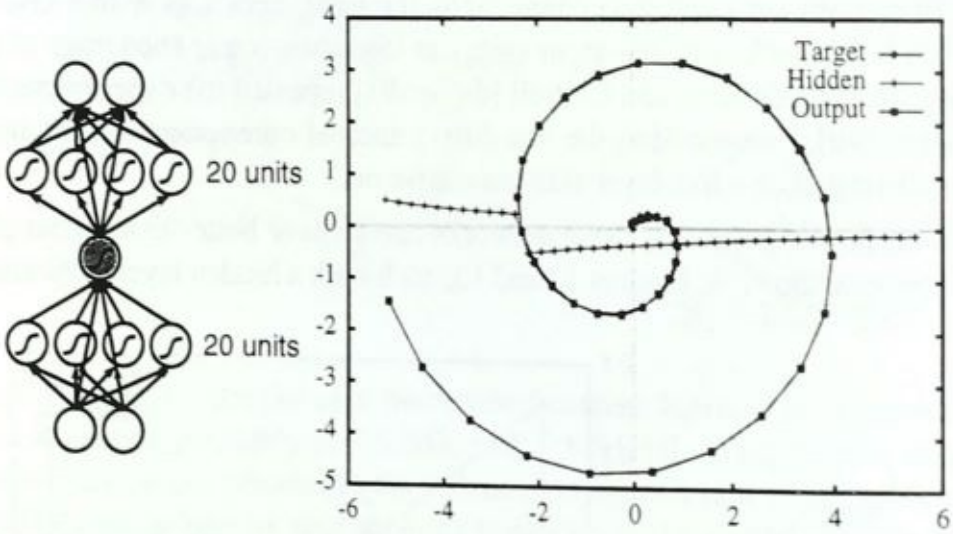


Figure 13: Non Linear Compression 2: Forty points on a spiral are passed with only slight error through a single hidden unit. The state shown was reached after fourteen million training epochs. The match between targets and outputs is almost perfect, except at the very centre of the spiral. See the previous figure for an explanation of the layout of the figure.

As a note of caution against an over optimistic assessment of the power of neural networks, it should be noted that it was not trivial to train these examples. The author is not confident that the examples given could have been learned without the use of the combination of short-cut connections outside the bottleneck, and comparatively large initial weights[9]. Even with these conditions satisfied, the networks took an extraordinarily large number of training passes[10] to move beyond a "linear" hidden representation, and thence to the performance shown.

Earlier in this chapter, models formed by projecting the vectors derived from each phone onto their first few (n) principal components were discussed. If, instead, bottleneck networks with the same number, n, of hidden units were trained on the same data, it is reasonable to hope that they would form a more compact representation that retains more information about the phones in the same sized representation. Given the difficulty of learning these non-linear representations, though, it is hard to predict how much of an advantage the representations formed in bottleneck networks will provide.

In the light of this discussion, the experiments reported below, which compare the two methods, are interesting in two separate respects. They serve both as a demonstration of extracting speaker information from phones, the ostensible and primary purpose, and as an experiment to compare the usefulness of PCA and bottleneck networks as dimensionality reduction tools on a real-world task.

9. It seems, perhaps, that the large initial weights (of the order of 2.0) prevent the network from too easily reaching local minima that involve setting the non-shortcut weights to zero, or making the weights symmetrical.
10. Fourteen million, in the case of the spiral problem.

3.6.3. Statistical Discrimination

Principal components analysis can reduce the dimensionality of speech segments by finding a linear projection of those segments that retains as much as possible of the variation in the segments, but in the lower dimensional space. Compression neural networks with a narrow bottleneck in one hidden layer can perform a similar reduction in the dimensionality of speech samples, either by finding an approximation of the principal components, or, if necessary, by learning a more complex non-linear encoding and a matching decoding.

Compressing the data in this way, however, is not necessarily the best choice if the goal is speaker modelling. While the projections will capture variation, they do not care where the variation comes from — there is no constraint that says that variation within a speaker is irrelevant and should be discarded in favour of variation amongst speakers. A purely variation based model may chose a representation that retains non-speaker dependent variation in the speech segments at the expense of speaker information. This loss of speaker information and addition of possibly irrelevant information may reduce the speed with which the speaker model reaches an appropriate location in speaker space, and the stability of that position within a single speaker.

Earlier it was noted that one degenerate form of speaker model would consist of a 1-from-n vector identifying the speaker. While one would not propose to use such a representation as a model, since it certainly will not generalise to new speakers, there are other techniques that can be used in an attempt to make speakers more distinct. As an alternative to the variational techniques, PPCs can be based on representations formed during attempts at classification. While the 1-from-n representation that classification learns for training speakers will not be used, it is to be hoped that the internal representations of such classifiers will separate these training speakers well, and that they will also distinguish novel speakers. It is these internal representation that will be used as PPCs.

The first of these classification methods is linear discriminant analysis (LDA). In this technique a linear transformation is learned that projects a database labelled by groups onto a linear subspace which maximises the ratio of between groups variation to within groups variation. In the speaker modelling application, of course, the technique is used to find the projection of the speech signal that maximises the distance between samples from different speakers, while minimising the distances between samples from a single speaker.

The projection matrix used in LDA is made up of a chosen number of the eigenvectors, with highest corresponding eigenvalues, of the ratio covariance matrix $\Sigma_B \Sigma_W^{-1}$, where Σ_B is the between groups covariance matrix (i.e. the covariance matrix of the group mean vectors), and Σ_W is the within groups covariance matrix (i.e. the covariance matrix of all the vectors, after the appropriate group means have been subtracted). Projecting onto these eigenvectors produces a set of vectors that maximise the amount of retained variability that is due to group differences in the original vectors, and minimises that due to within group variability.

3.6.4. Connectionist Discrimination

Perhaps the most popular use of connectionist networks is in pattern classification. If the data fall into n classes, the network is trained to produce a distinct output value on the one output unit, out of n, corresponding to the class of the input pattern. As was the case with the compression networks above, the activations of units in a hidden bottleneck layer can be used as a reduced dimensionality representation of the input, although not, this time, one that can depended upon to be invertible. In this case, these hidden unit vectors should be similar within speakers, and different between speakers, since they are the support for an output representation that, if training is successful, will be almost identical within speakers, and perfectly distinct between them. Webb and Lowe [webb90] show that, for a somewhat simpler network architecture using linear output units connected to the hidden layer, the network maximises a network discriminant function

While the same networks are employed, and they are still doing a sort of multivariate multiple non-linear regression, this use of neural networks differs somewhat from their use as a functional approximator described above. Different sorts of output error are acceptable in the two cases. While, in a function approximator, the output vector of the network should match the target vector as exactly as possible, indicating that training should be done with an error function that is proportional to the distance to be minimised, in classification, it does not matter what the output values are, exactly, as long as the target class's unit is the supremum of some chosen function. Usually one wants the target unit to have a higher value than all the non target units, but that is all that is required. John Hampshire [hampshire90] suggested an "error" function, called a classification figure of merit, based on this goal of minimising misclassifications. Using this error function, error is only backpropagated to the target unit, and to the non-target unit with the highest output. In some of the experiments reported here, an error function similar to Hampshire's CFM error measure[11] was used, as follows.

$$j = \underset{i}{\operatorname{argmax}}(t_i) \qquad k = \underset{i \neq j}{\operatorname{argmax}}(o_i)$$

$$cfm = \left(o_j - o_k + \frac{1}{2}\right) - \frac{1}{9}\log\left(1 + e^{9\left(o_j - o_k - \frac{1}{2}\right)}\right)$$

$$cfm' = \frac{1}{1 + e^{9\left(o_j - o_k - \frac{1}{2}\right)}}$$

The value cfm' is backpropagated into the highest non-target unit, k, and value $-cfm'$ into the target unit, j.

While this CFM does tend to speed convergence and decrease error on the training set, it isn't certain that it is best thing for speaker modelling purposes. While learning to classify the training speakers, the hope is that the network will produce a hidden unit representation that distinguishes speakers, while retaining the similarities that exist between similar speakers. CFM expends more of its effort than the usual squared difference measure in forcing

11. When the neural net code used here was being written, John Hampshire's function was, apparently, the subject of a patent application. For this reason, a function was independently produced that was intended to have similar properties.

similar speakers (the target, and its closest competitor) apart. While this will improve classification performance, it may decrease the utility of the hidden representation for speaker modelling purposes, by interferring with the aim of producing a topographic representation of speaker similarity in the representation space.

In fact, this is a general danger, not only with this particular measure, but with using discrimination of any sort to train the models. While discriminative training does provide an incentive for the model to concentrate on those features of the speech that distinguish speakers, there is a risk that the parts of the signal that best distinguish the training speakers do not include the features a human listener would consider important components of voice personality. It is to be hoped that the balance between these possibilities lies in favour of successful speaker modelling, and that the speakers will be separated as meaningful groups, before they are separated as individuals.

3.7. Measuring Performance

As various speaker models are developed, an objective measure under which they can be compared is necessary. An ideal such measure could have two forms. If the model is designed for a particular task, then, of course, the ideal performance measure is the change in a performance measure specific to that task, if, indeed, such a performance measure exists. If the model is intended to be task-neutral, the ideal speaker space would have the same topology as a human speaker space. That is, speakers perceived as being more similar by human listeners should lie closer together in this space. Unfortunately, obtaining similarity measures of any reliability from human subjects on a speaker set of any size would be a considerable research project in itself, and not one that could have been completed for this thesis.

Despite the *caveat* in the previous section against blindly separating speakers without regard for the larger speaker groups of which they are a part, a measure of model quality can be approximated by measuring the degree to which a candidate model distinguishes different speakers from each other.

Asoh [asoh90] used a discriminant criterion $J = \mathrm{tr}(\Sigma_B \Sigma_W^{-1})$, where Σ_B is the between groups covariance matrix (i.e. the covariance matrix of the group mean vectors), and Σ_W is the within groups covariance matrix (i.e. the covariance matrix of all the vectors, after the appropriate group means have been subtracted), so $\Sigma_B \Sigma_W^{-1}$, is the ratio covariance matrix used in canonical discriminant analysis. The function tr() is the trace function, the sum of diagonal elements.

However, the trace of a matrix is equal to the sum of the eigenvalues of the matrix and, following [freidman86], these eigenvalues can be regarded as mean squared length. The square root of the trace can consequently be regarded as an overall radius for the data in the discriminant space, and is a figure of merit for classifiability. Consequently, in comparing reduced data sets, the following measure will be used.

$$J = \sqrt{tr(\Sigma_B \Sigma_W^{-1})}$$

The values of J reported with experiments should be used to compare them with other similar experiments, rather than regarded as an absolute measure. Since J increases with increasing dispersion of the group means relative to the dispersion within groups, larger values of J suggest a better speaker or phone model, or, at least, a model that better serves to distinguish speakers.

After experiments comparing models of speaker variation in phones have been described in the following sections, the results of measurements of both J and a more direct measure of classifiability will be compared in §3.13 to discover how well the measure predicts classification accuracy.

3.8. Experiment: Comparing Dimension Reductions

Since speaker models based on PPCs derived by all four of the dimensionality reduction techniques could not practically be produced, an experiment was performed to compare these PPCs in isolation from the rest of the system. Although the measure J described above is not entirely satisfactory, it served as a practical objective measure on which a comparison could be based. Since the main danger with the measure is a loss of speaker groupings, the techniques were also compared with respect to their retention of the difference between men and women, since this is the only obvious voice personality grouping for which have class information was available in the database.[12]

3.8.1. Materials

As explained at the beginning of this chapter, phone models were trained using speech only from regions 1, 2 and 3 (New England, Northern, North Midland) of the TIMIT database. For training, the speakers from the "train" subset of the database were used, and for these speakers, only speech from the five "sx" and three "si" (phonetically compact[13] and diverse, respectively) sentences per speaker was used. There were 190 speakers in the training subset used, and a total of 950 utterances.

Before the dimension reduction techniques were applied, the speech was preprocessed as follows: All instances of each of the phones ("ix", "s", "n", "tcl", "l", "r", "iy", "kcl", "ih", "dcl", "t", "k", "ax", "z", "m", "eh", "pcl", "q", "axr", "p", "d", "dh", "w", "f", "ae", "aa", "ah", "b", "ey" and "v") were excerpted from the files of digitised speech using the phone label files provided. In order that the original phone ordering could be reimposed on the separated phones later, indexing information, specifying the start and end time of each phone, and the sentence from which it was excerpted, was retained for each phone. The excerpted speech was zero padded at the end to a multiple of 128 samples, and analysed using a FFT on 128 sample non-overlapping windows, yielding 64 filterbank power coefficients per 128

12. Geographical region, while it is distinguished in the database, has its main affect on accent, which the current work does not attempt to model. Pilot analyses looking for modelled differences between the regional groups did not produce positive results.
13. These sentences were designed to provide good coverage of pairs of phones, and to include extra occurrences of phonetic contexts thought to be difficult or of particular interest by the corpus designers.

sample frame. Frames inside phones were linearly warped to a constant five frame length as described earlier in this chapter, yielding a single 320 coefficient vector per phone. Speaker identity, regional identification, and phonetic context information was stored with these vectors. These 320 coefficient vectors extracted from each phone were subjected to the dimension reduction methods being compared.

3.8.2. Procedure

Although the input and the PPCs formed by each of the four techniques were identically represented as vectors of floating point numbers, the way they were processed differed somewhat. The method of construction for each of the models will be described before their performance is compared. In all cases, the set of vectors representing a single phone were modelled separately. For example, when the text describes the training of neural nets using phone vectors, it means that thirty such nets were trained, each of these nets being trained and tested only on the subset of vectors corresponding to a particular one of the thirty phones.

For each dimension reduction technique, models producing PPC vectors of length 1, 2, 3, 4, 5, 10 and 15 were trained. In total, 210 (seven PPC lengths by thirty phones) models were built for each modelling method.

Projection onto Principal Components

Using a version of the PCA program described in [dennis91] that had been slightly modified to allow it to handle larger vectors, eigenvectors of the covariance matrices for the phone vectors were calculated. These eigenvectors were sorted by decreasing eigenvalue, the eigenvectors with the n largest eigenvalues being the first n principal components. Projections of phone vectors onto the first n principal components were calculated, for the seven values of n listed above, and these were used as PPCs of dimension n.

Neural net compression

Phone instance vectors were compressed by neural networks with the five-layer topology shown in Figure 14. Bypass connections are present between every pair of layers, except

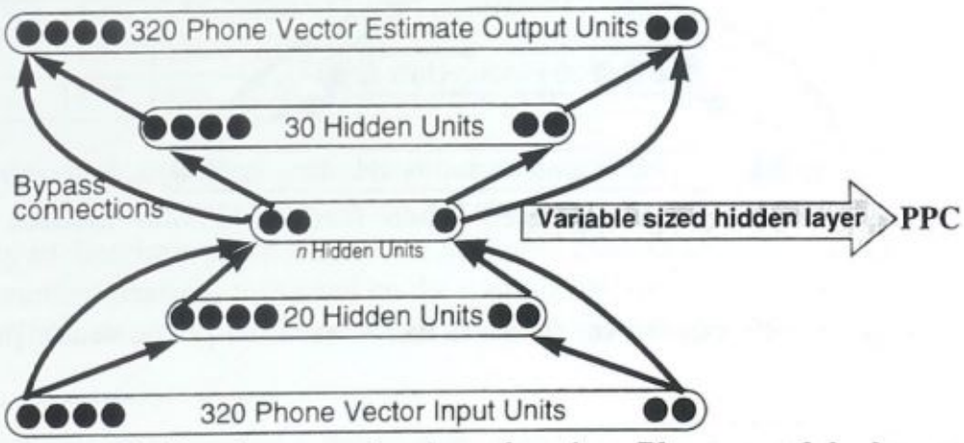


Figure 14: Network Topology used when forming Phone models by nonlinear compression.

where such connections would bypass the bottleneck in the second hidden layer. The number of units in the two fully hidden layers were chosen on the basis of a combination of the results of pilot experiments, the belief that the decoding task is more difficult than the encoding task, and the need to control the amount of computation required - there are probably many other choices for the size of these layers that would do just as well. During training, the same phone vector was used as input and target. These networks were therefore trained as constrained function approximators, where the function being approximated was the identity function. In learning to reproduce their inputs on their output units[14], by changing their weights to minimise the mean squared error between their outputs and the target, the networks had to form a representation of the input that could be contained in the outputs of the n units in the 2nd hidden layer (shown in red). These n hidden layer outputs were collected for each phone vector, and were used as PPCs.

Projection onto canonical discriminants

Using a version of a program (CDA) described in [dennis91, again slightly modified to allow larger vectors, eigenvectors of the between group/within group ratio covariance matrices[15] for the phone vectors were calculated. These eigenvectors were sorted by decreasing eigenvalue, the eigenvectors with the n largest eigenvalues being the first n canonical discriminants - the directions that maximally separated the speakers while keeping the utterances of a phone by a single speaker tightly clustered. Projections of phone vectors onto the first n canonical discriminants were calculated, for the usual values of n, and these were used as PPCs of dimension n.

Neural Net discriminator

Neural networks having the five-layer topology shown in Figure 14 were trained to iden-

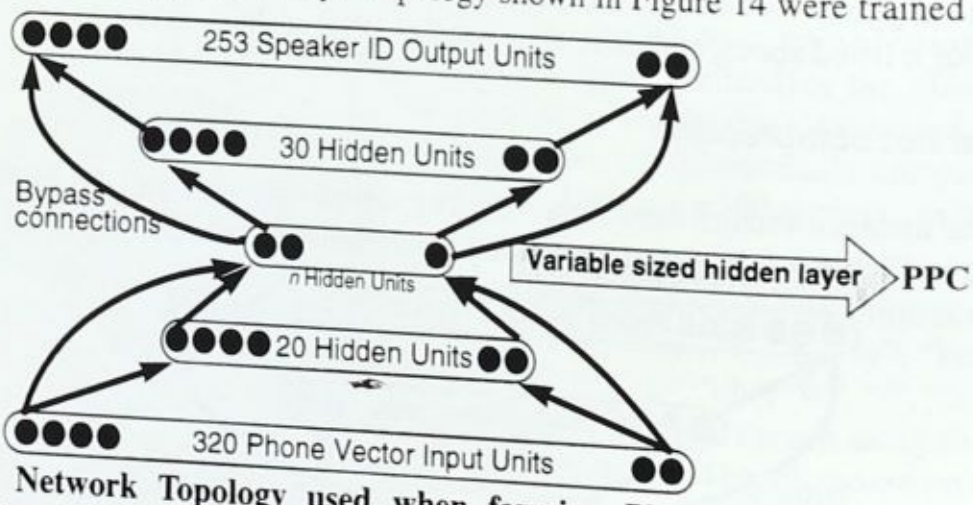


Figure 15: Network Topology used when forming Phone models by nonlinear discrimination.

tify which of the 189 training set speakers had uttered the phone vector presented to the

14. Training parameters for the networks are given in Table B-1 in Appendix B.
15. The reader may wish to refer back to the description of CDA given earlier in this chapter.

input[16]. Again, the vector of outputs of the units in the second hidden layer was used as the PPC for the phone instance whose vector had produced them.

To investigate the role of the training criterion in deciding speaker modelling performance, these experiments were done using both mean squared error and the previously described approximation to Hampshire's [hampshire90] CFM as the error function for the network during training.

3.8.3. Results

Once the PPCs for the thirty phones had been calculated for all four of the modelling methods, the corresponding speaker labels were used to permit calculation of the measure J for the PPCs. These values are a measure of the relative linear discriminability of the speakers based on the PPCs. There is a strong relationship between the measure J and actual discrimination scores that will be described later in this chapter. Full tables of the discriminant measure, calculated on training set data, by phone and PPC size for the four techniques, PCA, NNCompress, LDA and NNDA are given in Appendix D. as Tables D-1, D-2, D-3 and D-4 respectively. Tables 5 and 6 and summarize these results and similar results for test set data for consonants and vowels respectively, by giving the mean value of J across all PPC sizes.

Table 5: Discriminant measure (J) for the four types of PPC, averaged across all PPC sizes, for vowels. The discriminant models separate speakers better than the variational models. The clear advantage of LDA in separating speakers in the training set is lost for the test set, on which the neural net discriminator has slightly better performance.

	Method	Phone										Mean
		ix	iy	ih	ax	eh	axr	ae	aa	ah	ey	
Train	PCA	1.10	1.30	1.29	1.20	1.42	1.40	1.72	1.72	1.55	1.78	1.45
	NNCompress	1.05	1.24	1.16	1.15	1.31	1.36	1.53	1.58	1.45	1.61	1.34
	LDA	2.37	3.07	3.28	3.08	3.74	4.42	5.55	5.88	5.66	5.84	4.29
	NNDA	2.09	2.45	2.59	2.00	2.78	2.17	3.26	3.03	2.81	3.25	2.64
Test	PCA	1.05	1.34	1.18	1.15	1.42	1.11	1.71	1.59	1.53	1.80	1.39
	NNCompress	1.00	1.26	1.07	1.13	1.22	1.11	1.52	1.41	1.37	1.61	1.27
	LDA	1.57	1.69	1.81	1.22	1.76	1.10	2.09	1.48	1.45	1.96	1.61
	NNDA	1.80	1.99	2.00	1.51	2.16	1.39	2.62	2.02	2.04	2.47	2.00

Surprisingly enough, considering the importance that has been ascribed to vowels in determining voice personality, vowel models are not strongly favoured over those for consonants in their ability to discriminate speakers. For principal components, for example, the mean pooled discriminant measure, measured on the training set, for vowels was 1.45 and for consonants 1.34. Since the standard deviations (0.22 and 0.28 respectively) are > 0.2, this differ-

16. There are 253 output units for convenience in dealing with both training and testing speakers, to avoid the necessity relabelling the speakers. The 64 targets corresponding to test speakers were not used during training, and one would not expect them to be meaningful during testing.

Table 6: Discriminant measure for the four types of PPC, averaged across all PPC sizes, for consonants.

	Method	Phone																				Mean
		s	n	tcl	l	r	kcl	dcl	t	k	z	m	pcl	q	p	d	dh	w	f	b	v	
Train	PCA	1.88	1.42	0.78	0.95	1.02	1.04	1.14	1.47	1.29	1.84	1.63	1.20	1.17	1.49	1.38	1.32	1.31	1.60	1.34	1.54	1.34
	NNCompress	1.75	1.42	0.80	0.89	0.95	1.04	1.14	1.47	1.30	1.84	1.62	1.17	1.19	1.48	1.38	1.31	1.29	1.61	1.35	1.54	1.33
	LDA	2.92	2.82	1.96	2.61	2.71	2.75	2.85	3.08	3.02	3.58	3.77	4.74	3.97	4.35	4.21	5.00	4.84	5.64	9.91	6.63	4.07
	NNDA	2.65	2.27	0.90	1.93	1.91	1.11	1.32	2.34	1.98	2.64	2.30	0.97	1.93	1.97	2.00	1.80	1.93	1.86	1.75	1.88	1.87
Test	PCA	1.75	1.35	0.84	0.89	0.96	1.07	1.25	1.41	1.31	1.88	1.50	1.20	1.18	1.37	1.55	1.28	1.30	1.69	1.41	1.59	1.34
	NNCompress	1.66	1.36	0.85	0.84	0.90	1.09	1.25	1.41	1.31	1.89	1.49	1.16	1.16	1.39	1.57	1.29	1.27	1.70	1.41	1.57	1.33
	LDA	1.64	1.66	0.84	1.22	1.14	1.03	1.06	1.15	1.12	1.47	1.62	1.26	1.23	1.10	1.18	1.23	1.24	1.25	0.99	1.31	1.24
	NNDA	1.78	1.87	0.86	1.33	1.39	1.10	1.30	1.40	1.48	1.88	1.88	0.92	1.31	1.60	1.64	1.27	1.51	1.70	1.42	1.71	1.47

ence is not significant (t_{28} =1.03 P(equal means) < 0.31). Visual inspection suggests that across all techniques, there might be slight trend toward vowels distinguishing speakers more easily than consonants do, although statistical tests do not provide any compelling confirmation of this trend. Vowels from the training set, pooled across all techniques have J measures with a mean of 2.43 and s.d of 1.39. Consonants have a mean of 2.15 and s.d of 1.47. This difference is not significant (t_{118}=0.996, P(equal means) < 0.32).

The same measure of speaker discriminability for each PPC size, pooled across all thirty phones is given in Table 7 and plotted in Figure 16. The "variational" PCA and NNCom-

Table 7: Discriminant measure (J) for PPCs of various dimensions, for the four techniques. Each cell represents an average across all phones.

	Method	Width							Means
		1	2	3	4	5	10	15	
Train	PCA	0.6661	0.9520	1.1328	1.2575	1.4271	1.9218	2.2713	1.3755
	NNCompress	0.6394	0.9068	1.0804	1.2203	1.3732	1.8688	2.2378	1.3324
	LDA	2.3270	3.0750	3.5700	3.9689	4.3052	5.4825	6.2636	4.1417
	NNDA	1.1295	1.6258	1.8922	2.0941	2.2495	2.8063	3.1099	2.1296
Test	PCA	0.6390	0.9255	1.0989	1.2210	1.4077	1.9114	2.2829	1.3552
	NNCompress	0.6169	0.8912	1.0523	1.1968	1.3459	1.8500	2.2169	1.3074
	LDA	0.7190	0.9711	1.1491	1.2800	1.4095	1.8395	2.1721	1.3629
	NNLDA	0.8179	1.2218	1.4707	1.6145	1.7368	2.2069	2.4499	1.6455

press methods appear to be similarly effective, with the linear, PCA, method having slightly better performance. The linear discriminant model separated training speakers substantially better than the other methods. It has a significantly higher J measure than NNDA, the next best (t_{12} = 3.50, P(means equal) < 0.005). However, this very high value of the discrim-inability metric for LDA on training speakers is, perhaps, to be expected, since it is precisely this J measure that linear discriminant analysis attempts to maximise. On testing speakers the method performed less well - separating speakers no more than the variational methods did. For test data, the neural net discriminator did better, outperforming the linear and varia-tional methods, although the difference was not significant (t_{12}=0.9975, p(equal means) < 0.34).

The second trend, displayed in Figure 16, is that, as one would expect, the discriminability

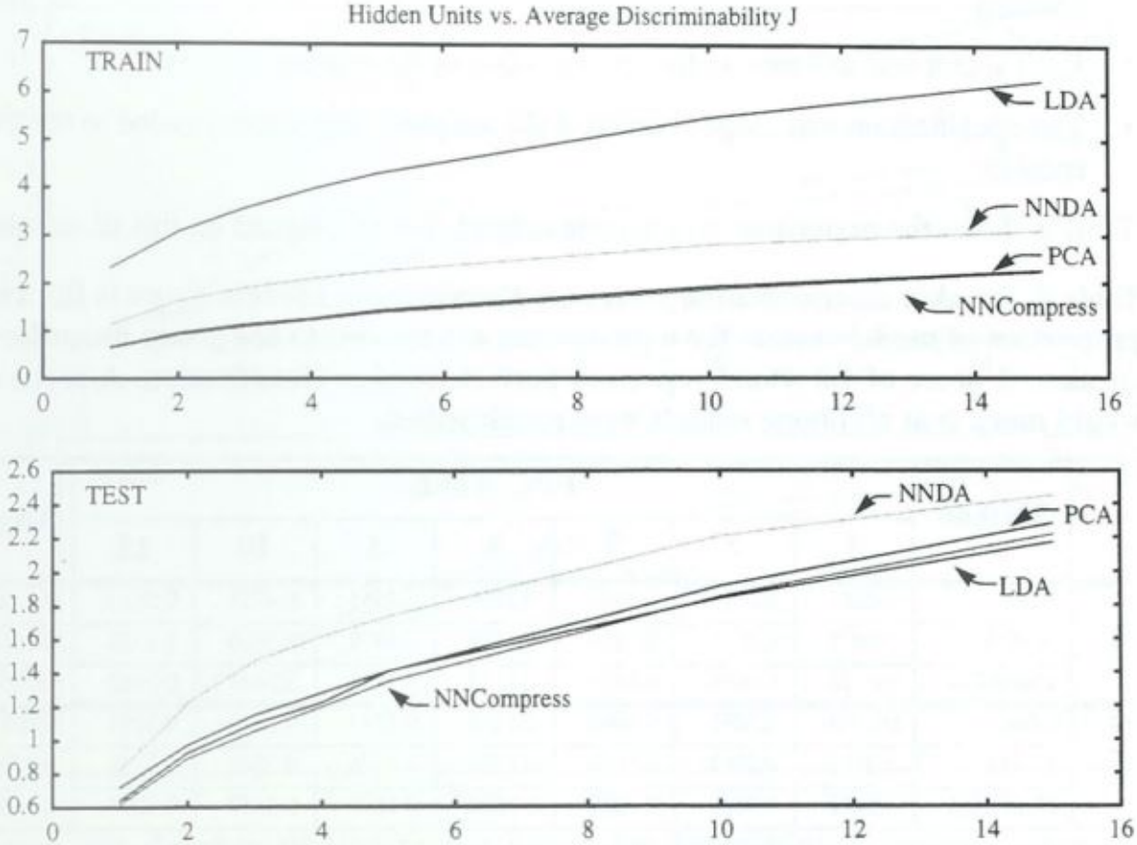


Figure 16: Discriminability of speakers on the basis of phones projected onto PPCs of various dimensions, for the four techniques, each pooled across phones. Linear discriminant analysis produce the most separated speaker codes for the training speakers, but for the test set, this advantage was lost, and the neural net discriminator was most successful.

of speakers in each model increases with the number of parameters in the PPC devoted to modelling each phone, with diminishing returns from extra hidden units. Clearly the amount of speaker information increases with the size of the PPC. Since the curve was levelling off above this point, PPCs with ten components seemed a reasonable choice to carry forward for use in building speaker codes.

Performance measured directly on a discrimination task

Since the training technique used for the NNDA model was trying to achieve good discrimination performance, rather than match the LDA model in maximising the separation of speakers in the space whose volume is approximated by the J measure, it was important, for comparison purposes, to see how well the NNDA networks and LDA performed on the discrimination task itself. To this end, a more direct measure of discriminability was calculated for each condition. For each speaker, the centroid for all the PPCs associated with a given phoneme was calculated. These centroids where used to calculate speaker classification rates, using nearest centroid classification for both the output vector of the NNDA model network and the PPCs formed in the hidden units, and for the LDA PPCs. The classification was done as follows:

- Within each phoneme, means were calculated for the set of vectors from each speaker.
- Each vector was assigned to the speaker class of the nearest mean vector.
- The classification was judged correct if the assigned class corresponded to the true speaker.

Table 8 shows the performance (0 = none correct, 1 = all correct) on this discrimination

Table 8: Speaker discrimination scores for Discriminant models. Score is the average proportion of model vectors for a phone that are nearest to the group mean for their speaker. A score of 1.0 would represent perfect speaker classification. A score of 0.0 would mean that all phone models were misclassified.

	Method	PPC Width							Means
		1	2	3	4	5	10	15	
Train	LDA	0.0291	0.0780	0.1355	0.1963	0.2581	0.4753	0.5933	0.2522
	NNDA	0.0437	0.0546	0.0886	0.1188	0.1478	0.2666	0.3323	0.1504
	HidNNDA	0.0226	0.0491	0.0841	0.1153	0.1435	0.2444	0.2862	0.1350
Test	LDA	0.0383	0.0697	0.0960	0.1219	0.1547	0.2539	0.3273	0.1517
	NNDA	0.0713	0.0891	0.1250	0.1568	0.1876	0.2881	0.3359	0.1791
	HidNNDA	0.0441	0.0839	0.1227	0.1520	0.1803	0.2682	0.2994	0.1644

task for LDA PPCs (LDA), the outputs of the NNDA network (NNDA) and the PPCs formed in the hidden units of that network (HidNNDA). The complete tables of results for the discrimination tasks are given in Appendix D. as Tables D-5, D-6 and D-7.

As one would expect, the more units that were used, the better the discrimination. Reflecting the results with the J measure, LDA based PPCs serve to discriminate the training set somewhat better than those based NNDA ($t_{12} = 1.34$, P(LDA<HidNNDA) < 0.102), but this advantage is not robust. For test speakers, the hidden representations formed by the neural net discriminator are more distinct than the LDA representations[17] although this difference is not significant (t_{12}=0.24 p(means equal)<0.814).

Although, again, the difference is not significant, the discrimination score is higher for the output units of the NNDA than for the hidden units, suggesting that the previously stated reservations about the use of the J score in this case were not entirely unfounded - the internal representations formed are somewhat more distinct with respect to a non-linear classifier than the linear classifiability measure J would lead one to expect. The measure seems to slightly underestimate the quality of PPCs formed in non-linear models.

3.8.4. Conclusions

Since the vowel and consonant models have similar power to distinguish speakers, the choice of the most frequent phonemes regardless of class, as a basis for building the models was confirmed as a reasonable one.

17. In fact, the NNDA does slightly better for test speakers than for training speakers. This is probably coincidental.

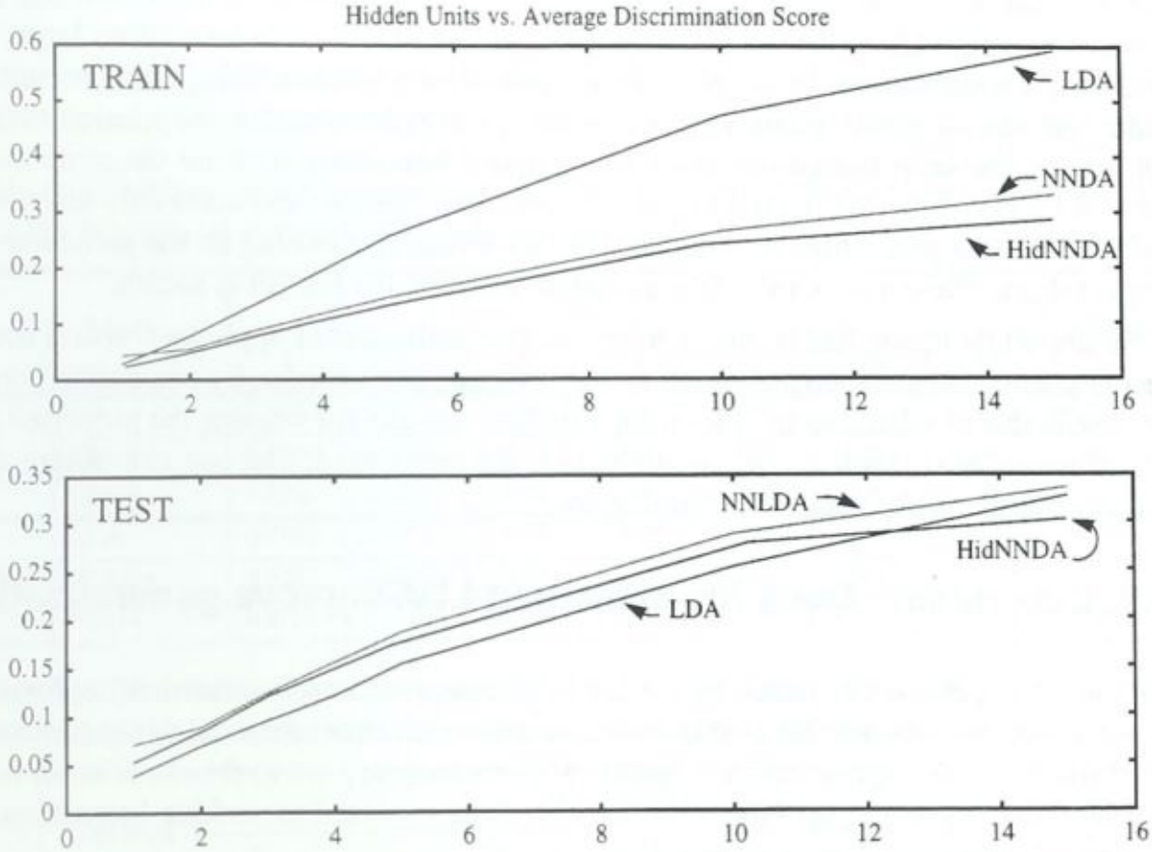


Figure 17: Speaker discrimination scores for Discriminant models. Score is the average proportion of model vectors for a phone that are nearest to the group mean for their speaker. Again, in this case, the training set advantage for LDA is lost on testing data, but here there is no clear advantage for the neural net during testing.

For the variational techniques, the principal components analysis and the neural network compressor performed nearly identically. This suggests that these networks, like the ones used in Cottrel's [cottrell90] early work on face compression, were simply calculating principal components of the phone example vectors, and very inefficiently at that. When the training of example compression networks was described earlier in this chapter, it was pointed out that it took a very long time, even with the very clean data used, to escape from the linear approximations of the solutions that the networks learned initially. Even if the networks could have learnt a non-linear compression from the noisy data used here, it is far from clear that the number of training epochs used was sufficient, despite the fact that this amount of training took a vast amount of computation when summed over the 210 networks trained. If bottleneck compression networks are to be a useful technique for training non-linear encodings, then it is clear that specialised methods for improving the training of such networks need to be found. One possibility, that there was not sufficient time to explore, would be to preload the network with weights derived from principal components of the training data, ensuring that the network's training is focused entirely on learning a non-linear component of the encoding.

In the discriminant case - the more usual application of neural networks - there was weak evidence that the neural nets performed somewhat better than the linear discriminant analysis on testing data. Since the LDA discriminated training speakers better, it is likely that this

result is not due to the availability of non-linear decision surfaces in the neural net, but to overfitting in the LDA. While the amount of training data was, *in toto*, rather large, there was only a small number of samples of each phone for a speaker. This limitation may have led to the sample within group variances giving poor estimates of the population statistics. Of course, the same limitations apply to the neural nets, since they use the same training data, but it seems that for the training regime used, the nets have extracted only discriminant information that generalises to new speakers, rather than specialising for the particular training speakers. These questions will be explored further in the following section.

An important lesson lies in this experiment. The usefulness of applying a neural network technique to an application is difficult to fully evaluate in a vacuum. By comparing performance with that of related techniques from statistics, one can tell whether the purported benefits of neural nets, such as their nonlinearity, are being used, and one can obtain useful information about the nature of a training set.

3.9. Experiment: Does NN training of LDA improve generalisation?

An interesting possibility raised by the last experiment is that the somewhat improved performance of NNDA over LDA on testing data derives not from the availability of nonlinear discriminant surfaces, but from the nature of the training applied to the classifier. If nonlinearities were important, the NNDA would have been expected to perform better than LDA on training data as well as on testing data.

Bridle [personal communication 95] has suggested that the difference might be due to underfitting. Although, in the limit, a linear neural network classifier, and a classifier using linear discriminant functions that have been learned using discriminant analysis, should perform identically, it is possible that, in fact, this limit is not reached. By gradually approaching a classification model approximating the training data, the neural net training may underfit the data in a way more likely to model those statistics of the training set that are appropriate for generalisation, rather than qualities of training set outliers that are not shared by the data in testing sets.

3.9.1. Experiment Part I

To test this hypothesis, entirely linear three layer neural net classifiers (NNLDA) were trained, and their performance at producing speaker-discriminating PPCs compared with that of both of the previously discussed LDA and the non-linear NNDA classifiers.

The networks were similar to the NNDA nets used above, having 253 linear output units, corresponding to speakers, 320 linear inputs[18] for the fixed length phone acoustic vectors, and one hidden layer with a number of linear units corresponding to the desired PPC width. Networks for each phone and each PPC width were trained for 1 000 epochs each with a learning rate of 0.0001, momentum of 0.9 and weight decay of 0.00001.

18. Five frames of sixty-four FFT filterbank coefficients each.

Results

Discrimination performance, both as estimated by the J measure, and as measured using nearest centroid classification, is given in Tables 9 and 10, respectively. Again, in terms of

Table 9: Discriminability measure (J) for PPCs for three discriminant methods. The NNLDA is a neural network with only linear units.

	Method	PPC Width							Means
		1	2	3	4	5	10	15	
Train	LDA	2.3270	3.0750	3.5700	3.9689	4.3052	5.4825	6.2636	4.1417
	NNLDA	0.7084	1.5231	1.9435	2.2141	2.4505	3.2561	3.7732	2.2670
	NNDA	1.1295	1.6258	1.8922	2.0941	2.2495	2.8063	3.1099	2.1296
Test	LDA	0.7190	0.9711	1.1491	1.2800	1.4095	1.8395	2.1721	1.3629
	NNLDA	0.5307	0.9656	1.2483	1.4253	1.5698	2.0647	2.4172	1.4602
	NNDA	0.8179	1.2218	1.4707	1.6145	1.7368	2.2069	2.4499	1.6455

Table 10: Discriminant performance of PPCs for three discriminant methods. Figures are correct classification of vectors using a nearest centroid method.

	Method	PPC Width							Means
		1	2	3	4	5	10	15	
Train	LDA	0.0291	0.0780	0.1355	0.1963	0.2581	0.4753	0.5933	0.2522
	NNLDA	0.0155	0.0338	0.0606	0.0964	0.1340	0.3008	0.4098	0.1501
	NNDA	0.0226	0.0491	0.0841	0.1153	0.1435	0.2444	0.2862	0.1350
Test	LDA	0.0383	0.0697	0.0960	0.1219	0.1547	0.2539	0.3273	0.1517
	NNLDA	0.0342	0.0512	0.0826	0.1116	0.1483	0.2758	0.3442	0.1497
	NNDA	0.0441	0.0839	0.1227	0.1520	0.1803	0.2682	0.2994	0.1644

the J measure, the neural network seems to have been trading off performance on the training set for performance on the test set. LDA generated the projection that best maximised the J measure on the training set, followed by NNLDA, the linear network, and NNDA, the potentially non-linear one.[19] However, this ordering was not consistent across dimensions; at low PPC widths (1,2) NNDA outperformed NNLDA on training data. For testing data, the relative performance of the methods on the J measure was reversed. The NNDA network did best, followed by the NNLDA net and the LDA projection. Again, the ranking was not consistent, with LDA outperforming NNLDA for the lowest two dimensions.

Although weak, these results seemed to lend support the idea that one can get better testing set discriminant performance, even from a linear discriminator, by using neural net training rather than the direct calculation of eigenvectors of the ratio matrix. However, when the performance of the projections on a classification task is examined, the results are even less clear. In this case, results for which are given in Table 10, the ordering of performance on

19. It should be noted at this point that there are more differences between NNLDA and NNDA than just whether linear units are used. The NNDA network had five hidden layers with shortcut connections outside the bottleneck; the NNLDA net had three layers.

the three methods on training data followed that predicted by the J measure, except that the NNLDA does not begin to outperform the NNDA until a PPC dimension of 10 is reached. For testing, the NNDA outperformed the other methods, and the LDA and NNLDA techniques have similar performance. The NNDA's superior performance was most evident at lower dimensions; for ten- and fifteen-dimensional PPCs, the NNLDA net worked best, outperforming the NNDA and LDA classifiers, respectively.

Discussion

The hypothesis was that the neural network worked best because it failed to learn spurious features of the training set that would interfere with testing set performance. While this hypothesis would seem to have been supported for the J measure, when it came to classification the story was unclear. Although, as was noted before, the NNDA network generalised better than LDA, though not at the highest PPC dimension, the generalisation performance of the NNLDA was the same as or slightly below that of LDA, except at high dimensions. While the results on the J measure do suggest that training classifiers with neural network methods, possibly resulting in underfitting, has a good effect on generalisation, such an effect remains to be clearly demonstrated. The improved performance of the NNDA over NNLDA on classification tests in lower dimensions remains to be explained. More than just the mere fact of neural net training is needed to account for this difference.

A possible mechanism for improved generalisation performance of a non-linear discriminator is suggested by Ayer [ayer93]. In this paper, the authors pointed out that the nonlinearities inherent in neural network outputs limit the effect of large recognition score differences, and encourage the networks to concentrate on cases near the borders of classification regions. This argument is made in the context of networks with logistic outputs, and cannot be directly applied to the networks used here, where linear output units were used. However, [ayer93] is concerned with producing a similar concentration on borderline cases for HMM training, and, in that context, derives an error measure similar to CFM. While one would expect an improvement in generalisation from the use of CFM to apply in equal measure to the NNLDA and the NNDA network, there are remain hidden unit nonlinearities in the NNDA that could further limit the effect of outliers.

Another possibility is simply that the NNLDA net reaches its maximum generalisation performance earlier on than NNDA does, and that the "advantage" of NNDA on testing data is due to its slower learning, relative to the fixed 1 000 epoch training interval. The following experiment examines that possibility.

3.9.2. Part II: Time course of training.

To see whether this explanation, that the NNDA's performance advantage was due to relatively greater underfitting due to slower training of the NNDA, was plausible, networks with identical structure to the NNDA and NNLDA networks used in the first part of this experiment were retrained from scratch. This time, however, classification performance was tested on every twentieth epoch of a 1 520 epoch training interval, in the case of NNDA, and a 1020 epoch training interval, in the case of NNLDA[20]. Because running these tests was rather time consuming, only networks with five hidden units were trained.

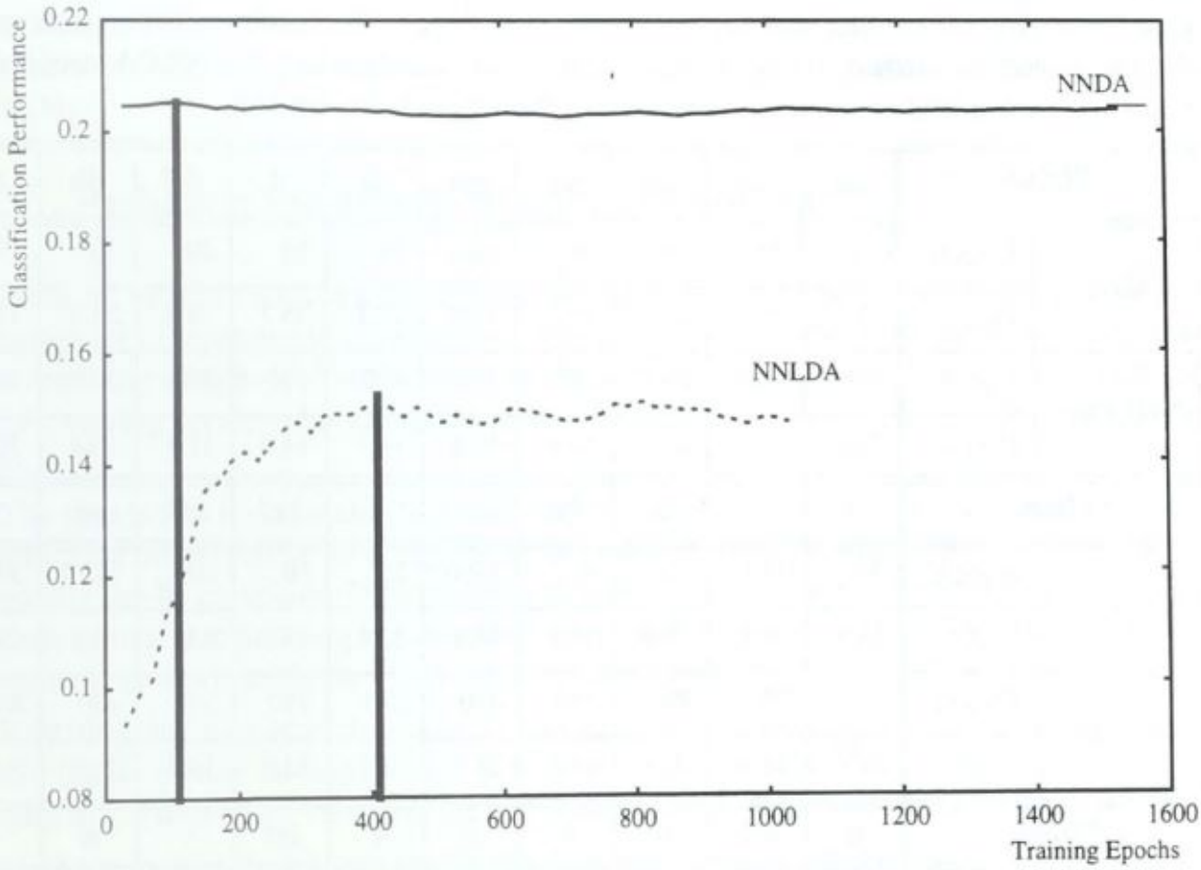


Figure 18: Evolution of classification performance, averaged over phones, through time for 5-hidden-unit NNDA and NNLDA networks. Maximum performance is reached after one hundred and ten, and four hundred training epochs respectively.[a]

a. The slight difference in performance compared with table 10 were probably due to different random initial weights.

Results

Figure 18 graphs the mean classification accuracy, averaged over phones, on test data for 5-hidden-unit NNDA and NNLDA networks measured on every twentieth training epoch. Unaveraged data appear in Table 11. Maximum performance was reached after 110(±10) and 410(±10) epochs respectively, with the NNDA network achieving a maximum classification performance of 20.5% and the NNLDA network reaching 15.2%. Since the NNDA classifier reached its maximum generalisation performance early, well before the NNLDA, it is clear that the generalisation advantage is not caused by underfitting due to undertraining. It is important to note that the difference between the methods on training data is somewhat exaggerated by the graph, since the ordinate does not start at zero.

20. Using the NN simulator written for this thesis, this could produce slightly different results than for the previous training run. Stopping the training to measure performance on the test set resulted in accumulated momentum values being lost.

Table 11: Epoch of maximum test set performance, and classification performance at that epoch, for all phones and the linear and non linear NN classifiers. There is a great deal of variation within network types and across phones, but the NNDA recogniser tends to reach a higher performance, and earlier. Epoch #'s are ±10.

Phone		aa	ae	ah	ax	axr	b	d	dcl	dh	eh
NNDA	Epoch	10	10	190	50	10	10	50	10	10	70
	Perf	23.1	28.2	26.1	15.2	14.0	22.7	35.7	18.2	23.3	15.9
NNLDA	Epoch	470	390	590	790	330	490	130	370	30	390
	Perf	23.7	32.8	23.3	16.4	12.6	14.7	14.0	11.7	13.4	20.4
Phone		**ey**	**f**	**ih**	**ix**	**iy**	**k**	**kcl**	**l**	**m**	**n**
NNDA	Epoch	30	110	10	90	1490	50	50	50	1270	330
	Perf	27.1	29.8	17.0	13.1	18.8	22.4	15.2	9.9	24.4	21.2
NNLDA	Epoch	250	790	390	290	410	270	750	310	430	810
	Perf	27.8	16.6	19.9	15.8	21.3	13.1	12.0	14.4	21.5	24.1
Phone		**p**	**pcl**	**q**	**r**	**s**	**t**	**tcl**	**v**	**w**	**z**
NNDA	Epoch	1010	410	230	930	1470	10	1210	510	10	130
	Perf	26.9	15.7	20.0	11.0	27.4	24.4	8.9	24.2	18.9	28.6
NNLDA	Epoch	370	910	130	130	170	10	810	210	410	170
	Perf	17.8	12.5	12.0	12.4	19.3	12.1	6.8	15.6	17.4	17.3

3.10. Discussion

Nonlinear classifiers like NNDA can, in principle, develop more compact encodings of information than linear methods, or permit the encoding of more information in a given size of representation. Regrettably, from the point of view of a proponent of neural networks, this promise was not realised for the compression of information about inter-speaker differences in phone pronunciation. For training set data, it was generally possible to do better speaker discrimination with linear classifiers than with the multilayer neural network.

Although the better performance, as assessed by the J measure, of the NNDA on testing data held out the hope that its learning was at least more robust, this advantage was scarcely retained when classification accuracy was measured directly.

Since the advantage of NNDA networks over their linear equivalent, in a case where an advantage was present, appeared early on in training, it cannot be that the NNLDA and other linear methods are suffering an disadvantage due to overfitting the data.

It seems likely that there are no substantial modelling advantages to any of the networks for this application, and that the differences in performance on training data that were apparent between them might well have been due, for example, the initial choice of network weights.

3.11. Reducing the effects of phonetic context

Although, as was mentioned during the discussion of the choice of phones (§3.2), it is impractical to separate out the effects of phonetic context by the ideal method of modelling the resulting allophones[21] separately, in essence holding the context and phone constant while varying speaker characteristics, it may be possible to achieve partial control for context.

The idea is that if one is able to model the effect of context on the acoustic realisation of a phone, that model can be used to generate the reference against which phones from different speakers can be compared. What remains should show the effects of speaker variation more clearly than if one had simply subtracted the overall phone mean from the speaker specific instances, as was done in the experiments reported above.

Restating this idea more formally: if it is a reasonable approximation to assume that the fixed length phone instance vector $\mathbf{p}$ is generated additively[22] from an allophone mean model $\mathbf{a}_{l,c,r}$ (where l,c,r represent the left context phone, the phone itself, and the right context phone, respectively), and the speaker's effect, $\mathbf{s}$, on the phone *i.e.* $\mathbf{p} \sim \mathbf{a}_{l,c,r} + \mathbf{s}$, then the system's ability to estimate the variation in $\mathbf{s}$ can be improved if an estimate of $\mathbf{a}_{l,c,r}$ can be obtained.

One way to obtain such an estimate would be simply to calculate the allophone means from the database, and, indeed, for frequently occurring allophones, this might be the best solution. Unfortunately, in the TIMIT training database, many of the possible allophones occur only once or not at all. If the allophone mean is "estimated" from a sole exemplar, there will be nothing left from which to estimate speaker variation in the phone c, which is, after all, the point of the exercise. In cases where the training database contains no examples at all of an allophone used in testing, there is no data at all from which to estimate the allophone mean directly.

Instead of trying to model each allophone separately, in this way, one can suppose that the context dependencies are regular, just as the speaker modelling work supposes that speaker dependencies are regular. Unlike speaker dependencies, however, context dependencies are transparent - the dimensions of variation are known: they are the set of values the left and right phoneme can have. It is therefore possible to make an estimate of $\mathbf{a}_{l,c,r}$ as a function f of the phone labels, i.e. $\hat{\mathbf{a}}_{l,c,r} = f(l, c, r)$. In practice, the phone labels are represented as *one-from-n* binary vectors, concatenated to form a label vector $\mathbf{l}$ specifying the allophone:

21. The term "allophone" will be used rather loosely in this section. Concerned readers may wish to read it as the more precise "phone in context" or the commonly used "context dependent phone".

22. Noting as we do so that we don't believe this assumption for a moment, since it is certain that there is a strong interaction between speaker and phonetic context in determining allophone means. We hope however that this simplified model helps.

$\hat{\mathbf{a}}_{l,c,r} = f(\mathbf{l})$. In the following experiment, such estimates of allophone acoustics are generated and compared.

3.12. Experiment

In this experiment, a number of ways of generating the allophone estimates $\hat{\mathbf{a}}_{l,c,r}$ were evaluated by comparing their ability to predict actual values of $\mathbf{a}_{l,c,r}$ measured from the database. This comparison was performed only for allophones that occurred twice or more in the testing set.

As a pessimal baseline estimate against which others could be compared, a constant, per phone estimate $\hat{\mathbf{a}}_{l,c,r} = \bar{\mathbf{a}}_{l,c,r}$ was generated. This estimate was the overall mean value of the means of the allophones means for those allophones of the phone c that occurred twice or more in the *training* set. There were also four allophone specific estimates generated by four forms for the function f mentioned in the previous paragraph. The first of these was a linear transformation from phone labels onto phone acoustics, implemented as a neural net with no hidden units and linear outputs, and the remaining three estimation functions were implemented as three layer neural nets, attempting the same transformation, and having five, ten, and fifty sigmoidal hidden units respectively.

3.12.1. Procedure

For each centre phone c in the set of thirty frequently occurring phones, mean vectors were calculated for each phonetic context for which there was more than one instance. A corresponding 152 component vector was also generated to specify the identities of the left (l), centre (c) and right (r) phones. The centre phone was specified using thirty components of which twenty-nine were set to zero, and the remaining one, corresponding to a phone index, was set to one. The two context phones were specified similarly, but in this case the vectors used had sixty-one elements each, to allow for the full set of sixty-one possible phones, frequent or not, to be used as context.[23] 5 513 training patterns were generated from the TIMIT "train" data in this way, along with 2 417 testing patterns from the "test" data.

The four allophone estimation models described in the previous paragraph were trained to use the binary phone-in-context specification as input, and produce a prediction for the average acoustic representation of that allophone as output. Neural networks with "bypass" direct connections from the input to output as well as the usual connections to units in the hidden layer, were each trained for 1 000 epochs (5 513 000 pattern presentations). The learning rate was 0.0001, momentum 0.9 and decay 0.00001.

Performance for each model was measured by running it in feed-forward mode with the binary patterns from the testing set as input, and measuring mean Euclidean distance

23. The complete list of these phones is given in §A.1 on page 181 of Appendix A.

between the output allophone acoustic representation $\hat{\mathbf{a}}_{l,c,r}$, and the corresponding target $\bar{\mathbf{a}}_{l,c,r}$, measured over the instances of that allophone found in the testing set.

Table 12: Comparison of methods for estimating the effect of allophone variation. Four methods of estimating the acoustics of a context allophone were compared with an estimate based on the mean phone acoustics. All four of the estimates were closer to the actual allophone means than the phone mean was, and the nonlinear estimates were closer than the linear one. The confidence tests are for the hypothesis that the estimates better approximate allophone acoustics than the overall phone mean does.

$\hat{a}$	hidden layer	mean distance[a] from $\hat{a}$		Test Set Statistics	
		Train	Test	t_{2416}	p-value[b]
Overall phone mean		5.81	6.35		
Linear estimate	0 units	5.63	6.30	1.42	<0.08
Nonlinear estimate 1	5 units	5.02	5.84	11.79	0.0
Nonlinear estimate 2	10 units	4.74	5.80	11.02	0.0
Nonlinear estimate 3	50 units	4.01	5.98	6.26	0.0

a. Euclidean distance between vectors

b. probability that true mean of difference between estimate and that from overall phone mean is not greater than zero, using a paired t-test over all test phone means

3.12.2. Results

Table 12 gives the average Euclidean distance between the estimates generated as specified in the preceding paragraph and the corresponding allophone means calculated from the actual data. This comparison is made between the estimated allophone acoustics and both the set of means estimated from the training data, and the set estimated from the testing data.

One tailed t-tests were performed to test the hypothesis that the estimates matched the test set allophone statistics more closely than the overall phone mean did. Even the very simple linear estimate was slightly better than the overall phone mean estimate (the probability that it was not was less than 10%). The non linear estimates were all substantially better fits to the data than the phone mean. Of these estimates, there was some evidence of overfitting by the network with 50 hidden units which had better training set performance but worse testing set performance than the 5 and 10 unit networks.

Although the t-tests that demonstrate this have not been included in the table, the non-linear estimates of the effects of phone context were also all significantly better than the linear one, with slightly greater confidence than for the tests shown.

3.12.3. Discussion

Nonlinear estimates of phonetic context effects on phone acoustics were able to explain a significant source of non-speaker-related variability in phone acoustics. By using these estimates to reduce the context effects on phone acoustics before attempting to build models of speaker variability, it should be possible to reduce the noise in these models and to improve their quality. Since these experiments were done rather late in the course of this work, they were not applied to any of the models reported. Their application should be pursued in future work. Using neural networks to estimate the effects of context on phone acoustics also has potential application in speech recognition, where gathering sufficient data to estimate distributions over context dependent phones is also problematic.

3.13. What does the J measure mean?

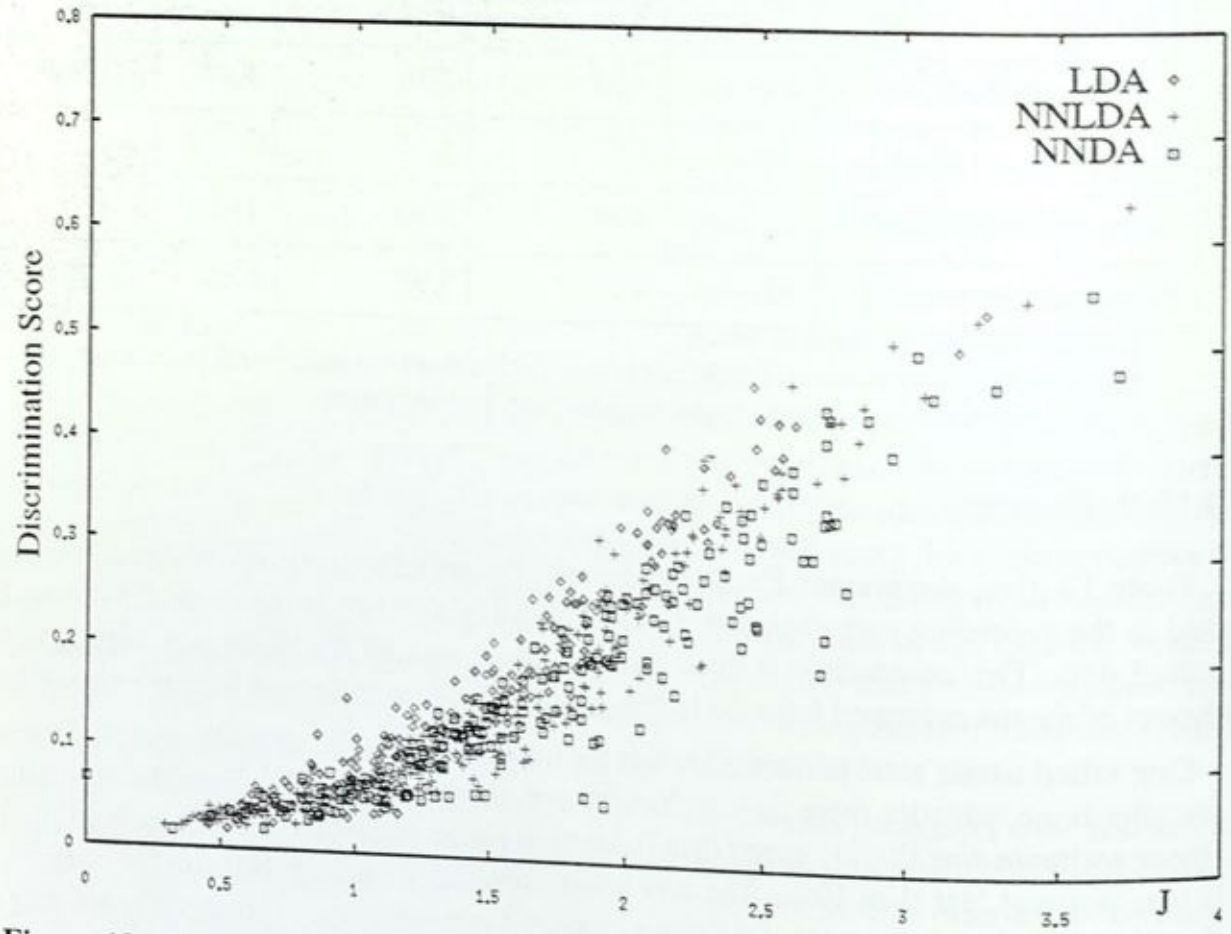


Figure 19: The relationship between the J measure and discrimination performance on the testing set for three phone models. Each point represents a particular combination of phoneme and PPC dimension. Values of J are on the ordinate and discriminant performance, measured as proportion of correct classifications, is given on the abscissa.

When it was introduced, it was pointed out that the purpose of the J measure was to give a readily calculated estimate of the discriminability of speakers in the space in which a set of PPCs or SVCs lie. Since, in the course of doing experiments, nearest centroid classification

scores were gathered along with the J measure for the PPC candidates in this chapter, it was worth comparing the two measures.

Figure 19 shows the discriminant performance plotted against the J measure for PPCs produced by the three discriminant phone models applied to data in the test set. Each point marked on the graph represents a particular combination of phoneme identity and PPC dimension.

After inspecting the shape of the scatter plots, and making the reasonable assumption that when the value of J is zero, the discrimination performance should be zero, an attempt was made to fit a model relating the discrimination performance to the square of the J measure as follows:

$$disc = ax^2 + \varepsilon$$

Table 20 shows that when such a model is fit to the data using linear least squares, an

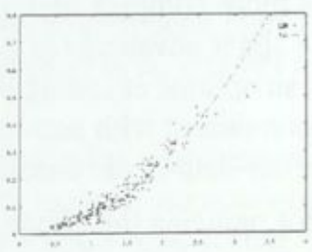
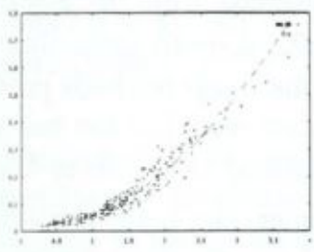
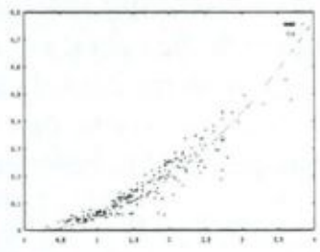
	LDA	NNLDA	NNDA
			
Equation	$disc = 0.066x^2 + \varepsilon$	$disc = 0.055x^2 + \varepsilon$	$disc = 0.048x^2 + \varepsilon$
s.d	0.0008	0.0006	0.0007
% Fit	97.1	97.2	95.6

Figure 20: The relationship between J and discrimination performance is well accounted for by modelling discrimination as the square of J for each sort of PPC. Such a model accounts for more than 95% of the variance in discrimination performance in each case. The s.d. values are the standard error for fitted coefficients.

extremely good fit is obtained. The variation in J^2 accounts for over 95% of the variation in discriminant performance,[24] and the probability of obtaining a fit this good by chance is, in each case, approximately zero.

There is clearly a difference in the coefficient relating J and discriminant performance for the PPCs generated with the three different methods: LDA, NNLDA and NNDA. While the source of this difference is not clear, it does indicate that caution is warranted in drawing conclusions about the relative merits of phone models generated by the different methods. Both measures, J and nearest centroid classification scores, measure the ability of a model to discriminate between speakers, and although they are closely related within a model, this relationship differs between models. Since it is difficult to say unequivocally that either the J

24. Other models, including disc = exp(J) and quadratics with more parameters were tried, but the simple square model fit best with fewest coefficients.

measure or the discrimination score is a better way of measuring the quality of a model, the choice of which one to use when choosing between models may be regarded as largely a matter of taste.

3.14. General observations and review

In this chapter, the first steps in modelling speaker differences were taken by investigating methods for capturing the speaker based variation in the segments that make up the speech stream. The chapter began with a comparison of the basic encoding of the speech signal in a spectral representation, and then moved on to discuss ways to deal with variability in the time course of segment production.

The bulk of the chapter was taken up with a discussion of the potential benefits of neural network encoders, and experiments to determine whether these benefits were realised in the course of forming phone pronunciation codes. In general, they were not. The use of linear methods to produce lower dimensional encodings of the speaker dependent phone variants was statistically indistinguishable from the use of the more complex neural net models, although the neural net models showed some signs of a slight advantage in generalisation. Since it seems unlikely that the linear methods produce an optimal encoding of speaker differences, it may be that the data were just too badly contaminated with non-speaker-related variation for hill climbing learning to be able to find a better-than-linear encoding solution.

In the final part of the chapter, one possible method for reducing the effects of phonemic context — a major source of non-speaker related variation — was investigated. Neural networks were successfully trained to predict these effects, giving hope that in future versions of a speaker modelling system, the quality of PPCs can be improved.

For the present, the PPCs based on NNDA will be carried forward for use in forming the overall speaker model, since they exhibited, if only equivocally, the best test generalisation performance.

Chapter 4. Overall Speaker Models

The work described in the previous chapter furnished models that captured at least some of the useful variation in individual phones. The ultimate goal, however, is to build models that can support a human-like ability to rapidly adapt to voice differences. Inferences must be made about a speaker's pronunciation of unheard phones on the basis of the phones that have already been heard, either for the purpose of better recognising them, or in order that phones sounding them might be synthesised. If, for example, the phones /iy/, /ay/ and /ch/ from a speaker had so far been heard, it would be desirable to be able to predict the sound of the phone /b/ from the same speaker.

The ideal solution would be to have available a model that yields predictions about /b/ in terms of just that subset of phones -- /iy/, /ay/ and /ch/ -- that have already been heard. This is, alas, a vain desire; since there are 61 phones used in the TIMIT data base, $2^{61} \times 61$ such models[1] would be needed - too many to store, let alone train. Of course, in suitable tasks, one can attempt to gain a benefit from a smaller set of correlations, as in Cox's [cox93] sensible work with interphone regression models of variability. In some respects, the current work can be seen as an attempt to generalise and extend the class of regression models applied.

As explained in the introduction, the motivation for this work was the hypothesis that the human ability to make use of arbitrary small sets of previously heard phones in adapting to a new speaker's voice is most simply explained by the notion that people learn a continuum, or space, in which speakers lie. Under this model, phones that have been heard at a certain point in time are used to identify the position of the speaker in this speaker space, and this position is then used to make predictions about voice quality. The speaker space is a compact model of the underlying variables that explain the variation between speakers. To give this predictive ability to computers, then, such an underlying representation (an SVC, or Speaker Voice Code) must be learned from the consistencies in the relations between the qualities of the speech tokens in a speech stream heard from a single speaker. For the current purposes, of course, these speech tokens will be represented by the PPCs developed in the previous chapter.

4.1. Design Goals

In producing this speaker space from phone pronunciation codes, a number of design goals have been pursued. While any realisable model will fail to meet these goals in some respects, the final speaker voice code should exhibit:

- *Consistency within a speaker:* a single speaker should be placed at a single position in the space.
- *Separation between distinct speakers:* different speakers should be represented at distinct positions in the space, if their voices are distinguishable.
- *Perceptual relevance:* Speakers who are nearby, or who are widely separated in the

1. Which evaluates to about 1.4×10^{20}.

speaker space, should have voices that sound similar or different, respectively, when judged by human listeners.

- *Compactness:* To permit the model, and applications that use it, to be used to generalise to new speakers, the space needs to be reasonably densely covered by training speakers. This can only be achieved if the model has low dimension.
- *Text independence:* Human listeners do not need to have the voices they listen to utter a fixed enrollment phrase, and neither should computers. The model should reach the same point in the speaker space, for the same speaker, irrespective of what the speaker has said.
- *Rapidity of formation:* Human beings show significant adaptation after a few syllables have been uttered. Similarly, the SVC produced by the model should approach the final speaker position in speaker space as rapidly as possible, using information from additional speech, as it becomes available, to refine the position.
- *Robustness in the face of noise:* If the speaker says some tokens oddly, or some tokens are obscured by noise, they shouldn't prevent the speaker model from reaching the correct position. Given enough additional speech, the model should recover from such noise in its input.
- *Thoroughness:* The model should retain enough of the available information about speaker variability to make the voice codes produced with it useful in applications.

The obvious first step to take in building such a model is to simply concatenate the phone models produced by one of the methods outlined in the previous chapter, filling in the models for unheard phones with some estimate of their value. Then, as with the PPCs, a neural net, or a linear method, can be used to reduce the dimension of this concatenated vector, yielding a vector giving the position of the speaker in the speaker space.

To assemble the PPCs into Speaker Voice Codes (SVCs), essentially the same techniques used in the last chapter to build phone models are used. PPCs for the phones in heard speech, and estimates of PPCs for unheard phones are concatenated together into a single vector, whose dimensionality is reduced using either a linear projection or a neural net. This process is outlined in Figure 21.

Within this framework, some of the goals for speaker modelling are easier to satisfy than others. Text independence is easily maintained so long as the training data contains a sufficient variety of strings to prevent the modelling of text characteristics in the training set — since the majority of the speech for each speaker in the TIMIT database is for a set of sentences unique to that speaker, the database chosen satisfies this constraint.

Similarly, a compact representation will be formed by any of the techniques if a low dimension is chosen on which to project the concatenated vectors. The only task is to ensure that enough of the speaker information is retained. Compactness and thoroughness are somewhat incompatible, a difficulty that can only be reduced by finding the most efficient possible encoding. It is this goal that drives the attempt to apply neural networks to the modelling task. As discussed in the previous chapter, the nonlinear functions that neural nets can compute ought to be able to provide more compact encodings for the same amount of data than

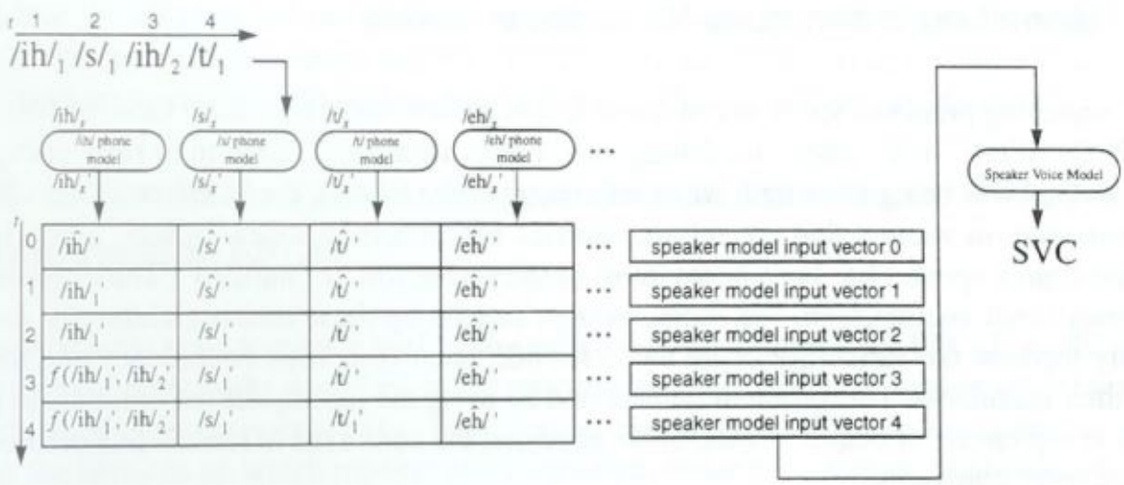


Figure 21: The general speaker modelling scheme. Outputs p_x' of the phone model for phone p on example x are concatenated to form speaker model inputs. PPCs for unheard phones are estimated as $\hat{p}$'. Successive examples of a single phone are combined by some function $f()$, which may combine both PPCs based on actual observations, and PPCs estimated from the SVC.

those found by linear statistical methods — although this was not clearly demonstrated for the phone models.

Other model qualities compete with each other, and it is necessary to choose which to favour. Consistency within a speaker competes with text independence, since no matter how consistent individual PPC phone codes within a speaker may be, choosing a new set of them, as a result of using different text, is bound to produce a somewhat different speaker code. In this case, text independence will be favoured, since this strikes the author as an indispensable part of human speaker modelling performance.

Although the aim is to produce a general model of speaker variation that, like the model, or models, human beings are hypothesised to use, can be applied successfully to a wide variety of applications, there are also trade-offs between design goals, driven, to some extent, by the applications to which the model is to be applied. For this reason, simply comparing model performance on some set of speaker discriminability measures may not be enough. In some cases, one can imagine wanting to use variational methods, even if discriminant techniques make models that better distinguish speakers. One might, for example, have a use for the estimates of PPCs that can be generated from SVCs produced by variational techniques as inputs to a phoneme based speaker adaptation technique. In this case, using those techniques would be justified, even if, by doing so, the consistency of the SVC was reduced.[2] Moreover, as pointed out in the previous chapter, forming the model in the process of doing speaker discrimination, as the discriminant models do, may cause the loss of some perceptually relevant variation that humans see as characteristic of speakers, if this information doesn't serve as one of the main features distinguishing the speakers in the training set.

2. This is likely, since a speaker discriminant constraint is intended to make the model head for a single known target for a given speaker.

4.2. Modelling techniques for speaker codes

The modelling problems faced in constructing SVCs differ somewhat from those of forming the PPCs. While, in the phone modelling case, it was always the variation in fully occupied vectors that was being modelled, when building speaker models, the intention is to capture the variation in vectors that are missing entries for unheard phones. In fact, when little speech from a speaker has been heard, most of the vector will be "unfilled". The differences between input vectors from the same speaker, caused by these missing elements, could greatly increase the variability in the SVCs formed, decreasing their consistency and therefore their usefulness. This problem is addressed by using the incomplete set of inputs to predict a complete set of output targets, in the hope that the code used to furnish this prediction will be more consistent.

When the neural network using a bottleneck (NNCompress) to do non-linear compression is used, on the other hand, the flexibility exists to train the system to do completion within the existing structure. It is possible, while training, to use a target vector that differs from the input vector. While the input vectors contain the phone codes that have been seen so far, the target vectors can contain the PPC, for each phone, that was seen most recently, or, in the case of phones that have yet to be "heard", the PPC that will be seen next. Alternatively, the target can contain the PPC that is heard least distantly in time, either in the future, or the past, or it can contain the mean value of all PPCs, representing the phone in question, that will be uttered by the speaker. Yet another possibility is to use the same vectors for input and "target", as in PCA, but to set up the training procedure so that it does not back-propagate any error at all from unoccupied target codes. This last method leaves the network free to make whichever estimates it likes for missing entries, so long as it efficiently represents the information it does have in the SVC. In all these cases, of course, what the network can learn to do is no longer directly analogous to PCA, since it is being used as a general function approximator.

There is another modelling improvement available when neural net compressors are used to do pattern completion: If the output prediction for missing inputs is a better estimate than the mean, it should be possible to improve both the consistency of the models, and the rapidity with which the models reach a consistent position, by repeatedly copying — or recirculating — the output unit predictions for missing inputs back to the input, and re-running the network to find an new set of output predictions along with the new SVC. Preliminary modelling experiments with French digits, which will be described briefly in chapter 5, supported this approach [witbrock92], so it is instructive to explore whether the technique is useful for larger modelling problems.

If connectionist techniques are able to outperform the linear models, one would expect them to do so most substantially when used in building variational speaker models.

A number of experiments are described in the remainder of this chapter, starting with two experiments in which SVCs were built in the same way as PPCs, in one case using NNCompress PPCs as input, and in the other using the better performing NNDA PPCs. Following these are descriptions of experiments with models trained, as outlined above, to do completion of their partial input, both with and without recirculation.

Since the aim here is to contrast the models built using this variety of techniques, these could be regarded as a single experiment. Their division into groups is more intended to break them up for easier digestion, than as a claim of some fundamental division.

4.3. Experiment: Speaker Models derived from Neural Network Compression PPCs.

The first large scale speaker model built was based on ten-dimensional PPCs from the neural compression networks described in the previous chapter. While this phone model actually performed the most poorly out of those tested, according to the evaluation criteria used, it was the first one on which training was completed, making it a natural candidate for use in building an initial speaker model. More importantly, comparison of this model with the one, described in the following experiment, formed from the PPCs output by discriminant nets will serve to give a sense of how PPC quality influences SVC quality.

4.3.1. Method

Before being assembled into speaker model input vectors, as shown in Figure 21, the PPCs were normalised by subtracting the global mean of the PPCs for each phone from their respective examples. This mean was computed over all training speakers. PPCs were not normalised to uniform variance in this experiment. PPCs were presented to the speaker modelling system in the order the corresponding phones appeared in the speech contained in the database. These PPCs were inserted, one by one, into the three-hundred element[3] input vector. The vector was reset to zero for each new speaker, equivalent to estimating unheard phone PPCs by their mean[4]. The function used to combine successive PPCs for the same phone within a speaker was replacement, i.e. $f(p_{x-1}', p_x') = p_x'$. Since this resulted in a total of 43 354 training and 14 275 testing patterns, only every fifth pattern generated was used for training or testing[5].

A single speaker model was trained for each of the four modelling techniques, Linear Discriminant Analysis, Principal Components Analysis, Neural Net Discriminant training, and Neural Net Compression, described in the previous chapter, using parameters listed in Appendix B.

4.3.2. Results

To estimate the dimensionality of the speaker information, sorted eigenvalues for the input vector covariance matrix were examined. These values are plotted in Figure 22. Most of the variation seemed to be contained in the first three dimensions. Although there was no sudden fall off in the eigenvalues, which would indicate a hard limit on the dimension of the data, they fell off slowly and smoothly beyond around the tenth value. There was, therefore, no clear reason to pick a particular value above ten for a model dimension. The presence of

3. Thirty phones by ten values per PPC.
4. Normalisation having ensured that all the phone PPC means were zero
5. The final training set of 8 670 patterns still occupied 17.5Mb, explaining, at least in historical context, why keeping all the patterns was impractical.

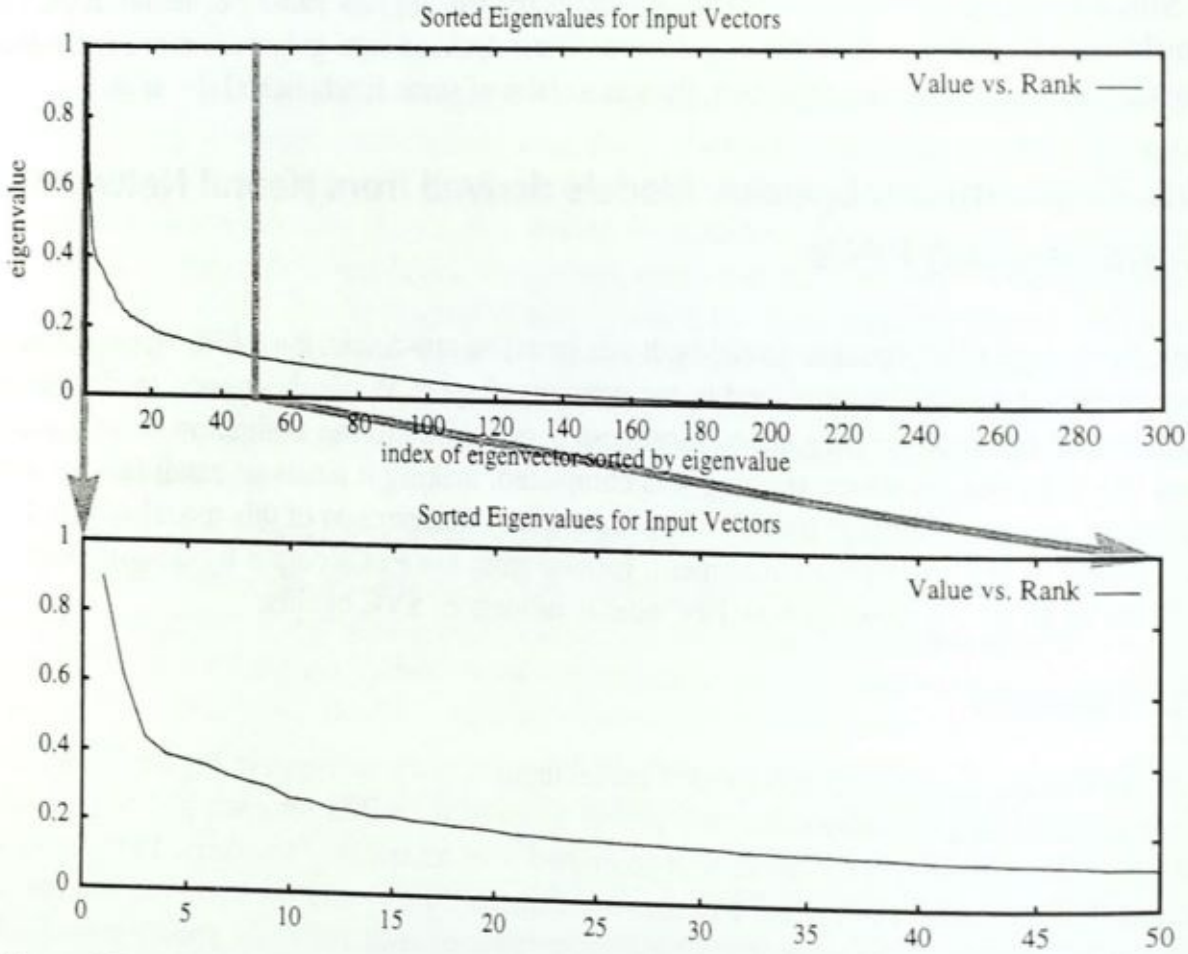


Figure 22: Eigenvalues the input PPC vectors used to construct the SVC, sorted by size. The lower graph plots the same data as the upper graph, but the domain is limited to the first fifty eigenvalues to show detail.

many reasonably large eigenvalues (up to about the 150th or so) — many more than the number of parameters one would imagine a model of speaker variation to have — suggests that the much of the variation in the inputs, and by implication the PPCs, was noise.

The four kinds of speaker model were compared both by calculating the J measure for SVCs from each, and also by measuring the accuracy with which speakers could be identified using a nearest mean match on the SVCs. Table 13 gives the discriminant measure, also graphed in Figure 23, for each of the models, for the training and testing sets. For this experiment, in both the variational and discriminant models, the linear methods outperformed the neural networks — in the discriminant case, by a considerable margin — suggesting that for this training regimen at least, the neural nets had learned, at best, to approximate the linear models. The speaker identification rates given in Table 14 repeated the story told by the J measure; the LDA derived model was more successful than other models in all cases. The NNCompress network (HidNNComp) had a level of performance almost indistinguishable from PCA when measured with the J measure. While the NNDA network (HidNNDA) actually started off, at low dimension, with lower performance than PCA, when the model size was increased to fifteen it was able to learn to outperform PCA, and to almost reach the training, but not the testing, discrimination performance of LDA.

Table 13: Discriminability measures (J) for SVCs from speaker models derived from PPCs produced by neural net bottleneck compression phone models. Larger values of J indicate that the SVCs discriminate speakers more effectively.

	Method	Width							Means
		1	2	3	4	5	10	15	
Train	LDA	3.0371	3.8042	4.2959	4.6840	5.0200	6.2315	7.0465	4.8742
	NNDA	0.9992	1.5292	2.1115	2.7131	2.9118	4.7918	6.0262	3.0118
	PCA	1.6034	2.4196	2.4983	2.6243	2.8015	3.2190	3.6905	2.6938
	NNCompress	1.5705	2.3873	2.4577	2.5670	2.7902	3.2226	3.6838	2.6684
Test	LDA	2.2094	2.7600	3.0328	3.2051	3.3643	3.9509	4.2783	3.2572
	NNDA	0.6925	0.9594	1.4573	1.8756	1.7024	2.7982	3.6317	1.8739
	PCA	1.2162	1.9933	2.0684	2.1477	2.4047	2.7847	3.1924	2.2582
	NNCompress	1.1789	1.9289	2.0118	2.0914	2.4030	2.7911	3.1857	2.2273

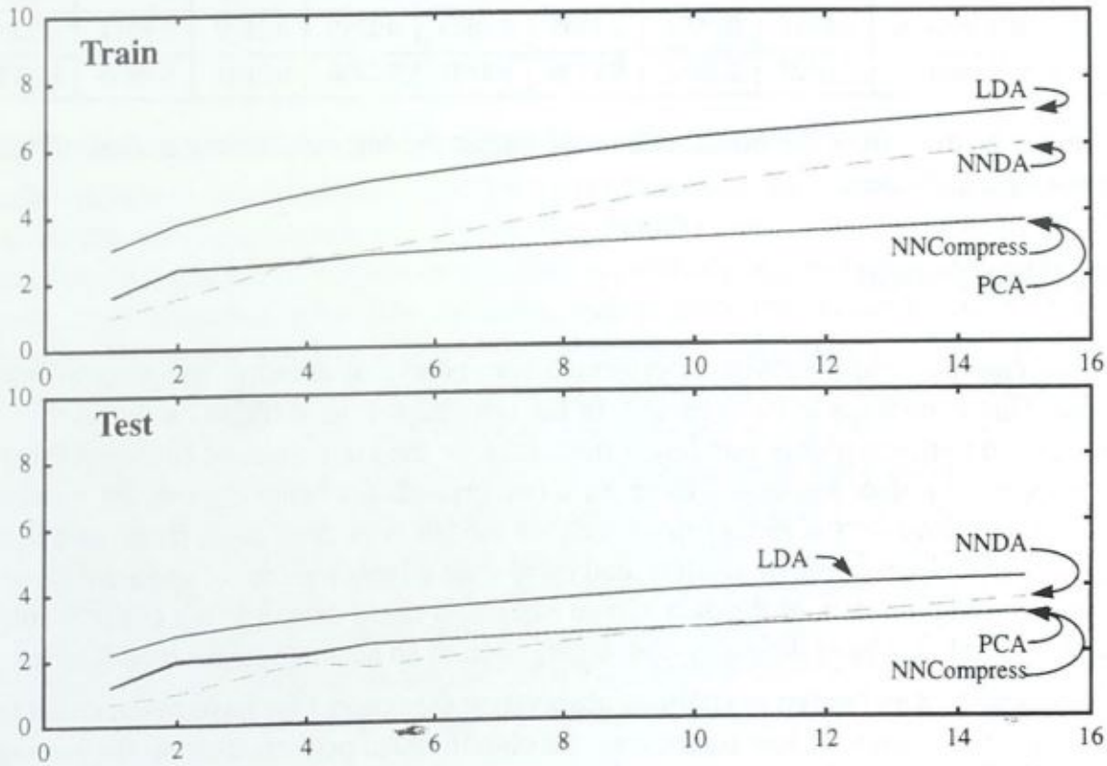


Figure 23: Discriminability measure for the speaker models in Table 13. Performance for NNCompress and PCA is nearly identical in both cases.

For the neural net techniques, Table 12 gives two performance measurements for each network. The "Hid" measurement in each case is the identification rate based on hidden unit activities (i.e. on the SVC). The other is based on the output of the network. The reason both figures are given is to give some indication of how well the decoding layers beyond the bottleneck performed. For the compression network, the reconstituted output vector was no more distinctive for different speakers than the SVC was. For the discrimination network,

Table 14: Speaker discrimination scores for Discriminant models. Score is the average proportion of model vectors for a phone that are nearest to the group mean for their speaker. A score of zero means no speaaker was identified correctly, while a score of 1.0 represents perfect speaker identification.

	Method	SVC Width							Means
		1	2	3	4	5	10	15	
Train	LDA	0.0291	0.1186	0.2452	0.3732	0.4646	0.7612	0.8507	0.4061
	NNDA-out	0.0248	0.0475	0.1000	0.1965	0.2648	0.6592	0.8371	0.3043
	NNDA	0.0189	0.0468	0.0985	0.1941	0.2627	0.6498	0.8275	0.2998
	PCA	0.0218	0.0510	0.0774	0.1118	0.1479	0.3120	0.4460	0.1668
	NNCompout	0.0219	0.0546	0.0785	0.1097	0.1491	0.3105	0.4441	0.1669
	NNComp	0.0215	0.0550	0.0777	0.1076	0.1453	0.3093	0.4442	0.1658
Test	LDA	0.0673	0.1667	0.2595	0.3408	0.4105	0.6266	0.7198	0.3702
	NNDA-out	0.0371	0.0616	0.1243	0.2042	0.2053	0.4911	0.6550	0.2541
	NNDA	0.0333	0.0595	0.1268	0.2021	0.2032	0.4806	0.6438	0.2499
	PCA	0.0532	0.1149	0.1506	0.1709	0.2308	0.3779	0.4932	0.2274
	NNCompout	0.0515	0.1075	0.1405	0.1674	0.2249	0.3807	0.4911	0.2234
	NNComp	0.0508	0.1075	0.1426	0.1681	0.2200	0.3681	0.4935	0.2215

the layers leading from the bottleneck to the output did increase the separation of speakers, but not by a great deal.

4.4. Discussion

For this data the neural networks performed very poorly, at or below the level of the linear model. This is perhaps to be expected. In the last chapter the difficulties in training neural networks to perform highly non linear mappings, of the kind required to outperform linear methods, were noted. Moreover, there were two grounds for believing that the training data used here were noisy: the PPCs from which the models were built came from the poorly performing NNCompress phone models, and there were a large number of apparent dimensions in the covariance matrix of the data. Given extremely noisy data, it is not entirely surprising that the neural nets have difficulty converging even to an accurate linear model of the input.

One consistent and rather mysterious observation the reader may have made about the data is that, for the models of low dimension, the classification performance on the testing set is often higher than the classification performance on the training set, despite the fact that the data are separated using a projection derived from the training set. There is a plausible explanation for this effect. Note first that, for the hidden units perforce, and for the output units approximately, because of their training, the projection space is bounded (by -0.5 0.5 in each dimension. That is, the input vectors are projected in both cases into a unit cube of dimension d (d in 1,2,3,4,5,10,15). Also note that there are 190 training speakers, and 63 testing speakers (k and k', respectively). Since there are 8 670 training and 2 855 testing patterns for this experiment, the average number of patterns per speaker is approximately the same (45.63 and 45.32 respectively). In measuring the classification performance, recall that

the centroids of these 45 or so vectors are taken, for each speaker, and a decision is made whether the speaker centroid nearest a pattern is that of the correct speaker.

Although the projection into the discriminant space is trained on the training data, and should, therefore, separate that data better, imagine for a second that it is equally good in both the testing and the training case, and, in fact, that it positions the centroids to optimally separate the classes, and that the classes have equal variance. In this case, the centroids should be arranged at the centres of the k or k' spheres of radius r and r' respectively that can fit in a unit hypercube of dimension d. Since these radii are hard to find, instead suppose that the unit hypercube is completely divided into k or k' equal regions, each of which will be regarded as a sphere. Then for each dimension d, the following equations hold.

$$1 = k'r'^d \qquad \text{and} \qquad 1 = kr^d$$

That is, as the number of speakers increases, the radius available to each decreases. The variance of the training and testing sets is likely to be the same, but a point that differs from its class's centroid by some amount a, where $r < a < r'$ will be misclassified in the case that there are k classes, but not in the case that there are k'. The expected proportion of cases misclassified for this reason is related to the ratio of r' to r, or

$$\frac{r'}{r} = \sqrt[d]{\frac{k}{k'}} \qquad \text{in this case} \qquad \sqrt[d]{\frac{190}{63}}$$

which rapidly approaches 1 as d, the dimension of the code, increases.

Bridle [personal communication 1995] suggests that classifiers of the kind used here do not distribute the classes uniformly through the classification space, but instead distribute them across the surface of a hypersphere. This is certainly approximately the case for the outputs of the classifier, since they are attempting to place the classes at the vertices of a hypercube. The extent to which it is true for the internal representations of neural networks is not entirely clear, but even if the classes are not distributed precisely across the surface of a sphere, the argument is likely to be similar. The problem in this case is the inverse of the "kissing problem" [mount95, personal communication] i.e. how many patches with a given angular separation can be fit on the surface of a sphere. Unfortunately, analytical solutions to this problem are not known for dimension greater than three[6]. There are approximations that provide bounds for the problem, but they are rather loose. Fortunately, these bounds can be used to show [conway88, mount95] that when placing k spherical patches on the surface of a sphere in a real space of dimension d, the minimum angular separation ϕ between the centres for even the best possible packing, is proportional to the following expression, where

$$\frac{1}{(1-o(1))\sqrt[d]{k}}$$

(o(1)) represents an unknown dependence on d that approaches zero with large values of d.

6. It is known, for example, that in four dimensions, the number of patches of angular separation $\frac{\pi}{3}$, is either 24 or 25, but it is not known which [conway88].

Since this dependence does not vary with k, we find, as before (working with angles instead of radii) that:

$$\frac{\phi'}{\phi} \sim d\sqrt{\frac{k}{k'}}$$

While this argument is approximate, and, in particular, does not yield a quantitative measure of expected misclassification rates for the actual data, it demonstrates that for low dimensional spaces, the mere fact that there are more training speakers than testing speakers will inflate the relative misclassification rate of training speakers. For higher dimensions, this effect diminishes, and the effect of the fact that the discriminant projection is trained on the training, rather than the testing, speakers should be expected to dominate. This appears to have been the effect observed in the data.

4.5. Experiment: Speaker Models derived from NNDA PPCs.

After all the experiments comparing training methods for phone models had been com-

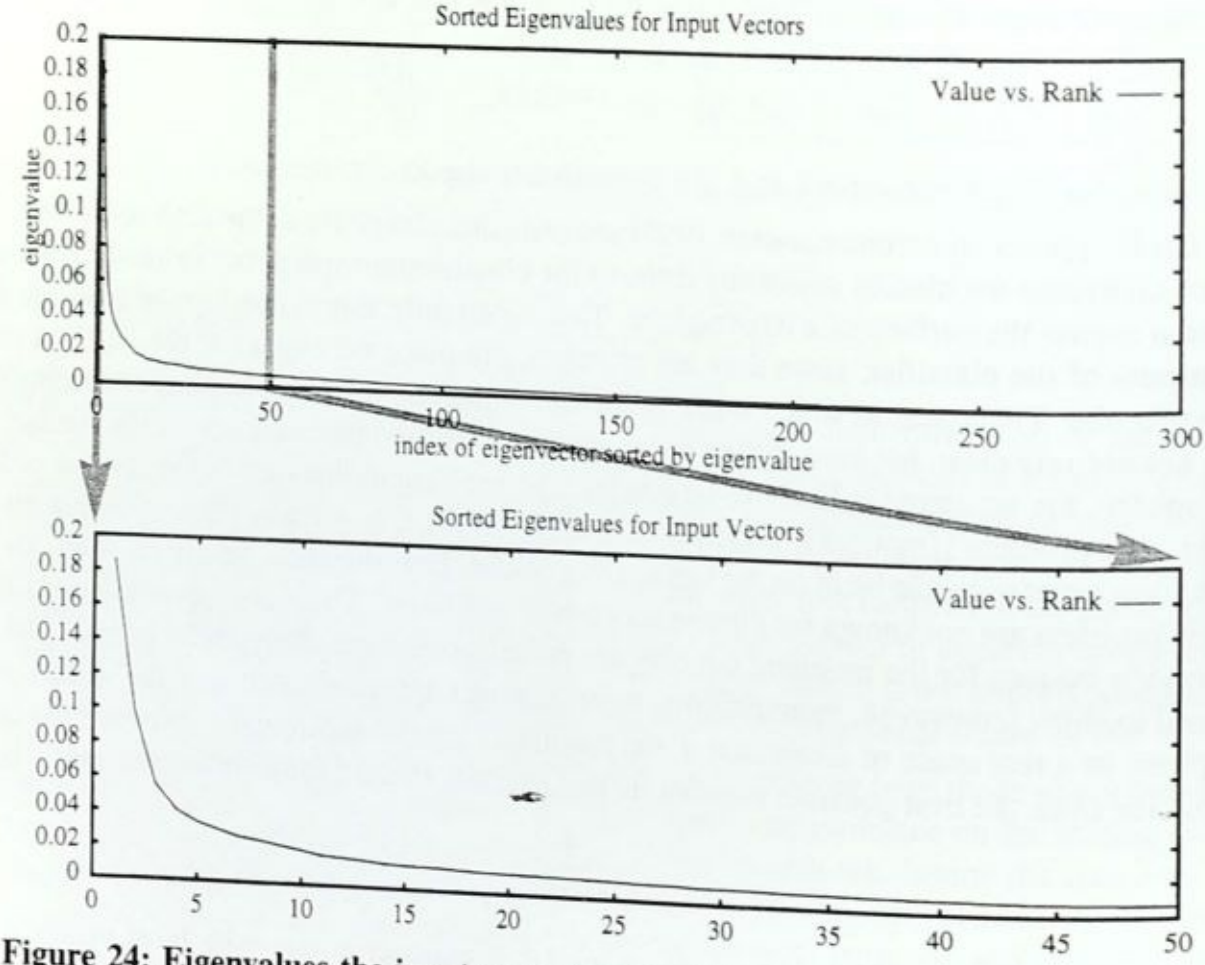


Figure 24: Eigenvalues the input vectors derived from NNDA PPCs, sorted by size. The lower graph is the same as the upper graph, but only the first fifty eigenvalues are shown, to make the detail in this region more visible.

pleted, it appeared that, although the advantage over linear discrimination was somewhat slight, the PPCs derived from the Neural Net Discriminant models seemed to have the best

performance with respect to the chosen criteria. A second experiment was therefore performed in an identical manner to the one described above, substituting these improved PPCs in place of the NNCompression ones. Since this was the only change in the way the experiment was done, repetition of the description of the experimental method is omitted here in favour of proceeding directly to the results.

4.5.1. Results

Figure 22 plots the eigenvalues of the covariance matrix for the input vectors formed by

Table 15: J Discriminability measures (J) for SVCs derived from NNDA PPCs.

	Method	SVC Width							Means
		1	2	3	4	5	10	15	
Train	LDA	3.8410	4.8839	5.5951	6.1682	6.6645	8.3703	9.4642	6.4267
	NNDA	1.2729	1.7766	2.5474	3.2388	3.7107	5.6766	7.2345	3.6368
	PCA	2.5132	3.3117	3.8949	4.2093	4.6437	6.2049	6.9899	4.5382
	NNComp	2.4772	3.2997	3.9259	4.2260	4.6618	6.2263	6.9660	4.5404
Test	LDA	2.5584	3.2983	3.6517	3.9665	4.2912	4.9929	5.4525	4.0302
	NNDA	0.6650	1.0654	1.5408	1.9233	2.1321	3.1885	3.9276	2.0632
	PCA	2.1182	2.8221	3.4018	3.6836	4.0170	5.2018	5.7125	3.8510
	NNComp	2.0729	2.8082	3.4257	3.7044	4.0494	5.2030	5.6931	3.8509

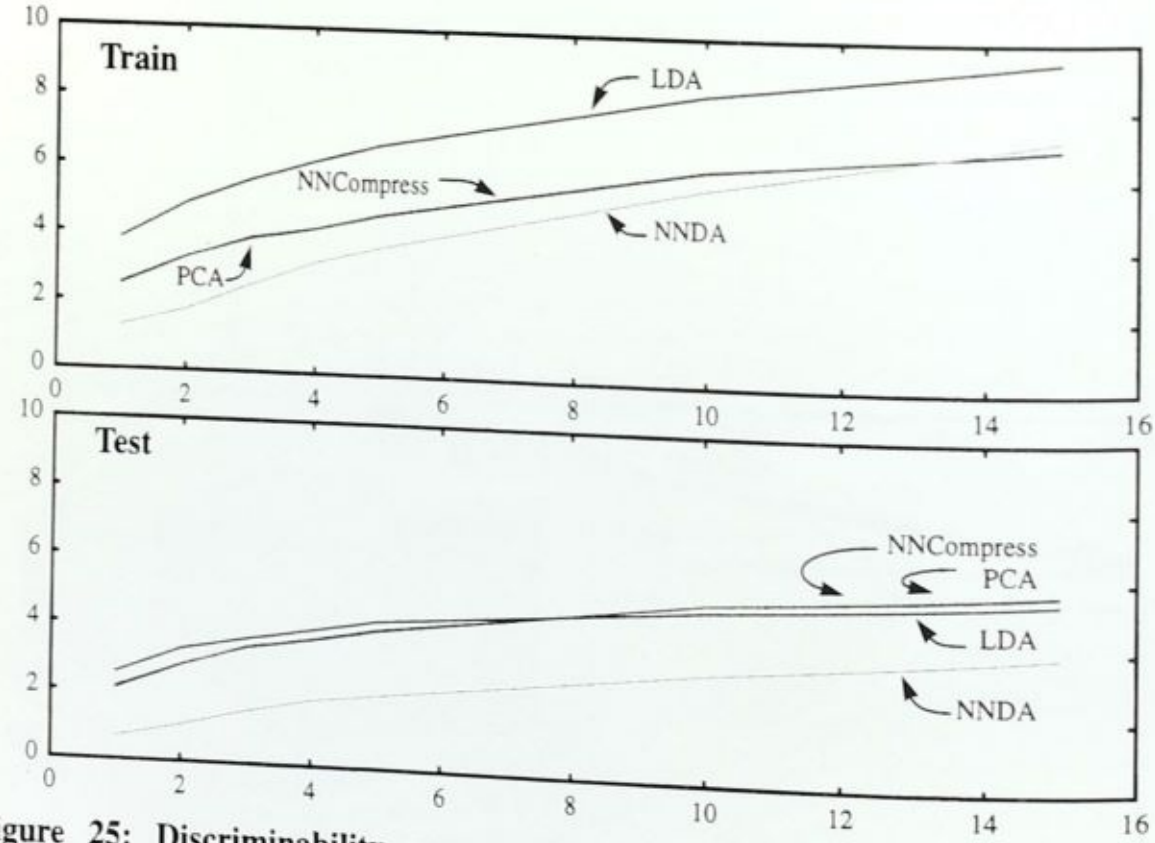


Figure 25: Discriminability measure (J) for speaker models. The PCA and NNCompress results coincide, obscuring the plot. NNCompress and PCA are nearly identical in both cases.

concatenating NNDA based PPC vectors. These values fall off much more sharply than in the previous experiment, suggesting that there was little variance in the PPCs unaccounted for after the first ten to fifteen or so eigenvectors. In particular, the relatively small values that occurred in the tail suggest that these PPCs are less noisy than the ones produced by the compression networks.

Table 16: Actual speaker classification scores for SVCs derived from NNDA phone models. Scores are correct speaker identification rates using nearest centroid classification.

	Method	SVC Width							Means
		1	2	3	4	5	10	15	
Train	LDA	0.0446	0.1810	0.3468	0.5190	0.6404	0.8651	0.9140	0.5015
	NNDA	0.0262	0.0549	0.1349	0.2555	0.3772	0.7646	0.8905	0.3577
	PCA	0.0280	0.1012	0.1664	0.2311	0.3303	0.6589	0.7791	0.3279
	NNComp	0.0276	0.1001	0.1608	0.2245	0.3248	0.6806	0.7916	0.3300
Test	LDA	0.0739	0.2494	0.3856	0.4928	0.5891	0.7454	0.8165	0.4790
	NNDA	0.0336	0.0844	0.1447	0.2067	0.2669	0.5695	0.7184	0.2892
	PCA	0.0739	0.1797	0.2785	0.3520	0.4357	0.6946	0.7884	0.4004
	NNComp	0.0651	0.1828	0.2687	0.3370	0.4270	0.7142	0.8046	0.3999
Output layer discrimination scores for neural nets									
Train	NNDA	0.0334	0.0557	0.1356	0.2536	0.3787	0.7714	0.8897	0.3597
	NNComp	0.0304	0.1007	0.1661	0.2273	0.3300	0.6585	0.7682	0.3259
Test	NNDA	0.0448	0.0834	0.1426	0.2133	0.2729	0.5741	0.7268	0.2940
	NNComp	0.0680	0.1828	0.2827	0.3464	0.4340	0.6963	0.7842	0.3992

As in the previous experiment, the discriminability measure J for the four types of models over seven SVC dimensions is tabulated, in Table 15, and plotted, in Figure 23. The measures were substantially higher than those for the models based on the PPCs from compression networks. Once again, the speaker models derived from the PPCs using LDA achieved the greatest amount of separation of the speakers in the training data, and for all but the ten- and fifteen-dimensional models on testing data. In the latter cases, the PCA and NNCompress based SVCs were more widely separated. The NNDA models were strikingly unsuccessful, having smaller J measures than the other models in all testing cases, and in all but one training case. Even these measures were higher than for corresponding models derived from NNCompress PPCs, however.

Following the pattern of the last experiment, speaker classification scores derived using nearest neighbour classification are given in Table 16. Classification scores for this model based on NNDA PPCs are considerably higher than for the previously discussed NNCompress PPCs, most markedly for the variation-based SVCs. When NNCompress PPCs were used to build the speaker models, classification performance for these SVCs was around half that for the LDA models. Using NNDA based phone models, the testing performance for these SVCs (PCA and NNComp Hidden) is over 80% of the LDA performance, with the difference narrowing for higher dimensional models. Classification performance for the neural discriminant model confirmed the story told by the J measure; although the classifier network learned to separate training speakers more than the variational methods, it failed to generalise, producing a worse basis for classification of testing speakers than either the PCA or NNCompression SVCs. The other methods, classification using LDA, and capturing the validation using neural compression or PCA all preformed similarly on testing data.

4.5.2. Discussion

As might have been expected from the differences in performance, on similar measures, between the PPCs used to form the models, the SVCs based on NNLDA PPCs contained more speaker distinguishing information than the SVCs based on NNCompress PPCs.

More interesting, perhaps, is the observation that these original PPCs seem to have captured, in their ten dimensions each, much of the information that distinguishes speakers, or rather, much of the speaker-distinguishing information that is contained in the original whole-phone representations. While the discriminant methods were able to find combinations of these PPC vectors that were better at distinguishing training speakers than the principal components, these combinations did not greatly outperform projections onto ten or fifteen principal components for testing speakers. It appears that most of the separation between speakers that can be found in ten dimensions is already available in any one of the frequently occurring NNDA based PPCs.

4.6. Experiment: Speaker model based on Pattern completion neural nets

In the experiments described above, the training techniques used aimed either to learn to classify the training speakers, or to reproduce the partial input vector exactly on the output. In the former case, there was a risk of concentrating too heavily on qualities of the training speakers and consequently of failing to generalise to test speakers. More generally, there was a risk of discarding important information about voice quality that did not help much with speaker identification. In the latter case, where the system was learning an identity function, it was possible that the system would both over-constrain the model to be learned, when there was missing data, and under-use the available training data as follows: As the introduction of this chapter pointed out, during training, the entire body of training data for a particular speaker can be used to construct target patterns for a "compression" neural network. There is no good reason to limit targets to just the subset of the data that "has already been heard" and that will be presented on the inputs. It is also to be hoped that the compression networks are doing what they are designed to do — pattern completion — and that this can be used to improve the incomplete input vectors received for a speaker.

In this experiment, the classification networks were left aside, and, still using the NNDA based PPCs as inputs, neural network compression networks were trained applying the following techniques:

- **Maximally instantiated targets:** Instead of having the targets be the subset of PPCs presented to the inputs, with an identical estimate, the mean, for missing values, a randomly chosen instance of every phone PPC available for the speaker was used in the target vector.
- **Minimally constrained training**: Instead of backpropagating error from those target units for which no PPC is available for the speaker, training the network to output the overall mean, the output for those parts of the vector were left untrained, free to output what might be a better estimated value of the missing PPC.

- **Recirculation**: Since the networks were being trained to produce estimates of PPCs for unheard phones, it was reasonable to hope that they would produce more consistent SVCs from PPC subsets on the network inputs, if the network outputs were recirculated to the PPC inputs for unheard phones. The SVC used was the activation pattern of the bottleneck layer after some number of iterations of this process.

4.6.1. Method

In this experiment, neural networks were trained to do completion using both of the first two of these techniques, both using (CR), and without using (C) the recirculation technique. Networks with recirculation were also trained on input vectors with two different average numbers of missing PPCs (CR and CR2).

Targets were constructed by choosing a random starting point in the list of PPCs for a speaker, and looking forward from that point, adding the first instance encountered for each phone to the target vector, until either all 50 phones had been found, or the starting point had been reached again. On average, 99.4% of target phones where available[7,8]. Missing phones in the target vector were replaced by a marker value that was used to prevent error-back-propagation from the corresponding outputs. Inputs patterns were chosen by randomly choosing thirty (C, CR) or sixty (CR2) PPCs at random from the speaker, with replacement, and inserting these into the inputs. This procedure resulted, on average, in 36% and 54% respectively of the fifty phone PPC sections of the input vector being filled. Unfilled inputs were marked, and were replaced by overall PPC means, and later, in the case of the recirculating networks, by estimated PPC values from the network outputs.

Since the training patterns were assembled internally by the training program from PPCs, it was no longer necessary to discard 80 percent of the training patterns as had been done previously to save space. The difference in number of training patterns between this and the previous experiments was compensated for by reducing the number of training epochs so the total number of pattern presentations was the same in both cases.

4.6.2. Results .

The discriminant-space volume measure, J, and the nearest centroid classification scores for the three networks are given in Tables 17 and 18, respectively. A comparison with Table 15 shows that on training data, the recirculating completion networks (RC and RC2) have a larger discriminant volume than all other models, including the one based on LDA. The simple completion network (RC) outperformed all but the LDA based model on training data. On testing data, the performance of these networks is even better - all of them produce a larger discriminant space than the models of the previous experiment, and this space is

7. This number is rather high for the number of sentences used. It can be explained by the fact that only the fifty most frequently occurring phones were being used, and by the fact that the TIMIT database was deliberately designed to be phonetically balanced.

8. While this procedure results in correlated sets of target pronunciations from a speaker, it seemed a reasonable trade-off against the time that would be required to search the speakers PPCs for each phone separately from different starting points, for each pattern presentation.

Table 17: The discriminant volume measure J for the three more complex neural network based speaker models. Method *C* represents completion training with partial input patterns and maximally completed target patterns. *CR* is similar, with the addition of two iterations of output values to the inputs of "missing" input phones. *CR2* is similar, except that CR filled in, on average, 36% of its inputs during training, and CR2 filled in 53%.

Mode	Method	1	2	3	4	5	10	15	Mean
Train	C	4.3011	5.4299	5.8392	5.9433	6.0667	7.4008	8.1892	6.1672
Train	CR	4.8560	6.4519	7.2323	7.6002	7.9201	9.3040	9.9626	7.6181
Train	CR2	4.6046	6.3648	7.1177	7.5319	7.7648	9.4174	9.9392	7.5343
Test	C	3.4051	4.4627	4.6852	4.7270	4.9821	5.7830	6.1780	4.8890
Test	CR	3.5536	4.8766	5.3294	5.5600	5.8318	6.7531	6.9626	5.5524
Test	CR2	3.4885	4.7620	5.2373	5.4805	5.7804	6.8074	7.0387	5.5136

Table 18: Correct speaker identification rates using nearest centroids for three more complex neural net compression training regimens. The conditions are those described in Table 17.

		Method	1	2	3	4	5	10	15	Mean
Model	Train	C	0.0331	0.1333	0.2647	0.4066	0.4986	0.7871	0.8488	0.4246
		CR	0.0376	0.1416	0.2541	0.4047	0.5253	0.7865	0.8345	0.4263
		CR2	0.0400	0.1605	0.2880	0.4434	0.5673	0.8196	0.8645	0.4548
	Test	C	0.0782	0.2455	0.3323	0.4472	0.5686	0.7650	0.8321	0.4670
		CR	0.1007	0.2618	0.3475	0.4514	0.5891	0.7463	0.7865	0.4691
		CR2	0.0994	0.2932	0.3879	0.4811	0.6215	0.7752	0.8173	0.4965
Output	Train	C	0.0383	0.1303	0.2593	0.4145	0.5323	0.8050	0.8531	0.4333
		CR	0.0344	0.1197	0.2375	0.3839	0.5183	0.8085	0.8683	0.4244
		CR2	0.0373	0.1380	0.2681	0.4151	0.5411	0.8193	0.8740	0.4418
	Test	C	0.0794	0.2357	0.3383	0.4537	0.5954	0.7854	0.8346	0.4746
		CR	0.0860	0.2272	0.3236	0.4399	0.5767	0.7815	0.8368	0.4674
		CR2	0.0958	0.2541	0.3456	0.4625	0.5945	0.7971	0.8502	0.4857

larger by a considerable margin. Within the recirculating nets, the proportion of inputs originally available (36% vs. 54%) does not seem to make a substantial difference.

When looked at with respect to speaker classification accuracy, the advantages of these networks are less clear. All three models had similar performance at classification of speakers using nearest centroids, and all outperformed the neural net models used in the previous experiment[9]. The networks generally reached almost the same level of classification accuracy as LDA, but they did not, with one exception, exceed it. The CR2 network had a higher

mean classification accuracy than the LDA classifier, but this advantage was not consistent across dimensions.

4.6.3. Discussion

The fact that the recirculating completion networks produced a larger discriminant space than the other methods, including the stubbornly successful linear discriminant analysis, suggests that they were doing well at concentrating inputs from a single speaker into a small region of space, and pushing inputs from different speakers apart. Although their success in doing this did not generally translate into a higher speaker identification rate than that given by LDA, the fact that they reached the same rate on test data is impressive in itself, since, while the training in LDA was strongly supervised, the completion networks were weakly supervised; the networks were not being given an explicit direction to separate the speakers — they were only being told to predict PPCs from the same speaker. The discriminant training of the NNLDA PPCs caused these PPCs from different speakers to be differently distributed, and the networks were able to take advantage of this to learn an overall speaker model that separated speakers well, without having to be told that this was a goal of learning.

4.7. Perceptual relevance

Ultimately, it would be desirable if the dimensions onto which the models project speakers were to correspond to some quality that human beings regard as being important to voice quality. Unfortunately, a well defined set of descriptions of voice quality is not available, and even if it were, one would be hard pressed to label the entire database with them. The labels that are readily available in the database are sex and dialect region. To see whether either of these qualities was captured by the two most successful models, the LDA and CR2 models based on NNDA PPCs, a plot was generated of the mean model value computed over the SVC state after each of ten PPC additions, starting from the point at which one hundred phones had already been heard. Figure 26 shows these plots for both training and test set data labelled with the sex of the speaker. Speaker sex was apparent from the model in almost all cases, and was represented by the first of the two model dimensions[10]. For the LDA model, apparently, and perhaps not surprisingly, the speaker's sex was the most distinctive feature of their voice quality. In fact, although this is not shown in these plots, models separate for sex in most cases after only three phones had been heard from a speaker.

Figure 27, on the other hand, shows the same SVCs labelled by the dialect region of the speaker: where in the US they grew up. If regional dialect is an important determiner of voice quality, it was not captured by the strictly segmental model that has been adopted here. It is likely that the major effects of dialect are felt on prosody and phonology. While it is evident these components of accent are important in explaining perceived voice quality, it has

9. The NNCompress network and the PCA projection in the previous experiment reached similar classification performance in models of dimension fifteen. This is not too surprising, perhaps, in light of the fact that Figure 22 suggests that most of the variability of the original data can be accounted for in fifteen dimensions. The higher performance at lower dimensions of the present networks suggests that their encoding is more efficient.

10. This is meaningful in the case of the LDA model, where the dimensions are sorted by their ability to distinguish speakers, but fortuitous in the case of the neural net model, since there is no intrinsic ordering on the hidden layer units in which the model is formed.

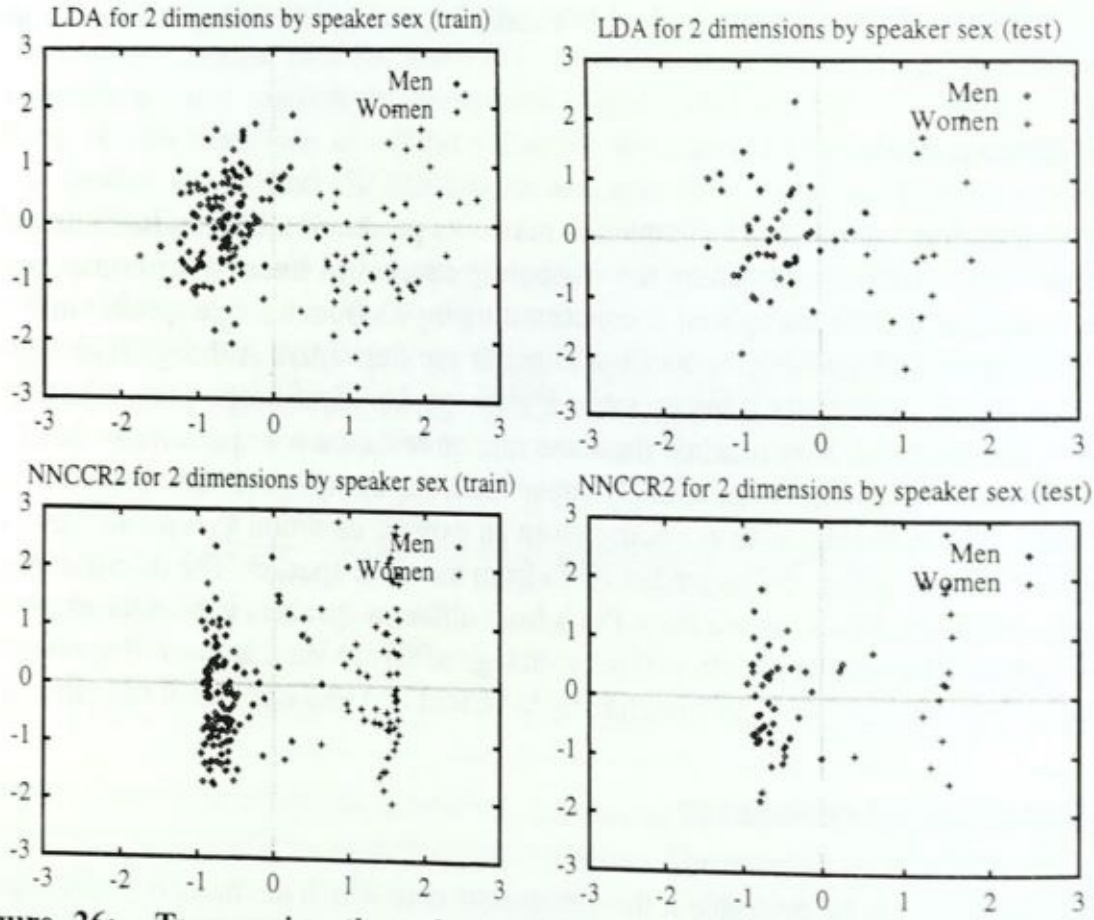


Figure 26: To examine the relationship between the models and perceptually relevant components of voice quality, two dimensional speaker models were plotted here labelled for speaker sex.

not been practical to include their investigation in the scope of this thesis. It is worth noting though, if the phone model had been extended to include phone duration, one might have expected to measure an effect of dialect on that model component.

4.8. Rapidity of formation

One of the main aims in building these models was to approach the rapidity with which human beings adapt to new speakers. It was consequently worth attempting to measure how rapidly the speaker models were approaching a final stable SVC value that represents a speaker's voice. The question was whether the SVCs formed after some small number of phones had been heard were similar to those formed after many phones had been heard. Figure 26 plots the components of the two-dimensional CR2 model after five phones had been heard from a speaker against the same component after one hundred phones had been heard.

Although there is by no means a total agreement between the models at these times, there was a strong correlation between the SVCs produced at the two times for both SVC components. Table 19 enumerates significance tests for the correlation between SVCs formed after five and two-hundred phones had been heard from a speaker, along with similar tests for two other pairs of times [becker88, chambers93]. Even for SVCs formed when the third phone was added to the input, the SVC was approaching its final position in speaker space, as demonstrated by the positive, and highly significant correlation between it and SVCs formed

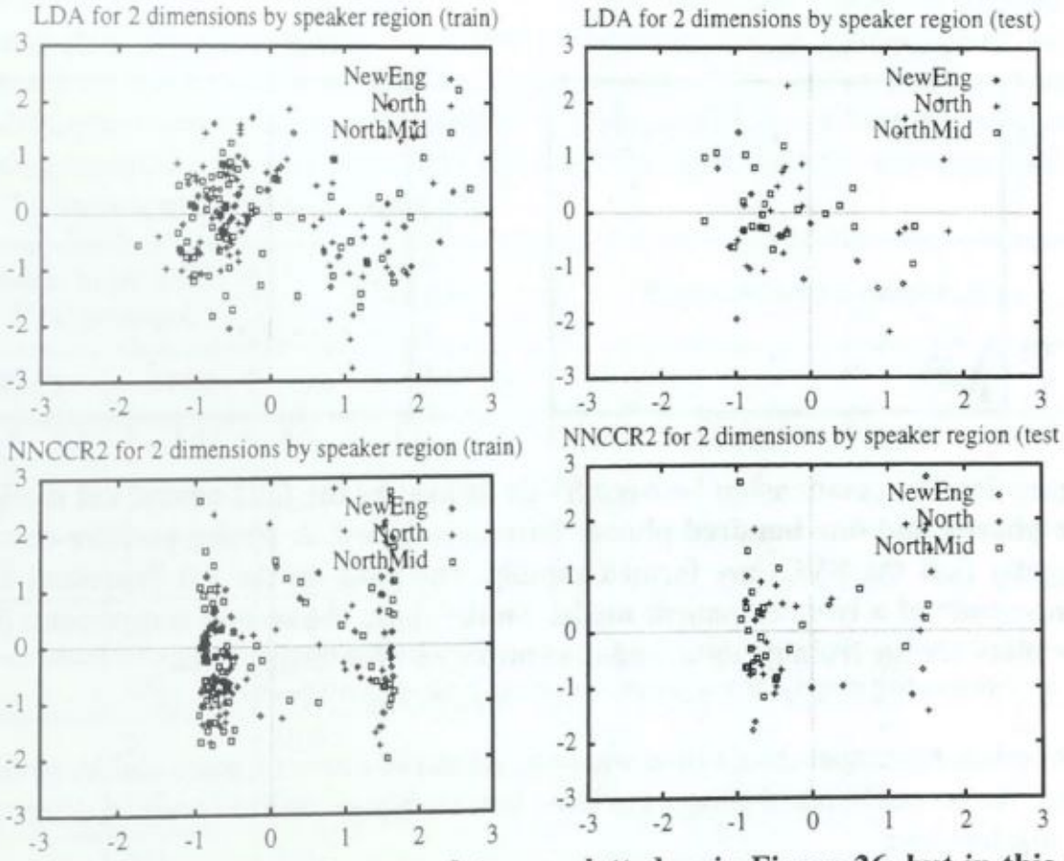


Figure 27: In this figure the same data are plotted as in Figure 26, but in this case they are labelled for the geographical region in the US where the speaker grew up. There is no readily apparent component of the models that corresponds to regional dialect.

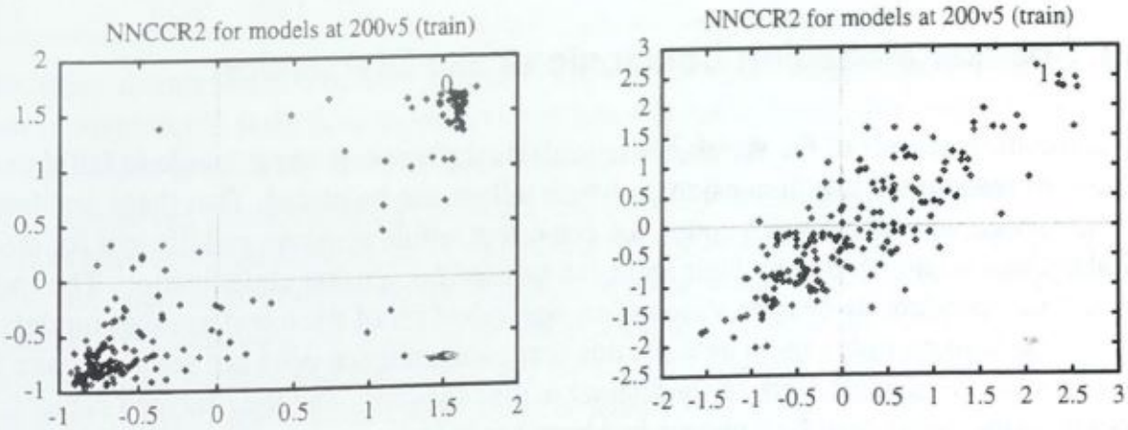


Figure 28: The relationship between SVCs formed by the CR2 neural net model after five phones and one hundred phones have been heard. A strong positive correlation suggests that the SVCs are formed rapidly. The plots on the left represent the first component[a] of a two component model, on the right, the second component. The top row plots are for training data, and the bottom row for testing data.

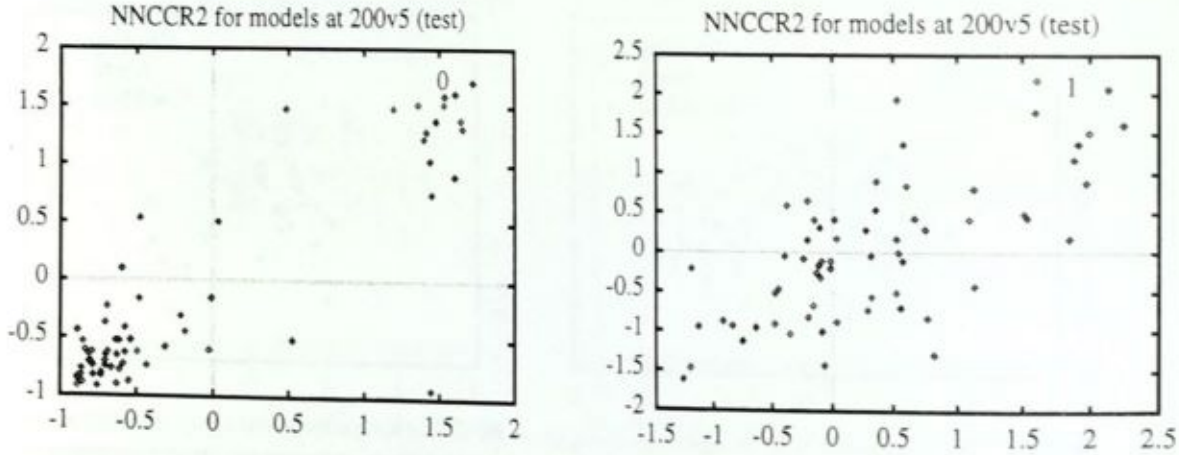


Figure 28: The relationship between SVCs formed by the CR2 neural net model after five phones and one hundred phones have been heard. A strong positive correlation suggests that the SVCs are formed rapidly. The plots on the left represent the first component[a] of a two component model, on the right, the second component. The top row plots are for training data, and the bottom row for testing data.

a. corresponding roughly, as was pointed out above, to the speaker's gender.

from much more speech. As time went on, the models became more stable, to the point where the second hundred phones produced little change in the SVC position determined by the first hundred.

It should be noted that these measurements were made on the two-dimensional models that performed rather poorly on speaker classification because it was straightforward to do so. One would, of course, expect higher dimensional models to be highly correlated within speakers over time. In this case of multidimensional data, instead of the simple correlation measurement used here, one would use the similar canonical correlation analysis.

4.9. Speaker Modelling Conclusions and Discussion

By combining models of the variability in individual phones, it was possible to build overall spaces, of reasonably low dimension, in which talkers can be placed. That these positions in speaker space, or speaker voice codes, are consistent within speakers and distinct for distinct speakers was demonstrated by their ability to be used for speaker classification. The models are text independent by design; although an equivalent set of fixed text speaker models was not available to compare them against, this text independence does not seem to have been too harmful: the models were fairly distinct across speakers, and they formed rapidly. The models formed after only four phones had been heard from a speaker were highly correlated with those formed after one hundred and four phones had been heard. Clearly, which phones a speaker uttered within the first four was not a critical determiner of the SVC.

Repeating the observations of the previous chapter covering phone variation modelling, there was no consistent advantage to be gained from the fact that the neural network based models were capable of non-linear encodings. While the most complex of the neural net based models matched LDA in supporting discrimination, while not requiring discriminant training, this result was obtained when the SVCs were constructed from PPCs that were

Table 19: Tests for the consistency of SVCs generated after different numbers of phones from the same speaker, measured as correlation. The p values given are the probability that t exceeds the given value if the correlation, r, is not positive. Speaker models generated from smaller numbers of phones (SVC1) are highly predictive of those generated from more speech in all cases. The shaded entries correspond to the SVC components plotted in Figure 26.

Phones heard before SVC generated		Tested data		Pearson's Product Moment test			
SVC1	SVC2	Set	SVC Cpt	r	t	df	p
100	200	Train	0	0.974	58.65	188	0
100	200	Train	1	0.888	26.43	188	0
100	200	Test	0	0.962	27.72	61	0
100	200	Test	1	0.881	14.54	61	0
5	200	Train	0	0.928	34.04	188	0
5	200	Train	1	0.816	19.34	188	0
5	200	Test	0	0.872	13.89	61	0
5	200	Test	1	0.683	7.31	61	0
2	100	Train	0	0.894	27.33	188	0
2	100	Train	1	0.766	16.35	188	0
2	100	Test	0	0.690	7.45	61	0
2	100	Test	1	0.573	5.46	61	0

inherently discriminant. For the input encodings of the speech signal used in this thesis, at least, conventional statistical techniques produced speaker spaces that were as effective at distinguishing speakers as those produced by neural networks.

It is, moreover, likely that the PCA based model can be improved on. Since there is no distinction between inputs and targets in PCA, there is little choice when using this technique but to use just the overall phone model mean as an estimate of the value of missing data. However, by construing the problem as one of finding a linear least square fit between the partially filled vectors and the completed ones, this deficiency can, perhaps be reduced. Doing so, using singular value decomposition, should be a useful addition to the set of non-neural models used as reference points when future work provides an improved set of features on which to base models.

At present, the degree of within speaker variability in all the models developed here, limiting their ability to be used even for accurate speaker identification, suggests that the voice models used by human beings are based on far more specific and reliable voice features than the ones supporting the statistical models described here.

4.10. Applications

Up to this point, the main concern has been with constructing a description of a speaker's voice in terms of its relation to a model of how voices vary. This has been done by constructing variation spaces for speech segments, and assembling them into an overall speaker space.

By some measures, at least, this space has the qualities that it was designed to have. It distinguishes speakers placed in it, it corresponds to a perceptually relevant speaker distinction in at least one dimension, and it allows rapid identification of a speaker's position after only a few phones have been heard. Since the model packages speaker information in a compact vector, it is technically straightforward to integrate the information it provides with other inputs to speech processing systems that use neural networks or statistical learning.

By other measures the models were less successful, since they do not appear to have the within speaker stability and descriptive and discriminant power that are available in the models human beings were presumed to have. In part to better measure the power of the current models, and in part to explore what could be achieved if higher quality voice models were available, some speech technology related applications of speaker modelling were explored.

The next two chapters describe efforts to integrate speaker information with two such speech processing systems. In the next chapter, a study is made of the feasibility of applying speaker models to the problem of quickly adapting speech recognition systems to a user's voice. In the chapter that follows, the speaker models derived here are applied to the problem of synthesising speech with voice quality similar to that of the modelled speaker.

Chapter 5. Speaker Models and Speech Recognition

The original conception of this thesis was based, in part, on the notion that speaker differences were self-evidently a barrier to successful speech recognition, and that furnishing speaker information to a classifier would improve its performance. It was presumed that the main question to be addressed was how one could derive this information about a speaker rapidly and accurately.

In fact, although initial experiments yielded speaker models which formed rapidly and distinguished fairly well between speakers, the goal of making the recogniser make any use at all of these models proved frustratingly elusive.

The next section briefly reviews some of the work on speaker adaptation that had led to the expectation that it would be possible to successfully apply the voice models to this task. This expectation was explored in pilot experiments, described in the following section, with speech from a very small vocabulary. The results of these experiments, although unspectacular, had seemed promising and led to an attempt to extend the technique to a larger database. This attempt, described in §5.4, was unsuccessful even when perfect speaker information was used; it wasn't just the case that the information that could be derived from a speaker's speech and presented as an SVC wasn't particularly helpful — it turned out that it was very difficult to make knowledge of speaker identity help at all. This lack of success led to a series of experiments comparing the experimental setup used with that of experiments in the literature where large gains from speaker adaptation had been obtained. The fact that it was possible to replicate this gain, for the same data, but not possible, using any of the techniques tried, to extend the gain to a more realistic database, prompted a further series of experiments that attempted to diagnose the source of the difference. These experiments, described in §5.8 led to the conclusion that, for some databases, speaker information can improve recognition performance, but that, with the neural network recognisers used, these improvements were under rather more constraints and rather less spectacular than the HMM literature on adaptation might have lead one to expect.

Since the goal of the thesis was speaker modelling, and since it appeared that the best model that could possibly be derived for a speaker's voice — or, indeed, a perfect voice model — might not have an appreciable effect on recognition performance in the recognisers that were available, work on using recognition as a test application was abandoned. Voice conversion was selected as a more transparent target application, as described in Chapter 6.

5.1. A Glance at the Speaker Adaptation Literature

The work described in this chapter is not, of course, the first attempt at improving the accuracy of a speech recogniser by taking speaker characteristics into account. Although there has recently been some related work in this area, e.g. [cox93] which describes the use of interphone regression models to achieve similar purposes, the novel feature of the current

work is its exploration of the possibility that a permanent model of the variability of speakers can be built, and that model can be used to make adjustment to a new speaker happen more rapidly [witbrock92].

A number of speaker adaptation schemes were described in Chapter 1 and in particular in section 1.4.2 on page 20. In that section, speaker adaptation schemes were discussed in terms of the models of variation they implied.

The majority of speaker adaptation schemes, and especially the ones that have proved useful in working systems, have involved an off-line step in which the system is adjusted to the new speaker. This can be done using some fixed enrolment speech that is used to set the parameters of an acoustic normalisation [e.g. leggetter94, zhao93, lee93, rigoll89]. In this case, knowing which parameters to adjust is reasonably straightforward, since the speech unit, and the state within that speech unit, is easy to identify for a frame of the enrolment speech.

In other schemes, multiple passes are made through the speech with the first pass serving as an opportunity to either to adapt the system parameters immediately [e.g. hild93 "tuning in"], or to gather long term statistics about the speech that can be used to prepare the system for a subsequent pass. For example, in their system for the 1995 ARPA HUB4 evaluation, IBM [gopalkrishnan96] re-estimated Gaussian parameters from the nearest of a set of speaker-specific HMM recognisers in a preliminary pass through the speech, before doing the final decoding. The Abbot group at Cambridge did a similar initial pass to set parameters on a linear normalisation network at the input to their hybrid connectionist-HMM recogniser [kershaw96]. It should also be noted that Cox and Bridle's *RecNorm* system [cox89,90, bridle91] could also be used in this two pass mode, with the first pass used to estimate the spectral normalisation parameters.

Although the majority of schemes have concentrated on adapting to the filter characteristics of the vocal tract (e.g. [payan93]), this is not the only source of variation that must be accounted for. Blomberg [blomberg89], for example, describes an unusual recogniser that uses alignment of speech with synthetic reference frames generated by a model of speech production. In this system, the synthetic reference frames are generated after tuning parameters of a model of the speaker's glottal source. In the small experiments reported on in this paper, glottal source adaptation more than halved the recognition error rate.

In the current work, of course, the attempt is made identify the speaker within a single unified speaker space, for the reasons identified in the chapter on speaker modelling. This is the chief regard in which this work differs from Cox's [cox93] sensible work with interphone regression models of variability. Because the output of the speaker modelling system is a point in speaker space, the application to speaker adaptation shares similarities with multi-speaker systems (e.g. [hild93], [watrous93]). In fact, as it will turn out, the course of experimentation dictated that the majority of this chapter be devoted to an examination of when such multi-speaker systems can be made to use speaker information.

5.2. Preview of Experimental Sequence

In the course of the work described in the chapter, it became clear that, despite some initial promise, it was difficult at best to persuade speech recognisers to make use of speaker iden-

tity information. The bulk of the experimental work described here was done in an attempt to discover the source of this difficulty, so that it could be corrected. In particular, considerable work was done to explore the differences between the cases where speaker ID information was not of use to recognisers, and seemingly similar experiments reported in the literature in which it had had a large effect. Some of these latter differences were explained by the ability of recognisers with a wide input window of complete frames of speech to infer the information that could have be provided by speaker identity. It remains to be discovered, however, what speaker characteristics are learned by speaker dependent recognisers that enable them to perform better than speaker independent systems, and why these characteristics cannot be usefully specified by a speaker identification.

The pilot experiments in using speaker codes in recognition were done using a French Digits database. In these experiments, described in Section 5.3, providing a speaker code to an MSTDNN recogniser allowed it to perform substantially better than when it was given an identical, average, "speaker code" for all speakers. This was a promising result, although difficulties explored in later experimentation were presaged by the fact that a similar recogniser that was never trained to use speaker information had a performance intermediate between the speaker-code-using network given speaker codes, and the same network deprived of them. This was early evidence that the speaker codes were, in part, replacing information that the recogniser could, if necessary, derive from the speech itself.

Despite this, the partial success of the pilot experiment with digits led to an attempt, described in Section 5.4, to extend to use of speaker codes to a larger, spelled-word, database. In this case, the speaker information was presented in what should have been a more easily digestible form: each speaker was identified by a unique input unit in the network. Despite this, the network with speaker information performed identically to a speaker independent recogniser. A review of a similar experiment with a large database [hild93], suggested that this problem was not unique. However, in work that had, in part, motivated this thesis, Watrous [watrous93] had found a large effect of speaker ID for a formant-based vowel classification task. If speaker codes were to be made useful to recognisers, it would be necessary to discover how these tasks differed, and whether the conditions that made speaker ID useful to the vowel classifier could be duplicated in the larger-scale systems.

The original vowel classification experiments had used networks with second-order, multiplicative, connections that allowed the speaker identity to modulate the formant inputs. The first experiment in this sequence, detailed in Section 5.6.1, was intended to see whether this architectural difference accounted for the success of the vowel classifier over the ordinary back-propagation network used in the spelled-word recogniser. In fact it did not: the normal network actually outperformed the second-order network, besting the highest published recognition accuracy.

Since the point of the exercise was not to use speaker identity in general, but to provide it in the form of a position in speaker space, some experiments followed that looked at the ability of the formant recogniser to use this kind of information. It was worthwhile to know whether speaker models would be useful, if the salient differences between them and the spell-mode recogniser could be identified and corrected. The speaker code used was the hidden representation of a compression network trained, as described in Section 5.6.2, to produce a complete set of formant pairs describing phones from a speaker, from a partial set of them. A classifier trained to use this speaker code to provide speaker identity information

had performance intermediate between a speaker independent recogniser and the recogniser given speaker identity information. The performance was also dependent on the number of phones from the speaker were used to form the speaker code.

Since speaker codes were clearly an imperfect, though useful, provider of speaker identity information, several experiments, described in Section 5.6.3, were done to find out how much information they would provide. In the first of these, speaker codes were generated by a pattern completion network that recirculated output estimates of formant values to fill in missing inputs. A separate network was trained to classify speakers when given this speaker code as input. This network was able to correctly identify thirty-five percent of the speakers, confirming that the speaker code contained a substantial proportion of the speaker identity information, independent of what subset of phones was used to produce it. Further experiments in this section showed that the network using speaker ID to aid recognition was making use of more information than speaker age and sex. This was demonstrated by the examining the way the hidden units that compressed the speaker identity clustered speakers, and by training a classifier to use a man/woman/child input to improve recognition. These latter inputs, though, improved performance almost as much as the voice codes had. When voice codes were derived directly from the F1,F2 values in the speech, by providing these values in place of speaker ID during training and testing of the recogniser, classification accuracy was only a little worse than it was with speaker ID. Since the task independent voice codes had a similar form to the task dependent voice codes and the hidden unit representations derived from speaker ID, but the latter representations improved recognition more, it was concluded that forming speaker codes within the target task was a more productive path to take, and that useful one-size-fits-all speaker spaces would prove difficult to produce and apply than had been anticipated.

After it had been established that the formant-based vowel recogniser was consistently and strongly helped by speaker information, whether that information was provided in the form of a speaker ID or information derived from previous speech, the focus returned to the spell-mode task where speaker information had been ineffective. The first of the experiments described in Section 5.7 matched the formant recogniser by restricting attention to the vowels, and by using speaker ID inputs. This recogniser, which used three full frames of speech as its input, and which may have been affected by the presence of superfluous speaker ID and phone target units, repeated the disappointing results of the first experiment with spell-mode recognition: there was no performance gain from speaker ID compared to a control, speaker independent recogniser. Repeating the experiment with only a single input frame worsened overall performance, but did not demonstrate an effect of speaker ID information.

At this point, it had been established that it was not the use of an ordinary backpropagation network rather than a second order recogniser that was preventing the use of speaker identity information. It had also been established that, for the format based recogniser, strong effects of speaker ID were visible whether the speaker information was presented as an ID, as a speaker code, or as a selection of other phones from the same speaker. The remaining possibility was that there were differences in the characteristics of the databases, or in the way they were presented to the networks, that accounted for the differences in performance. Experiments designed to explore these possibilities are described in detail in Section 5.8.

The most glaring difference between the two databases was that the Peterson and Barney data was presented as pairs of formants, explicitly locating the spectral peaks in the vowels,

whereas the spell-mode data was presented as frames of melscale filterbank coefficients. An experiment was done in which the formant values were replaced with synthetic "spectrograms" derived from them. These inputs could be classified almost as well as the original formants, excluding input representation as a salient difference. This representation did, however, render the second-order networks unable to use speaker information, confirming the virtue in simplicity.

The other most visible difference between the databases was that there was simply a great deal more information contained in the spell-mode input frames than in the two formant values. It seemed possible that this additional information rendered speaker information less useful than it was when the formant values only were available. To explore this possibility, a series of experiments described in Section 5.8.2 explored the effect of speaker information on classifiers working with reduced-dimensionality versions of the spell-mode data. In general, for vowels, lowering the dimension of the input data increased the effect of speaker ID. In these later experiments, where the networks were constructed to more closely match their vowel classification task and training speaker set, there was still a very small effect of speaker information on classification accuracy on unmodified input frames. The most plausible explanation for the success of speaker adaptation in formant classification was that in these experiments, the classifier *had* to use speaker identity to reduce misclassifications, whereas, with more complete inputs, it could infer most of the same information from the data itself.

Sections 5.8.4 through to 5.8.6 describe the attempt to extend the result for vowels to the entire phoneme inventory. Interestingly enough, speaker ID had an effect on consonant recognition that was just as strong as for vowels, and more consistent across representation sizes. However, when a unified all-phone recogniser was constructed, the effect was greatly diminished. Analysis of confusion matrices suggested that the recogniser was using the speaker identity information to improve performance on vowels at the expense of consonants. This effect was ameliorated by partially separating the vowel, consonant and silence recognition functions of the network, as described in Section 5.8.6. At this stage, it had been shown that it was possible to produce an effect, although not a large one, of speaker identity on recognition of training set speech for a respectably large task. In Section 5.8.7, the ability of this effect to generalise to other speech from the same set of speakers was investigated. Although the effect was still present, it was diminished for the testing data.

Since, even in the case of the formant classifier, task independent speaker codes had substantially less effect on classification accuracy than speaker ID, it seemed unlikely at this point that they would have a useful effect in the spell-mode task. However, for the sake of completeness, Section 5.9 describes two attempts to use task independent speaker codes with this data. As predicted, neither code had a useful effect on recognition accuracy.

Since the following sections serve to describe these experiments in detail, readers may wish to use the foregoing outline to choose which experiments to read about in detail, or may wish simply to proceed directly to the overall conclusions in Section 5.10 on page 132.

5.3. Early Application to French Digit Recognition

The earliest of the experiments in using speaker information to adapt a working recogniser was done in collaboration with Patrick Haffner at CNET[1] [witbrock92]. In this experiment, speaker voice codes (SVCs) produced by a recirculating hierarchical phone compression network were provided as additional input to various layers of a multi-state time delay neural network (MS-TDNN, described in [haffner91]) that had been designed to recognize telephone quality French digits.

Training and testing data for the experiment consisted of connected sequences of French digits recorded over the French telephone network. Each speaker uttered, on average, nine of the ten French digits. There were 3 540 spoken digits in the training set and 3 335 digits, from different speakers, in the testing set.

The two dimensional SVCs for these experiments were generated off-line, using recirculating completion networks similar to those described in the previous chapter. In this case, however, since the vocabulary was very small, the units modelled by phone pronunciation codes (PPCs) were not whole phones. Instead, each spoken digit was broken into five states using an accurate HMM recogniser available at CNET, and the variation in each of these fifty states, together with silence, was modelled separately[2]. These models of variation in acoustical states were combined into a SVC by a five-layer bottleneck neural network using the recirculation scheme in which missing inputs are filled in using estimates from the output layer. These resulting SVCs were averaged over the entire utterance for each speaker before being presented to the MS-TDNN recogniser.

The SVC was made available as additional input to the MS-TDNN using two additional input units, fully connected to two additional hidden units, that were in turn connected to every unit in the MS-TDNN. While the information from the SVC was available to all units in the MS-TDNN during training, during testing it could be replaced, for a given layer, by its average value across speakers.

5.3.1. Performance results on digit recognition

Table 20 gives the performance of the MS-TDNN with the SVC available (✓) or not available (✗) to each of the three hidden layers (approximately corresponding to acoustic, state,

1. CNET, the Centre National d'Etudes des Télécommunications, where parts of these experiments were performed, is the French national telecommunications research centre.
2. Since modelling variation in states, like the nonlinear time warp described previously, reduces one form of possibly extraneous variation, it would be desirable if it could be done with other databases. This was not possible because an accurate division into acoustic states was not available.

and word level processing) of the MS-TDNN recogniser. The estimated standard deviation

Acoustic	State	Word	%Error	s.d.
✗	✗	✗	1.77	0.23
✗	✗	✓	1.62	0.22
✗	✓	✗	1.74	0.23
✗	✓	✓	1.47	0.21
✓	✗	✗	1.17	0.19
✓	✗	✓	1.14	0.18
✓	✓	✗	1.11	0.18
✓	✓	✓	0.99	0.17

Table 20: Error rates given are the percentage of digit misrecognitions on the test set. The marks in the column to the left specify which of the three levels of the MS-TDNN the SVC was made available to during testing. When the SCV was not made available, to a layer, it was replaced with the overall mean value of the SVC across all speakers.

for each of the measured error rates is also given. This figure was calculated by assuming that the digits were identified as correct or not according to a binomial distribution. The probability of correct identification (p) is equal to the complement of the given probability of error (q). Under this assumption, the standard deviation of each figure is estimated by $\sqrt{npq}$, where n is the total number of digits in the test set. As a percentage error, this is written as $\frac{100}{n}\sqrt{npq}$.

There is a general trend evident in the data that the more places in the network the speaker code was made available, the better was the recogniser performance. The most substantial improvement occurred when the Speaker Voice Code biased the 1st hidden layer, which is chiefly responsible for identification of acoustic features. At this level, the most straightforward use the MS-TDNN could make of the SVC would be to effect an acoustic normalisation, using the speaker information to separate out the acoustic variability due to speaker differences from that relevant to the recognition task. The second layer of an MS-TDNN combines acoustic features into state scores, and the speaker model could influence the relative importance of these features. The third layer combines state scores into word scores, so any influence the SVC had here represents, approximately, a transition penalty for a state.

This experiment suggested that the long-term information about a speaker's voice encoded in the SVC was, as intended, relevant to the recognition task, and that it was relevant to several components of that task. Most importantly, the speaker code was able to influence the acoustic level of the MS-TDNN, enabling it to differentiate some of the acoustic variability due to speaker differences from the other sources of variation with which that variability is usually confounded.

5.3.2. Speed of SVC formation

While the pilot experiment described above indicated that the SVC could supply useful information to this recogniser, it did not test for all the qualities the SVC had been designed to have. It was still necessary to verify that the SVC could be formed from a small amount of speech, and that it could be formed from speech other than that it would be used to recognise and still be useful. In the previous experiment, the SVCs were formed from the entire utterance to be recognised, which was a somewhat unrealistic test, since the SVC formation itself depended on the speech being used having been labelled already.

To verify that the SVC could be formed from a subset of speech smaller than the whole target utterance, and that it could be formed from different speech than that which it was used to recognise, the experiment was repeated. This time, only the last four digits (of the nine total) from each of the 383 testing speakers were recognised. The SVC was either applied to all layers of the recogniser, which had been trained using SVCs derived, as before, from entire training set utterances, or not at all. The SVCs used in testing were also derived from only four digits, either, as in the previous experiment, derived from the four digits to be recognised, or from the first four digits spoken by the same speaker.

The difference in recognition error rates, with SVC available, between Tables 20 and 21 seem to suggest that the final four digits from each speaker were easier to recognise than the first four, although this difference was not significant[3]. This difference makes comparison a little difficult, but it is clear in Table 21 that the SVCs derived from only four digits were not

Source of SVC	% Errors	s.d.
No speaker voice code	1.50	0.31
First 4 (different) digits	0.85	0.23
Last 4 (same) digits	0.78	0.22
All nine digits	0.99	0.17

Table 21: Testing set performance of the system for the last 4 digits from each speaker, with the SVC derived from either the four digits being recognised, or from another four digits from the same speaker.

a significantly less useful source of speaker information as those formed from nine. There was also little difference between recognition scores for the final four digits from a speaker whether recognition was done with the aid of SVCs derived from the digits being recognised or from a different set of four digits from the same speaker. To the extent that the SVCs were able to affect recognition accuracy, they satisfied the goal that SVCs derived from a sample of speech should predict the sound of unheard speech from the same speaker.

3. The difference is less than one standard deviation, and therefore cannot possibly be significant. When further use of this argument is made, it will not be explicitly stated.

5.3.3. Discussion

For a network trained to use them, the SVC input resulted in a 43% decrease in recognition errors on new speech. Although this seems impressive, these recognition results must be viewed in comparison with the 1.1% error rate achieved by the best similar MS-TDNN system trained with ordinary non-speaker-dependent biases. Although this latter performance was not as good as that of the speaker-biased recogniser, it was better than that of the biased recogniser deprived of the SVC. The good performance of the speaker independent recogniser could be due to the fact that the ordinary MS-TDNN still had available to it a considerable span (>100ms) of speech context from which it could derive an approximate speaker model. Since, in the current system, the MS-TDNN had speaker information provided to it by the SVC, its performance suffered when it was deprived of this speaker information. When the speaker code was available to the whole system, as it was during training, the recogniser did somewhat better than the ordinary MS-TDNN.

At the early point when this experiment was done, the practice of using speaker ID inputs, via a bottleneck, as an idealised speaker model, had not yet been adopted. This practice is useful, since it provides information both about the best-case gains from speaker information obtainable from a given recogniser, and about the relative level of performance achieved by an SVC system in extracting this speaker information. Unfortunately, the programs and data necessary to go back and do that experiment were proprietary to CNET and are no longer available to the author. If they were, and the experiment could be done, one would expect, based on the outcomes of other experiments, that the performance would be better than, but not qualitatively different from, that reported above for speaker models derived from data.

These preliminary experiments seemed to indicate that the voice code was useful in tuning a recogniser to a new speaker, but that the improvement was not substantial, especially when compared with the performance of a recogniser with no speaker information whatsoever available. Although it was fairly clear, even in this early experiment, that, for the most part, the speaker model was replacing information that could have been extracted from the raw input, the consequences of this observation for adaptation based on models related to speaker identity were not yet fully apparent. The next step in the experimental strategy was to try applying the techniques to a larger database, with more speech from each speaker, in the hope that more substantial gains in recognition accuracy could be realised.

5.4. Scaling up to Resource Management Spell Mode Data

Following the moderate success of the experiment applying the speaker model to the French Digits task, it seemed appropriate to measure the performance of variants of the model applied to the larger Resource Management Spell Mode (RMSpell) database. In a pilot experiment for this new database and in subsequent experiments, the practice was adopted of using 1-from-n[4] representations of speaker identity as a sort of idealised speaker model giving perfect information about speaker identity. Performance with this ideal speaker code

4. A 1-from-n speaker representation is one where each speaker is represented by an input unit, and a distinct input unit activation value is used to distinguish the current speaker.

would serve as a basis for comparison with the performance of various speaker models derived from actual speech. For the pilot experiment using the new RMSpell database, recognisers[5] were trained with speech from three sentences from each of 20 speakers, both without and with 1-from-*n* speaker identity input. The task was frame-by-frame phoneme classification, for the entire phoneme inventory (vowels and consonants) of the database.

The hope that a larger, more realistic database would more clearly demonstrate effective adaptation by speaker biasing was not borne out. Training performance for the recogniser with speaker identity available was indistinguishable from that of the speaker independent recogniser, at 81%[6] correct frame classification. The prospects were dimming for successful adaptation by using speaker information as extra input to a quasi-speaker-independent recogniser, as opposed to retraining the system[7] for new speakers, or modelling each speaker separately.

Herman Hild [hild93], who was at the time a visitor in the Neural Net speech group at CMU, had done some related experiments with multi-speaker recognition when developing his high-accuracy recogniser for the RMSpell Database. In these experiments he had applied a variety of speaker adaptation techniques to a high performance neural net based recogniser for spelled letters. He used either a speaker identification network, similar in principle to the speaker models investigated in this thesis, or a tuning-in procedure like that described by Cox and Bridle [cox90,bridle91], to produce speaker specific additional inputs. These inputs were used to bias a speech recognition network, or to combine the outputs of either complete networks, or speaker specific layers in a larger network, using multiplicative connections. In experiments with few speakers, using the six speaker CMU Alph database, the methods involving combining speaker specific networks or subnetworks were successful. However, biasing a single network with speaker identity information was only useful when the speaker was in the training set, and explicitly identified. Biases produced by the speaker identifying network were not significantly helpful, and neither were biases identifying speaker gender.

On the larger RMSpell database, with seventy-four male speakers clustered into six speaker groups, even tuning in was not helpful. For this case, with many speakers used, the only adaptation that was effective was supervised tuning-in of a cluster mixture separately for each phoneme. Since this required phoneme labels for the tuning-in speech, and a relatively large amount of speech (five spelled words) for each new speaker, and, in any case, required tuning in, rather than model identification, the modest success achieved does not satisfy the criteria for successful speaker adaptation that were adopted for this thesis. This is in no way a criticism of [hild93], the work reported in which is both valuable and interesting, particularly since it points out the same sort of difficulty in adaptation reported in the present work.

---

5. The recogniser used was an experimental neural net/Viterbi recogniser that used separate nets for each phone state, and that could relax the first order Markov assumption by using the recent best alignment path to decide what input to use in recognising the next state. Work on this recogniser was suspended to allow concentration on speaker modelling, and this experiment is the only one using it that will be reported in this thesis.

6. To within the +- 1% average change in accuracy between epochs.

7. That is, reestimating parameters using a training set.

5.5. The utility of speaker Information

Although the initial application of the speaker model to a real-world speech recognition task had shown some promise, the improvements had been less than spectacular, and review of a similar experiment [hild93], described above, had suggested that this problem was not confined to the current work alone. Since one of the assumptions driving the work had been that substantial improvements in recognition accuracy could be obtained by giving a recogniser speaker-specific information, it was important, if speaker adaptation was to continue to be used as a test bed for speaker modelling, to establish whether the problem lay in the speaker information itself, or in the recogniser's ability to use the information.

One of the major motivations for the belief that speaker information would help substantially had been Watrous's [watrous93] paper on speaker adaptation using second order connectionist networks. The experiments reported in that paper used phoneme recognition on a multi-speaker vowel database as a model for the speech recognition task. Replicating these experiments, using the SVC speaker modelling method for specifying speakers, presented itself as an excellent way to test whether the speaker model was limiting performance on the digits task, or whether the expectations engendered by Watrous's vowel discrimination work were not justified for other reasons.

5.6. Peterson and Barney Database Experiments

The Petersen and Barney (PB) vowel data [peterson52,watrous91] is a database consisting of formant values for two repetitions each of ten vowels spoken by seventy-six speakers (thirty-three men, twenty-eight women and fifteen children). Because it contains a substantial amount of speaker based variability within a compact database, and because it has been used to test other adaptation schemes, it was appropriate to use it as a vehicle for investigating why the speaker adaptation techniques applied to larger databases in the previously described experiments had shown, at best, limited success.

The first stage of this investigation was a replication of the simple adaptation scheme that had failed for the RMSpell data. Was it possible to use speaker identity explicitly as additional input to a network and obtain an improvement in vowel discrimination?

5.6.1. Speaker ID Biases

To provide a baseline for experiments using the model of speaker variation as an adaptation source, initial experiments were done testing classification with **a**) no speaker dependent information and **b**) complete knowledge of speaker identity. Speaker identity was made available to the net either as a speaker dependent bias, or, following Watrous's practice, as a modulatory input via second order units. The four conditions tested are shown in Figure 29.

In the first, *Control,* condition, the net was trained to output vowel labels given the first two formants, normalised to lie between 0 and 1, as input. In the *Bias* and *Second Order* conditions, the network was told which of the seventy-six speakers the formant values were from, using a 1 from *n* representation via a two unit bottleneck. In the *Bias* condition, the output of these units fed normally into the 'first' hidden layer, while in the *Second Order*

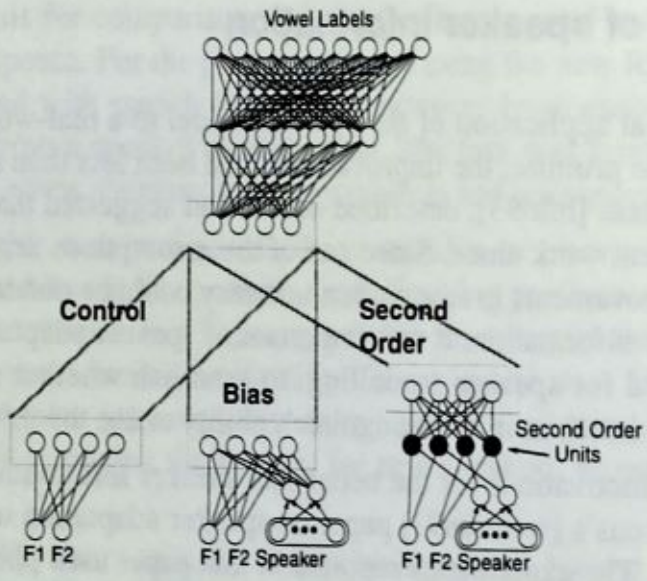


Figure 29: Architectures used for the Peterson-Barney experiments. Linked grey boxes surround corresponding units. The Bias Direct condition was the same as the Bias condition, except that like the formant inputs, the seventy-six speaker ID inputs were connected directly to the four hidden units.

condition, second order units[8] were used to form a linear combination between the two "compressed speaker ID" unit activities and the formant values. The second order units were connected conventionally to the hidden layer. The *Bias Direct* condition omitted the bottleneck between the speaker ID input units and the hidden layer.

The nets were trained using the backpropagation algorithm with momentum. All networks had two input units (first and second formants), four units in the first and seven in the second hidden layer, and ten outputs. In each case, training was done for 6 000 epochs. Following the practice used in [watrous93], asymptotic training performance was measured; reported results are the average of training set classification accuracy after epochs 5 600, 5 800 and 6 000.

The classification performances of the networks are displayed in Table 22. It is clear from

Network	% Correct
Control	77.98%
Bias	95.83%
Bias Direct	98.40%
Second Order	92.52%

Table 22: Results for Various Architectures for Speaker Adaptation applied to the Peterson and Barney database. The networks are described in the text.

this table that substantial performance gains were available from speaker adaptation on this

8. Although four second order units are shown, in fact nine were used, since two constant bias units are also necessary.

database, and that these gains were available even when speaker information was provided as simple additional input to the net. Architectural differences between the networks that were used in the previously described experiments and those used in Watrous's work, then, do not account for the unexpected failure of speaker information to affect recognition performance on the RMSpell databases. In fact, Watrous's suspicion [watrous93] that "in the limit, the approaches [normalisation using second order nets, and direct modulation of the classifier] may be equivalent" was more than confirmed. The simplest means of furnishing speaker ID information to the PB vowel classifier, furnishing the ID as extra input, produced the largest performance boost, slightly exceeding the best classification performance obtained in the original [watrous93] paper.

Specialised Networks for Phone Classification - a brief digression

Because earlier pilot experiments, not reported here, had suggested that adaptation information was more salient, and therefore more useful, to networks specialised for particular phones, an comparison was made between this condition and the default of a single classification network. The experiment reported above was actually done in each condition both with the usual single net, trained to classify among the ten phones, and with ten distinct, one output nets, trained to do the classification task collectively.

Unless otherwise noted, training was done for 6 000 epochs, and reported performance is the average of tests done after epochs 5 600, 5 800 and 6 000. All other training conditions were the same as specified above.

Speaker Info	Single (10 output) Net.	Multiple 1 output Nets
Control	77.98%	78.77%
Bias	95.83%	97.55%[a]
Bias Direct	98.40	97.63
Second Order	92.52%	88.95%

Table 23: Baseline figures for effects speaker bias in Petersen/Barney task. Four methods of providing the speaker information to the networks were used, for both a single recogniser, and a set of cooperating phoneme-specific recognisers.

a. Average of epoch 3 800, 4 000, 4 200. Training was stopped early

Although the recognition performance for the multiple networks was slightly higher in the *Control* and *Bias* (via 2 hidden unit) conditions, it was lower in the other two cases. Dividing the network into phone identification specialists did not produce substantially different results from the usual unified classifier, and was not continued. A less fine-grained variant of this division was, however, used in some later experiments that will be described towards the end of this chapter.

5.6.2. Using Speaker Models to Produce Biases

As an test of speaker model formation for this database, networks were trained to produce an F1, F2 pair for each phone for a speaker, given a different random sample of phones from the same speaker. Which of the two repetitions of each phone was used as the target was chosen at random for each pattern. Zeros were placed on the inputs for unselected input phones. The object of the exercise was to produce a network that would implement a model of voice variation capable of inferring the sound of target phones from an incomplete sample of phones from the same speaker. In doing so, the network would form an SVC for the speaker in its "hidden" units. Since variation in the target patterns for a speaker was independent the choice of which of the speaker's phones would appear in the input pattern, this system would be expected to produce a model of the speaker's voice which was largely independent of which subset of phones had been heard, and which was robust in the face of unheard phones.

The compression networks used in model formation each had twenty input and output units, and three hidden layers of five, two and ten units respectively. The two unit layer was the source of the SVC, and the five and ten unit layers served respectively to encode and decode this SVC. Training was done using a learning rate of 0.0001 and momentum of 0.8. Training was done for 15 000 epochs in three blocks of 5 000 epochs each. In the first block, weights were updated after each pattern presentation. In the second, they were updated after every thirty-nine patterns, and in the third after an entire epoch of 1 520 patterns had been presented.

Models were formed in three conditions. The "bias" case was strictly feed-forward. In the "bias recirc" case, phones missing from the input pattern were replaced with their estimates from the output during production of SVCs. The recirculation process was repeated five times before the SVC was extracted for use in adaptation. In the last condition, the recirculation was done during training as well as testing, with error gradients accumulated on each cycle. For this condition, a lower learning rate of 0.00001 was used. For each of the three conditions, networks were trained for every case in the range from having one phone placed on the input to having examples of all ten phones from the speaker on the input. Thirty SVC-producing networks were generated altogether.

To evaluate performance, a set of twenty SVCs from each network was produced for each speaker. These SVCs were used as additional inputs to conventional four layer classification networks with four units on the input, ten output units, and four and seven units in the first and second hidden layers, respectively. The networks were trained for 6 000 epochs, in three stages of 2 000 epochs each. In the first stage, weights were updated after each pattern, in the second, every thirty-nine patterns, and in the final stage, every complete epoch of 1 520 patterns. The learning rate used was 0.001 and the momentum was 0.9.

Results

Table 24 gives the classification performance on the training set of each of the thirty networks after the completion of training.

SVC used	Number of phones presented in each input pattern									
	1	2	3	4	5	6	7	8	9	10
Bias	85.3	87.8	87.9	89.0	89.3	88.8	89.2	89.6	89.9	89.6
Bias recirc	83.0	84.7	84.5	85.9	86.8	88.2	88.8	88.7	90.3	89.6
Bias recirc (train + test)	85.1	87.8	88.4	87.2	87.7	88.0	88.4	88.9	89.1	89.3

Table 24: Training set classification performance for Peterson and Barney Data. A description of the three conditions is given in the text. The figures in the cells are percentages of correct phone classifications from the input formant pairs.

Overall, these results were clearly better than the completely speaker independent classifier, which provided a baseline phoneme classification accuracy of 78%, but, as might be expected, did not reach the level of performance (98.4%) of a network trained to tune its performance to completely specified speakers in the "bias direct" condition of the experiment using speaker ID.

No particular benefit seems to have been gained from the technique of recirculating estimated phones back onto the input during speaker code production, in fact, the performance was slightly better, on average, for the unrecirculated speaker model. The models did, however, improve somewhat with increasing numbers of phones used to build the SVCs, going from an average of 84.5% correct when one phone was used up to an average of 89.5% correct when all ten phones contributed information.

Discussion

Speaker codes formed from formant pairs representing phones from the same speaker were an effective source of speaker information for speaker adaptation, providing more than 50% of the error reduction available using speaker ID. When used with the simplified speech data represented by the Peterson and Barney data-set, the hope that a useful space of speakers could be formed seemed to have been justified. Unfortunately, there were insufficiently many speakers represented in the data-set to test whether the networks would generalise, improving classification performance for speakers outside the set used in training.

Since speaker adaptation without retraining had been shown to be effective for the Peterson and Barney data, whether the speaker information was presented as speaker IDs or as a speaker code, but had been ineffective when applied to the RMSpell database, an attempt to discover the salient differences between the two databases was warranted.

Before proceeding to a description of that attempt, several more experiments that were done to explore the characteristics of the speaker models built from the Peterson and Barney data will be described.

5.6.3. Exploring Speaker Codes for Peterson and Barney

Did Speaker Models do Speaker ID?

Since the highest (training set) performance was reached when the net was able, using Speaker IDs, to distinguish perfectly between speakers, it seemed worthwhile to find out how well the adaptation codes resulting from neural net compression served to support this distinction.

The speaker codes used for this experiment were ones generated in the bottleneck layer of a five layer net, with twenty inputs, five units in the encoding layer, two units in the speaker code layer, ten units in the decoding layer, and twenty output units. The compression network was trained with a learning rate of 0.001 and momentum of 0.8 following a rather complicated training schedule involving varying update frequency, number of phones presented to the input and the target, and the number of times outputs were recirculated back to missing inputs[9]. No error was backpropagated from target units for which training values were not available. Speaker codes were generated after training by running the network in feed-forward mode with randomly chosen phones from the speaker placed on the input so that on average, 60% of the input units were used. Twenty speaker codes were generated this way for each speaker.

To test the usefulness of these speaker codes as a means for speaker identification, another net was trained to identify the speaker, given the (gender independent) speaker voice code. This was not, apparently, a simple task, since a number of attempts were required before a successful network architecture was found. A four layer network with bypass connections was trained to activate an output unit corresponding to one of the 76 input speakers when a speaker code consisting of two floating point numbers was placed on the input.[10] After 50 000 training epochs, it had reached a performance of 35.3% correct speaker identification over the 1 520 input patterns, suggesting that the speaker voice codes the network had formed were at least somewhat effective at capturing the speaker's vocal characteristics, independent of the subset of vowels used to form them.

Did the Speaker Models model Sex and Age?

Since using the speaker ID as a bias gave the highest recognition performance, 95.8%, of any of the networks employing a bottleneck, it was of interest to examine the representations formed in the hidden bias units. These were plotted, labelled for the three main speaker groups (man, woman, child), in Figure 30. While these groups clustered, they were not disjoint, suggesting that these groups do not represent the best possible division of speakers

9. The aim behind this schedule, for what it's worth, was to attempt to shape pattern completion behaviour in the network. Details of the schedule are found in Appendix F.

10. The network had two inputs, seventy six outputs, and ten units in each of two hidden layers. The learning rate used was 0.01 and the momentum was 0.9.

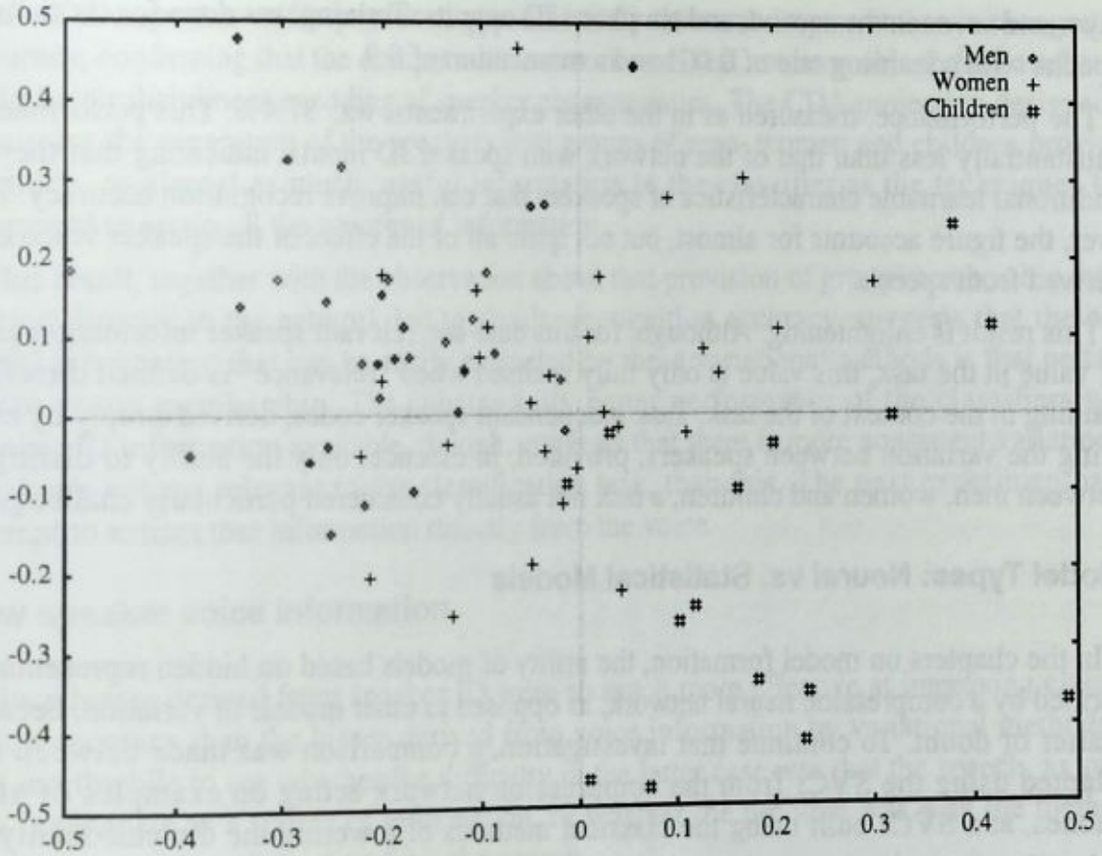


Figure 30: The outputs of the two hidden units through which speaker ID biases were presented to the "Bias" network, labelled by speaker class. Although much of the variation in the learned speaker representation was due to speaker class, the variation within the classes, and overlap between the classes, suggests that the classifier could make useful distinctions between speakers that extended beyond class alone.

with respect to vowel separability. Moreover, the groups did not map only onto a single position in bias space, indicating that there was information beyond group membership which was used to separate the vowels.

Speaker Models Compared to Sex and Age Information

It was clear that most of the variation in the code produced from speaker IDs was due to gender and age. What was less clear was how useful the residual variation modelled by speaker ID was for recognition. To answer this question, an experiment similar to that in which speaker IDs were provided was done. Using three input units to tell the network whether the speaker was a woman, a man or a child allowed it to make maximal use of this group information, and the difference in performance between this network and that given full speaker ID would indicate what proportion of the speaker adaptation effect was due to variation other than simple group membership. Apart from the training set and number of input units, the network and training procedure was identical to that used for the speaker ID bias network: the three group IDs were added, via two hidden units, to the first hidden layer of a four layer discriminant network with two formant inputs, four units in the first hidden

layer and seven in the second, and ten phone ID outputs. Training was done for six thousand epochs with a learning rate of 0.001 and a momentum of 0.9.

The performance, measured as in the other experiments, was 87.4%. This performance is substantially less than that of the network with speaker ID inputs, indicating that there are additional learnable characteristics of speakers that can improve recognition accuracy. However, the figure accounts for almost, but not quite all of the effect of the speaker voice codes derived from speech.

This result is enlightening. Although, for this data-set, relevant speaker information can be of value in the task, this value is only fully realised when "relevance" is defined directly by training in the context of the task. Task independent speaker codes, derived simply by examining the variation between speakers, provided, in essence, only the ability to distinguish between men, women and children, a task not usually considered particularly challenging.

Model Types: Neural vs. Statistical Models

In the chapters on model formation, the utility of models based on hidden representations formed by a compression neural network, as opposed to other models of variation, became a matter of doubt. To continue that investigation, a comparison was made between a net adapted using the SVCs from the compression network acting on examples of all ten phones, and SVCs built using the classical methods of lowering the dimensionality of a data-set: principal components analysis, and canonical discriminant analysis using "male", "female" and "child" as the target group labels[11].

The forty element vectors representing all twenty phones from a speaker (two utterances of ten phones each) were projected onto the first two directions of maximum variation and group separation respectively for the data, and those values were supplied to networks in a manner identical to that used for the compression network based speaker voice codes.[12]

Speaker Info	Performance
No Speaker Info	77.98%
SVC Adapt Bias	89.6%
Principal Cpts	88.64
Canonical Disc	88.00%

Table 25: Comparison of Neural and Statistically derived Speaker Biases. All three biases had a similar ability to improve classification accuracy.

11. There aren't sufficient samples in the Petersen Barney data to allow a discriminant function for speakers to be learned.
12. This is slightly more information that the randomly selected single example of each phone used for the compression network. However, the compression network was improving little, if at all, between having nine and ten phones available, so it seems improbable that adding extra examples of the same phone to the inputs of the statistical models furnished them any great advantage.

All of the speaker codes produced approximately the same improvement in recognition accuracy, confirming that the compression network used had not been able to find a substantially better than linear encoding of speaker characteristics. The CDA projection designed to maximise the separation of the speakers into groups of men, women and children provided as much, or almost as much, useful information to the classifier as the techniques that attempted to retain all the sources of information.

This result, together with the observation above that provision of group membership information directly to the network led to similar recognition accuracy, suggests that the only useful information that can be easily extracted by these variational methods is that pertaining to group membership. The substantially better performance of the classifiers with speaker *ID* information available, though, suggests that there is more consistent variation in speaker's voices, relevant to the classification task, than this. The next experiment is an attempt to extract that information directly from the voice.

Raw speaker voice information

Since biases derived from speaker ID were so much more effective at improving classification accuracy than the biases derived from voice information by variational methods, it was worthwhile to see whether the difficulty in the latter case was that the speech, as such, was inadequate as a source of information, or whether the problem was with the methods used to extract the information from the speech.

To this end, a net was trained with the objective of producing the values that had appeared as "bias unit"[13] activities, in the network using speaker ID inputs, directly from voice information. The three layer network used had twenty input units — two for each phone, ten hidden units, and two output units. Training was performed on a rather complicated schedule detailed in Appendix G. The biases estimated by the outputs of this network were used to provide adaptation input to a phone classifier of exactly the same type used in the previous experiments. As a control, a second phone recogniser was trained with the previously learned biases, from the network adapted with speaker ID, as adaptation input. This provided a direct comparison with the estimation of these biases from voice data. It should be noted that using the Speaker ID based biases was not quite the same using the speaker ID inputs themselves, since the biases now passed through another, two unit, hidden layer. Phone classification accuracy using the actual biases taken from the speaker ID network was 94.26%, and using the estimates of those biases based on voice data was 89.80%.

The classifier given biases derived from accurate information about speaker identity performed only slightly more poorly than the classifier using the speaker ID information directly. However, when an attempt was made to derive this same information from the voice directly, the performance of the classifier was similar to that of networks using codes derived using the variational methods.

While it might be tempting, at this point, to conclude that it is not the use of variational methods that is problematic, and that the speech information itself is not reliable enough to support the production of good speaker codes, this conclusion is not warranted, as the next experiment demonstrates.

13. The two hidden units through which the speaker IDs were furnished to the classifier.

A classifier using voice information directly for adaptation

The experiments using raw speech data for adaptation, up to this point, had involved using that data to construct a predetermined model, and then applying the speaker codes derived from that model to the adaptation task. In one case, the model involved was one of the variants on PCA or LDA, neural or otherwise, and in the other case, the model was one previously derived from adaptation with speaker ID. What remained to be seen was whether a classifier free to use this speech data in any way whatsoever could learn to derive a code from it that was effective for adapting the classifier, at least in the context of this greatly distilled speech data-set. To this end, a pair of phone classifiers were constructed to use the raw formant values for phones from a speaker as the "speaker code".

All forty F1,F2 pairs for each speaker were made available to a classification network as "adaptation information" by one of two means. In one network, they were placed on forty extra input units that were connected directly to the hidden layers; in the other they were connected via two extra hidden units in the usual way. Apart from this difference, the two networks were the same, having forty two units in the input layer, ten output units, and four and seven units, respectively, in the first and second of two hidden layers.

Training was carried out in thirty stages of two hundred epochs each, with 1 520 pattern presentations per epoch. The learning rate was 0.001 and the momentum was 0.9. For the first ten training stages, the weights were updated after every pattern, for the next ten, after every thirty nine patterns, and for the final ten, once per 1 520 pattern epoch.

The phone recognition accuracy for these two networks is given in Table 26. In both cases, the raw speech information proved useful to the classifier, whose performance approached,

Bias Presentation	Performance
All F1F2 via 2 hidden units	93.75%
All F1F2 direct	94.12%

Table 26: Phone classification performance for networks allowed to derive speaker information from all available phonemes. Formant values for all the phones from each speaker were made available to the networks during classification of a particular formant pair. The network was able to use this information to improve classification, whether the twenty formant pairs were made available directly or via a two unit bottleneck.

but did not match, that of the networks with explicit speaker ID input.

5.6.4. General observations from these experiments

Adaptation information derived from the variational techniques: SVC from networks completing partial phone information, PCA over all phone information for a speaker, and CDA separating speaker classes, was approximately equally useful; at most slightly better than knowledge of gender and age. Deriving a "speaker identity" code from speech information

provided a similar level of performance. None of these techniques matched the performance of networks given accurate speaker identity as "adaptation" input, or allowed to use all available acoustic information from the speaker in a similar manner.

The essential difference between these two cases is that in the former case, the speaker representations that were formed were independent of the recognition task, but in the latter case they were formed in the course of doing the recognition task. This observation does not rule out the possibility of forming stable speaker representations that can be used to improve recognition, but it does suggest that the mechanism that does so will have to be trained in the context of the task in which the representation will be used. The ability of networks applied to this small task to extract adaptation information from speech is a promising sign that this can be done. Of course, to generate improvements in real-world recognition tasks, the speaker information that is extracted in this way must exceed that previously available to the recogniser. The difficulties of ensuring this for data less idealised than the Peterson Barney set will be demonstrated in the remainder of this chapter.

Despite the disappointing performance of the task independent models, models based on voice data, when developed in the context of the recognition task, had shown some promise. The next step was to attempt to replicate this success on a more realistic data-set: a subset of the RM Spell data.

5.7. Using speaker information with the RM Spell Database

After the Peterson Barney Experiments, which showed a strong effect for known speaker, an attempt was made to replicate those experiments using the less processed speech from the RM Spell database. The first, pilot, experiment used vowels extracted from three sentences spoken by each of twenty speakers, five women and fifteen men, randomly selected from the database.

5.7.1. Initial Experiments with Speaker ID using the RM Spell Database.

A four layer backpropagation network was trained to do vowel recognition, using input consisting of three frames of melscale FFT analysed speech from the RM Spell mode database. The speech was presented on forty-eight input units, and there were eight units in the first and fourteen in the second of two hidden layers. For the speaker dependent condition, speaker ID inputs were fed, via a two unit bottleneck, into the first hidden layer. Although there were only twenty speakers in the subset of the database used for this experiment, ninety-six speaker ID inputs, one for each of the speakers in the full database, were used. The network was trained to activate a single phone output unit, corresponding to one out of the twenty-seven phonemes occurring in the RMSpell data. In this case, since the experiment was confined to vowel recognition, only nine of these twenty-seven phones were present in the training data.

Training was done in three stages of four thousand epochs, with intervals between weight updates of one, thirty-nine and 1 520 pattern presentations respectively. The learning rate was 0.001 and the momentum was 0.9.

Table 27 summarises the performance of the three networks that were trained. The recognition accuracies given are means calculated over the accuracy at the end of each of the last three sets of two hundred training epochs.

Network Architecture	Percent inputs labelled correctly	
	Random pattern order[a]	Fixed pattern order
Control (no speaker ID)	84.1	82.8
Speaker ID as Bias	83.8	82.8
Speaker ID via Second Order Units	81.4	81.7

Table 27: Comparison of performance of a control network, with no speaker information, and nets with two kinds of speaker ID bias. The task was vowel classification for a small subset of the RMSpell database. Three frames of speech were available to the classifier as input. In this case, the availability of speaker ID did not improve classification accuracy.

a. Random vs. Fixed order of presentation conditions represent an experiment to see whether the order of pattern presentation affected the results. Randomizing the order seemed to help slightly, but the distinction is not important. Where both ordering regimes were tried in further experiments, both figures will be reported without further comment.

Disappointingly in the light of the experiments using the Peterson and Barney database, there was no performance boost at all from the speaker dependent bias, and second order modulation of the input patterns using speaker identity actually worsened performance.

5.7.2. Some Simplified Experiments

In the hope that a somewhat simplified experimental setup would demonstrate an effect of speaker ID for real data, and that this would aid in the diagnosis of the cause of the failure of the network in the previous experiment to make any use speaker ID, the experiment was repeated with a smaller input window. In this case, the network was presented with a single frame of input via sixteen input units. It had four units in the first and seven in the second hidden layer, and output units for twenty seven phones, although only nine were used.

Training with a single frame of Input.

Training was done in three stages of four thousand epochs each, during which weights were updated every one, thirty nine, and 1 520 pattern presentations respectively. As in the previous experiment, for the speaker dependent case, speaker ID biases were provided through ninety six speaker ID inputs, of which only twenty were used. These biases passed through a layer of two hidden units, before reaching the recogniser. Training parameters were identical to those in the previous experiment.

The phoneme recognition accuracy, measured as mean classification performance for the last three batches of two hundred epochs, is given in Table 27.

Network Architecture	Percent inputs labelled correctly	
	Random pattern order	Fixed pattern order
Simple	73.7	73.1
Bias	73.4	73.4
Second Order	72.6	72.8

Table 28: Comparison of performance of a simple classification network, and nets with two kinds of speaker ID bias, for a small subset of the RMSpell database. In this case, only one frame of speech was used as input. This lowered performance compared to the three frame case, but did not render biases effective.

Discussion

The aim of this experiment using a restricted amount of speech input was to make the classification task more difficult, and thereby to improve the likelihood that speaker information would be useful for reducing ambiguity. The classification performance was significantly lower than for the classifier with three frames of input, indicating success in making the classification task more difficult.

Despite the increased difficulty, the speaker-id-based biases did not produce any improvement in recognition accuracy. The network was not able to use the biases to modulate its classification regions in any useful way, whether they were provided in the conventional manner or via second-order units.

Although the task independent speaker models had not been particularly effective at improving performance on the previous, Peterson and Barney, database, they had produced a measurable effect, and models constructed in the context of the task seemed a promising possibility. The failure of speaker identity, itself, to produce an effect when similar networks were applied to a recognition task involving less idealised speech, by contrast, suggested that the whole enterprise of speaker adaptation by network modulation, as opposed to retraining, might be doomed.

For this reason, considerable effort had to be focused on an attempt to identify the qualities of the Peterson and Barney task and the RM Spell task that might account for this difference.

5.8. How do Database Differences Affect Performance?

Even though the data, vowels from a smallish set of speakers, was essentially similar to that in the Peterson & Barney set, knowing who the speaker was did not help classifiers working on speech from the RM spell database. Since it had already been established that the network architecture was capable of supporting the kind of adaptation wanted, remaining possi-

ble sources for the differences between the effect of speaker information on classification of data from the two data-sets included:

- The nature of the input representation: The Peterson & Barney data was represented using formant frequencies extracted from steady vowels, whereas the Resource Management Spell data was presented as melscale spectra.
- The number of available parameters: The speech representation for the Resource Management data was much richer - there were forty-eight or sixteen input parameters for the experiments on that database, compared with only two for Peterson and Barney.

The following subsections describe efforts to investigate the effects that varying the data representation and the number of parameters used in the representation had on the ability of phone classifiers to use speaker information.

5.8.1. Input Representation - Frequencies vs. Filter banks

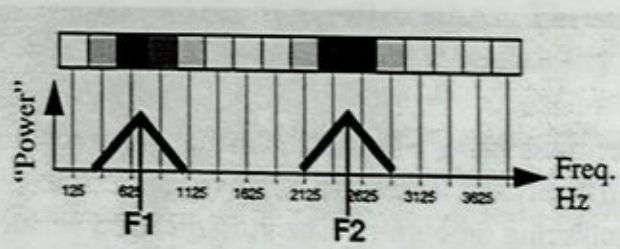


Figure 31: Synthetic spectrum formed by constructing triangular "power functions" with their maxima at the two formant frequencies given in the Peterson and Barney database.

The most obvious of the possible culprits for the lack of success with the Resource Management (RM) data was the change in input representation from formant values to filterbank outputs. To test whether formant frequencies were somehow more amenable to speaker adaptation,[14] synthetic spectra were generated from the Peterson & Barney data as shown in Figure 31. These sixteen-valued spectra replaced the two formant values on the inputs, and the recognition experiments were repeated in exactly the manner as before.

14. This is plausible, since a shift in fundamental frequency can be represented by addition in the formant representation, but only by a linear transform in the filter bank representation.

Recognition accuracies for networks trained with this kind of input are given in Table 29.

Network	Percent inputs labelled correctly
Simple	79.4
Bias	94.0
Second Order	79.1

Table 29: Recognition accuracy on Peterson and Barney formant pairs converted into a synthetic spectral representation. Ordinary biases inputs were nearly as effective with this represenation as with the raw formants, but the second-order biases were rendered ineffectial.

Although this representation seems to have prevented the second order networks from making any use at all of the speaker ID information,[15] it had very little effect on the normal "bias" networks. Clearly, the formant representation offers no great advantage over the more usual spectral representation, as far as speaker adaptation is concerned. The fact that a spectral representation was used for the RM database is unlikely to have been the reason speaker adaptation was ineffective in that case.

5.8.2. Number of Input Parameters - Reduced Input Representations

The Peterson & Barney data consisted of just two formant values per phone, an extremely parsimonious representation. It seemed possible that the RM speech data, with sixteen meaningful coefficients per frame, was able to support a partition of the input space into vowels sufficiently good to make speaker identity of marginal utility. In order to test this hypothesis by reproducing the training conditions of the Peterson Barney data as closely as possible, an experiment was done using input patterns produced by projecting RM data onto their first n canonical discriminants. The discriminant functions were built using vowel classes as the groups to be discriminated. Again, conditions with both one and three frames of input data were used.

The vowel discrimination networks used n input units, four units in the first and seven in the second hidden layer[16], and n output units (one for each of the vowels c.f. all twenty-seven for previous experiments). For the biased cases, twenty extra speaker ID inputs were presented to the network via two hidden units. The database was presented in both conventional[17] and randomised orders. Training was done in three stages of two thousand epochs, with update intervals of one, thirty-nine and 1 520 Epochs respectively.

15. It is interesting that this should be so. Although the second order units performed reasonably well when the task to be performed was more or less exactly a linear transform of the input (i.e. with formant inputs) they seem to be positively harmful instead of just neutral, with respect to other tasks. This reinforces the wisdom of using a simple model, like vanilla backpropagation, until one is *forced* to use a more complex one.

16. half the number of hidden units used in the preliminary experiment (§ 5.7).

17. That is, the order in which they appeared in the training set file., In particular, all input patterns from a single phone, and all phones from a single speaker were adjacent.

One Frame of input

The first condition was the projection of a single frame of speech onto lower dimensional representations identified using CDA. In addition to the fifteen reduced-dimension cases, a network (labelled as n=16) was trained using the raw input frames directly, for the purposes of comparison.

Table 30 gives classification accuracies for both the speaker-biased and unbiased networks for each dimension. The percentage error reduction produced from speaker ID is displayed

	Input Dimension													Raw (16)
Case[a]	1	2	3	4	5	6	7	8	9	10	11	12	13	
Bias	66.0	70.5	73.6	74.0	74.5	74.7	75.1	75.0	75.4	75.2	75.0	75.9	76.0	76.3
Simple	59.3	66	68.6	70.2	70.9	71.8	71.9	72.1	72.1	71.8	72.8	72.4	72.8	73.6
% Redn	16.5	13.2	15.9	12.8	12.4	10.3	11.4	10.4	11.8	12.1	8.1	12.7	11.8	10.2

Table 30: Improvement in recognition performance from speaker bias for RMSpell vowel data projected onto various dimensions of canonical discriminants. These figures are for networks trained with randomised training pattern order.

a. This table gives the result for randomised pattern presentation only.

in Figure 32, with number of canonical variants on the x-axis, and performance at the end of training on the y-axis, both for the randomised training set order of Table 30, and for the

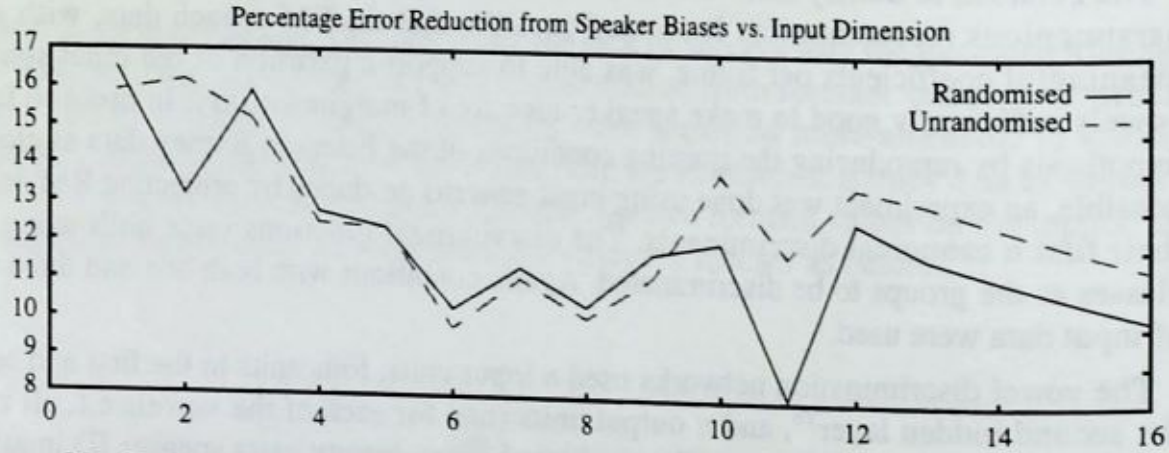


Figure 32: Reducing the dimensionality of the input representation appears to increase the effect that speaker ID information has on recognition accuracy.

original pattern order. The graph suggests two things:

- that there is some improvement in recognition accuracy[18] to be gained from speaker biases, for inputs one frame wide, and,
- that a possible source of the difference in the effect of speaker bias between the

18. Again, as in the Peterson and Barney data, on the training data.

RMSpell and the Peterson Barney data is the dimension of the data representation, since lower dimensional representations (in Figure 32) show stronger performance gains from speaker biases.

While the unbiased performance on raw frames (16 inputs) was almost identical to that in the preliminary experiment described in §5.7.2, in this case the biased network showed a modest improvement in recognition accuracy. However, despite the gain from speaker bias in this case, the best biased recognition performance was still lower than the 84.1% accuracy achieved in the preliminary experiments by a network with three frames of input speech available (see Table 27 in §5.7.1). This suggests that speaker information, even if it increases training set performance, may provide no more useful information than a recogniser could extract directly from a little more speech.

Three Frame Input Case

To get some measure of the extent to which the success of speaker modulation in the last experiment was simply due to the impoverished nature of input from a single frame, it was repeated using three frames of speech as input to the CDA projection. These inputs were generated by pasting together barrel-shifted versions of the FFT file used in the previous experiment. Table 30 shows the results of training for these networks which, with the exception of one network with forty-eight inputs, were identical to those used for the one-frame case.

	Input Dimension													Raw
Case[a]	1	2	3	4	5	6	7	8	9	10	11	12	13	48
Bias	66.6	71.4	75.3	75.3	76.5	75.9	76.6	76.7	76.5	77.1	77.7	77.7	77.6	79.3
Simple	59.8	66.6	70.5	71.6	72.7	73.3	73.5	73.8	74.4	73.9	74.6	74.4	73.9[b]	77.0
% Redn	16.7	14.2	16.1	12.7	14.2	9.7	11.8	11.1	8.1	12.4	12.0	12.8	14.0	9.7

Table 31: Improvement in vowel classification performance due to Speaker Bias for networks given three frames of RMSpell Vowel data projected onto canonical discriminants of various dimensions.

a. This table gives the result for randomised pattern presentation only.

b. Training for this case only continued for 1000 epochs, c.f. 5800 for the others.

As in the one frame of input case, the effect of using speaker biases was more pronounced at lower dimensions, when little data was available from the speech, and rather smaller (<10%) when all three frames of speech are presented to the input. Overall performance was similar to that of the network with one frame of input; the largest difference being was apparent for the raw input case for each network, where the availability of all forty-eight inputs allowed both the biased and unbiased networks to correctly classify three percent more of the inputs.

However, even the biased forty-eight input net in this case didn't match the performance of the unbiased net in the preliminary experiment (§5.7.1), in which biases made no per-

formance difference. The major difference between these cases was that the current networks, due to their number and the time required for their training, had been constructed with fewer hidden units -- and consequently fewer tunable parameters. For this reason, the experiment was repeated with a network with the same number of hidden units as had been used in the initial experiment. The superfluous I/O units for absent speakers and phones were, however, omitted. There were *n* input units, eight units in the first hidden layer and fourteen in the second, and nine output units. The biased network had twenty speaker ID inputs available via two hidden units in a separate layer.

These networks had the training[19] set classification performance given in Table 30. The unbiased raw input (48 input) performance was almost the same as in the preliminary experiment, although in this case the biased performance was slightly better, not worse, than the unbiased performance[20]. As before, the improvement in recognition performance was more

	Input Dimension													Raw
Case[a]	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**48**
Bias	66.9	74.0	77.5	78.4	79.1	80.4	81.3	80.7	81.7	81.1	82.1	82.0	82.4	85.7
Simple	59.8	66.5	71.3	72.7	74.9	75.8	76.3	76.4	77.4	77.7	79.9	79.1	79.3	84.0
% Redn	17.7	22.4	21.6	20.8	17.0	19.2	21.2	18.4	19.2	15.4	11.1	13.6	14.9	10.9

Table 32: Improvement in Recognition performance from Speaker Bias for RMSpell Vowel data projected to various dimensions.

a. This table gives the result for randomised pattern presentation only.

substantial when there was less speech data available for the classifier to work with. This effect is demonstrated in the following graph which displays the percentage error reduction available from speaker ID, plotted against input dimension, for both the networks with wide (8,14) hidden layers and the network with narrow (4,7) hidden layers:

5.8.3. Conclusions from Peterson Barney/RMSpell Comparison

These experiments were prompted by the observation that while the SVC-generating speaker models had been of unspectacular utility in the French Digits recogniser (§), and while speaker ID inputs had produced no improvement at all in the pilot RMSpell Experiment (§5.4), experiments in [watrous93] that ought to have served as a model for this kind of speaker "adaptation", using information about speaker identity, had shown impressive gains.

19. it is important to note that these results, with diminishing returns for speaker ID with increasing input data, are training results, since, while this is the practice of the experiments [watrous93] on which the current work was modelled, it is not, as we shall see later a wise one, if we wish to estimate the usefulness of speaker ID on real tasks. Even if the practice is justified when the data-sets available are very small, it would have been wise to abandon it immediately during the move to a larger data-set.

20. Why the effect if speaker ID bias on training set performance was not visible in the pilot run remains somewhat of a mystery. Since the biased performance of the presence network was better than the unbiased performance in the pilot, it is possible that the presence of spurious output units, trained to produce zero outputs, caused the network to find an initial solution to the problem that ignored the speaker IDs, and that could not be escaped during further training.

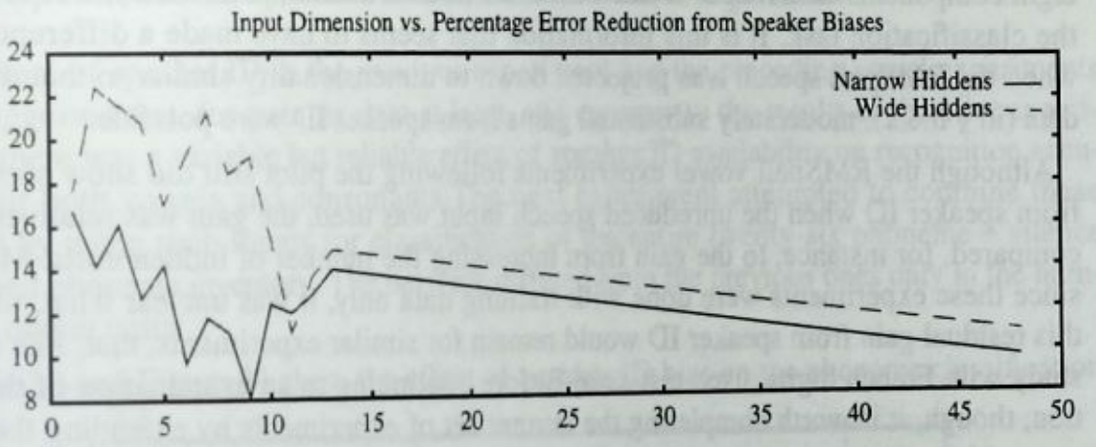


Figure 33: Error reduction vs. dimension onto which the three frames of input data were projected. The "Narrow Hiddens" case was for a network with four and seven hidden units in the first and second hidden layers, respectively. The "Wide Hiddens" network had eight and fourteen units in the corresponding layers. Especially in the case of the "Wide Hiddens" network, speaker biases produce a more substantial error reduction when less information is available from the speech directly.

The experiments in §5.6.2 showed that when speaker voice codes were derived from the Peterson and Barney vowel data (PB), they were effective at improving recognition performance on the task, although not as effective as perfect speaker information; if Peterson and Barney data could be used as a model for adaptation at all, it could be used as such with the sort of speaker "ID" provided by the SVCs investigated in this thesis. The almost complete failure of the pilot experiments with the more realistic Resource Management Spell-Mode (RMSpell) data-set suggested, therefore, that either the Peterson Barney data was not a good data set to work with as a model of real speech, or that there was something about the training done on the two data sets that made speaker adaptation work well with one, but not the other.

The experiments in §5.6.1 compared second order nets of the sort Watrous had used in [watrous93] with the more conventional first order nets used in the bulk of the experiments in this thesis. In no case did the second order nets provide any advantage over conventional ones. In fact, a conventional backpropagation network with speaker ID as additional input, with a 98.4% classification accuracy, outscored the best performance (97.5%) reported in Watrous's paper for a second-order net. Network architecture was clearly not the cause of the performance differences.

Moreover, the difference in input representation between the formant values used in the PB experiments and the spectra used for RMSpell did not appear to be an impediment to adaptation, at least for conventional nets, since converting the PB formant data to a spectral representation, while holding the amount of information constant (in §5.8.1) only slightly decreased recognition accuracy.

What did appear to make a difference was the amount of information available in the raw speech data. The PB data, containing only two formant values for each vowel, furnished only a very small, measured, amount of information that could be used for vowel identification. The speech used in the RMSpell experiments, by comparison, contained, along with information about formant position, a good deal of other information in the sixteen to forty-

eight components of the input vector. Some of this information, presumably, was relevant to the classification task. It is this information that seems to have made a difference, since when the RMSpell speech was projected down to dimensionality similar to that of the PB data (in § 5.8.2), moderately substantial gains from speaker ID were possible.

Although the RMSpell vowel experiments following the pilot still did show some effect from speaker ID when the unreduced speech input was used, the gain was relatively minor compared, for instance, to the gain from increasing the number of hidden units. Moreover, since these experiments were done with training data only, it was unclear what, of any, of this residual gain from speaker ID would remain for similar experiments, that, like the pilot study with French digits, used test data. Before continuing to an examination of that question, though, it is worth completing the current set of experiments by extending the task to recognition of constants, and to speech with a full phoneme inventory.

5.8.4. Speaker ID used with Consonants

Since it had been possible to demonstrate a small amount of utility in using speaker IDs to inform the recognition of vowels from the RMSpell database, but none at all for the all-phone recogniser in the pilot RMSpell experiment in §5.4 it seemed possible that speaker ID was unhelpful for consonant recognition, and that the rather small effect that had been shown for vowels had been diluted by the presence of consonants in the all-phone case. This hypothesis was also consistent with the widely held notion that voice quality or voice personalty is expressed more strongly in vowels.

	Input Dimension												Raw
Case	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**48**
Bias	41.9	68.0	68.5	73.1	73.0	75.6	76.3	76.0	76.6	77.5	77.1[a]	78.3	79.8
Simple	31.2	58.8	63.2	65.4	67.3	68.7	70.4	70.2	71.8	70.6	71.8	72.1[b]	75.1
% Redn	15.5	22.4	14.3	22.2	17.4	22.0	20.0	19.5	17.0	23.4	18.8	22.1	19.2

Table 33: Improvement from speaker bias in recognition performance for RMSpell Consonant data projected to various dimensions.

a. Training stopped after 5 600 epochs for this case. Others trained for 5 800 epochs.

b. Training stopped after 5 400 epochs for this case

A network like the one used in the vowel case, but with output units corresponding to the seventeen consonants, was trained to classify consonants extracted from the same sixty sentences used in the vowel experiments. Surprisingly, as shown in Table 30, the effect of speaker ID on recognition accuracy was actually stronger than it had been for vowels, and, in contrast to the vowel case, fairly stable across input dimensionalities, remaining present even when all the information in the three frames of speech was available.

5.8.5. Speaker ID used with Vowels, Consonants and Silence.

The effect of speaker ID in the previous experiment and the preceding vowel experiments had suggested that, for training data at least, and contrary to the results of the pilot experiment, there was a variable but reliable effect of speaker ID availability on recognition accuracy for both vowels and consonants The next experiment attempted to combine these results by using such inputs for classification of the entire twenty-six phoneme + silence RMSpell phoneme inventory. The network differed from the previous ones only in the number of output units.

Table 30 and Figure 34 show the effect of speaker ID bias on the phoneme classification

Case	Input Dimension						Raw
	1	2	4	6	9	13	48
Bias	48.7	65.7	71.9	74.8	75.2	77.0	78.1
Simple	44.6	59.9	68.2	71.9	73.6	74.9	77.0
% Error Reduction	7.5	14.5	11.9	10.4	6.0	8.3	4.7

Table 34: Improvement in Recognition performance from speaker ID bias for RMSpell data. This experiment used the full phoneme set. As usual, three frames of input were projected to various dimensions. The networks had 8 and 14 hidden units respectively in two hidden layers.

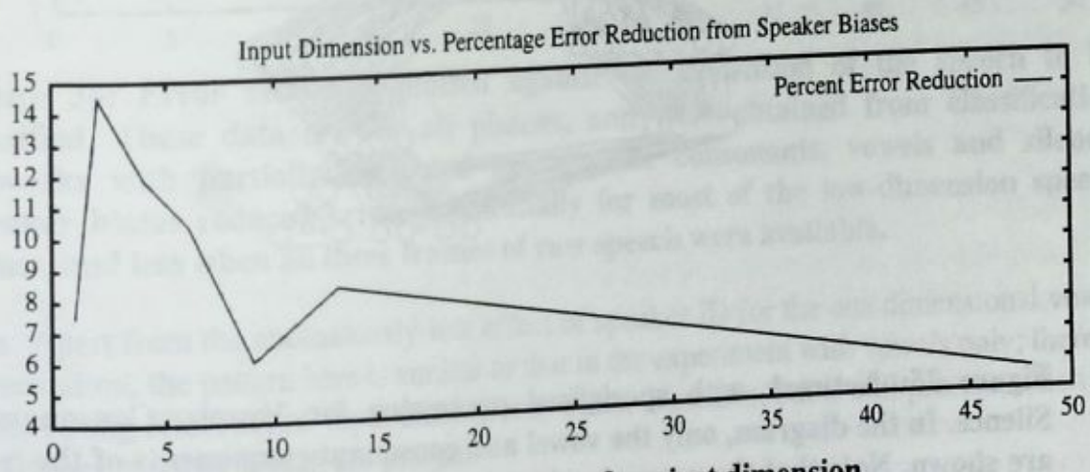


Figure 34: Error reductions of Table 30 plotted against dimension.

performance of this recogniser. Speaker biases helped considerably less in this case than in the recognisers specialised for vowels or consonants. Analysis of confusion matrices for the forty-eight input case suggested that the speaker ID biases were allowing the network to improve vowel recognition, but that this was achieved at the cost of reduced accuracy on consonants. While the number of vowels correctly classified increased with the bias, the number of correctly classified consonants decreased. These changes are enumerated in Table 35. Since, in the previous experiment, biases were more effective at improving the recogni-

tion accuracy for consonants than for vowels, it appeared that the effects of bias on vowels and consonants might be incompatible.

Vowel	AA	AH	AX	AY	EH	EY	IY	OW	UW								
	30	-	-	15	44	36	-	-3	20								
Consonant	B	CH	D	F	JH	K	L	M	N	P	R	S	T	V	W	Y	Z
	-13	-	5	-1	-7	-	41	-20	-24	-	-12	13	-22	-	1	18	-

Table 35: Speaker biases appear to affect vowels and consonants differently. This table summarises the change in the number of correctly classified frames for each phone when speaker information is available. In a unified network, improvements in the number of correctly classified inputs for vowels were largely offset by decreases for consonants. Cells containing "-" exhibited no change in recognition accuracy when speaker biases were used.

5.8.6. All Phones, split nets - Separate Vowel, Consonant and Silence Nets

In an effort to prevent any such interference, it was worthwhile to experiment with a system in which the network was divided up, below the level of the output units, into separate recognisers for vowels, consonants and silences, all of which received the same input. The architecture is illustrated in Figure 35. This splitting of the network was intended to force

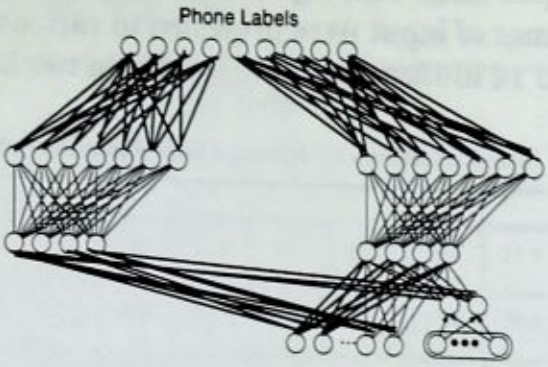


Figure 35: Network with specialised processing for Vowels, Consonants and Silence. In the diagram, only the vowel and consonant components of the network are shown. Note that the networks share a common input, and a common output layer, but that the hidden units are specialised for vowels and consonants (shown) and silence (omitted from the diagramme).

the lower layers of the network to adapt, if at all, in ways relevant to the speech sound being specialised for. It is important to note that in this architecture, since the three networks were still competing with each other to produce the highest output, the training was fully discriminant across the entire phone set.

Table 30 and Figure 36 give the effect of biasing the network as in the previous experi-

Case	Input Dimension						Raw
	1	2	4	6	9	13	48
Bias	48.7	65.9	73.3	76.3	77.7	80.1	81.1
Simple	44.6	59.9	68.6	72.6	73.7	76.6	79.2
% Error Reduction	7.4	15.1	14.9	13.3	15.1	15.0	9.1

Table 36: Improvement in Recognition performance from Speaker Bias for RMSpell data for all phones projected to various dimensions, using split nets (8,14 hiddens)

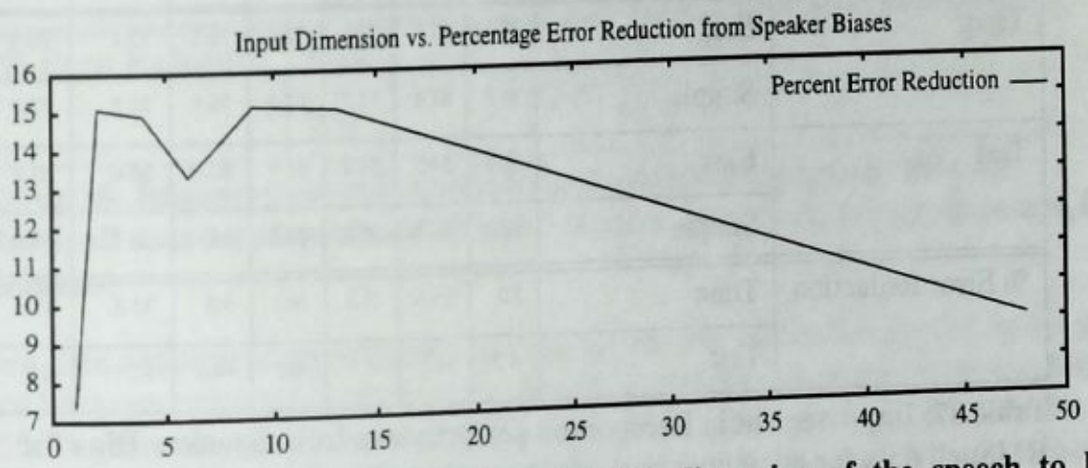


Figure 36: Error reduction plotted against the dimension of the speech to be classified. These data are for all phones, and were obtained from classification networks with partially separated modules for consonants, vowels and silence. Speaker biases reduced error substantially for most of the low-dimension speech inputs, and less when all three frames of raw speech were available.

ments. Apart from the anomalously low effect of speaker ID for the one dimensional vowel representation, the pattern here is similar to that in the experiment with vowels only; there is a rather strong effect at low input dimensions, which decreases, but is still present, when all three frames of input speech are present.

Discussion

Although a true speaker model was not being used, and although the experiments with speaker models for the Peterson and Barney data-set (§5.6.2) had suggested that the effect of such models would be smaller than that of the idealised speaker-ID "model" used here, and although the effect of bias was less than for the Peterson Barney data, it had, at least, by this stage, been possible to obtain an effect of speaker bias on phoneme recognition for all phones, on *training data*. This matched, more or less, the status of Watrous' work with Peterson & Barney data. Of course, if adaptation by biasing, whether by ideal inputs or by speaker models, was to be of any practical use, it would have to improve performance on

untrained speech as well as on the training set, and, in the case of speaker models, on untrained speech from novel speakers. The next experiment was designed to test for such an effect.

5.8.7. Testing the Speaker ID effect for Generalization

In order to see how well the effect of bias would generalise to other speech from the same speakers, the experiment was repeated. This time three utterances of training speech and one of testing speech were used for each speaker. As before, the networks were split into vowel, consonant and silence subnetworks internally. Table 30 gives the recognition accuracies for

Case		Input Dimension						Raw
		1	2	4	6	9	13	48
Train	Bias	47.1	67.4	76.9	72.8	79.1	75.1	76.1
	Simple	44.3	61.4	73.8	67.5	76.9	70.5	72.9
Test	Bias	36.9	54.2	58.0	61.9	62.0	66.0	65.8
	Simple	38.0	56.0	60.0	62.2	63.6	64.1	64.9
% Error Reduction	Train	5.0	15.6	11.9	16.3	9.6	15.8	11.6
	Test	-1.7	-4.2	-5.1	-0.8	-4.6	5.2	2.7

Table 37: Improvement in Recognition performance from Speaker Bias for RMSpell data for all phones projected to various dimensions, using split nets. In this case, three utterances of training and one of testing data were used for each speaker.

a number of input dimensions.

The difference between training and test set performance was wider than one might have expected. In the unbiased condition, the misclassification rate increased almost 30% for the test set, suggesting that the network might have been overtrained. More interestingly, there was an odd effect of the speaker ID information on the lower dimensionality versions of the test set (projected along the same axes as the training set). It would seem, at these dimensionalities, there was no generalisation at all of the large effect that speaker ID had on the training set. Even in the higher dimensional cases, the effect of speaker ID on the test set was rather small.

Given that these networks seemed to be failing to generalise well both with respect to the simple classification task itself and with respect to the effect of speaker ID biases on performance, it appeared that it would be worth while to construct a training set with more data and to run this experiment again.

More Training data

This experiment used six training utterances for each speaker, but was otherwise identical to the previous experiment. Table 30 displays the performance of this network in the usual

Case		Input Dimension						Raw
		1	2	4	6	9	13	48
Train	Bias	43.9	62.5	70.0	73.0	74.4	75.7	78.3
	Simple	41.7	58.9	66.5	69.7	71.8	73.8	76.2
Test	Bias	49.1	64.9	69.4	72.7	74.1	74.7	76.5
	Simple	49.3	62.5	68.8	70.9	73.2	74.2	74.0
% Error Reduction	Train	3.7	8.8	10.6	11.1	9.3	7.5	8.7
	Test	-0.2	6.6	2.2	6.0	3.2	2.1	9.6

Table 38: Improvement in Recognition performance from Speaker Bias for RMSpell data for all phones projected to various dimensions, using split nets (8,14 hiddens)

manner. The addition of more training data per speaker was successful at reducing the difference between training and testing set on the raw recognition task. In fact, there was better than 100% generalisation for some of the lower dimensionality results, probably just due to chance differences between the composition of the training and testing set. Figure 37. shows

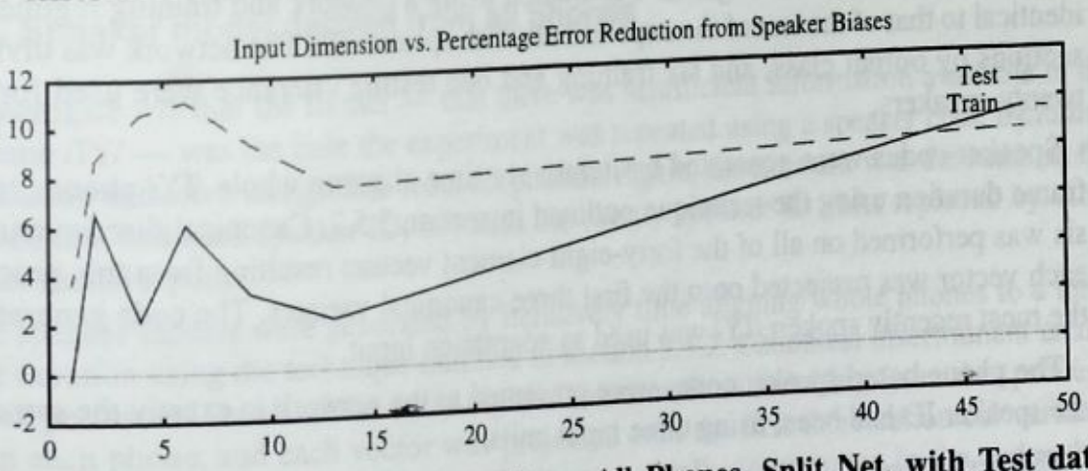


Figure 37: Error reduction vs. dimension - All Phones, Split Net, with Test data, trained on twice as much data as the previous experiment.

the percentage error reduction from Speaker ID biases, for this data. Although there was little sign of overtraining in the recognition performance, the effect of speaker ID on performance for the test set was still generally lower than for the training set. The speaker dependent characteristics of the data-set that the network is able to learn seem to be less robust than the speaker independent ones.

Although, again, in this experiment, as in the experiments where only training set classification performance was measured, there was some evidence that having access to information about speaker identity had a useful effect on recognition performance, even in these cases of perfect speaker information, the effect was slight. It was certainly too slight to justify any particular confidence that these sorts of connectionist classifiers, or full recognisers built on top of them, would furnish a practical tool for measuring the performance of competing imperfect speaker models, where the effects on classification performance, if present at all, would be more or less guaranteed to be smaller.

5.9. Using a Speaker Model in Recognition

Despite lack of a particularly promising outlook for the activity, an attempt was made to apply speaker models derived from speech, rather than speaker IDs, to the task of improving speech recognition. The speaker models used here differed somewhat from most of those described in the previous chapter, since not all of the work described there preceded these speech recognition experiments.

5.9.1. Information derived from only the phoneme /IY/.

Since full speaker models are somewhat complex, it is difficult, when using them, to have an accurate understanding how much information had been presented to the recogniser. Following a suggestion form Bridle [Bridle, 1993, personal communication], it seemed worthwhile to see whether any effect on recognition performance could be obtained from a single phoneme. The phoneme /IY/ was used for this purpose, since it is relatively frequent, and because, since it is a vowel, it should contain information about voice personality.

In this experiment, a recogniser was trained using a network and training regimen nearly identical to that of the last of the experiments with speaker ID: The network was divided into sections by output class, and six training and one testing utterance were used for each of twenty speakers.

Speaker codes were generated by iteratively time aligning whole /IY/ phones to a three frame duration using the technique outlined in section 3.5.2. Canonical discriminant analysis was performed on all of the forty-eight element vectors resulting from this process, and each vector was projected onto the first three canonical variates. The code generated from the most recently spoken /IY/ was used as adaptation input.

The phone based speaker codes were presented to the network in exactly the same way as the speaker ID had been, using three input units.

The performance of this network after training is given in Table 30.

There was no consistent improvement in classification accuracy for networks given the /IY/ bias.

Case		Input Dimension						Raw
		1	2	4	6	9	13	48
Train	Bias	42.4	58.9	66.1	68.3	70.4	71.3	73.5
	Simple	41.6	58.6	65.7	68.2	69.3	71.1	74.2
Test	Bias	49.5	61.9	68.2	70.9	71.8	71.7	72.8
	Simple	49.2	61.8	68.6	70.2	71.2	72.2	73.6
% Error Reduction	Train	1.4	0.7	1.2	0.3	3.6	0.7	-2.7
	Test	0.6	0.3	-1.3	2.3	2.1	-1.8	-3.0

Table 39: Improvement in Recognition performance from Speaker Bias for RMSpell data for all phones projected to various dimensions. In the Bias condition, a speaker code derived solely from the information in instances of the phoneme /IY/ was available to the network.

Although it is hard to doubt that there is meaningful information about speaker ID in the /IY/ phones, it was either insufficient to make a significant difference to the performance of the phone classification task, or it was information that the network could get from elsewhere in the three frames of speech it was attempting to classify.

5.9.2. Speaker model derived from all phones

To investigate whether the former — that there was insufficient information available in the phoneme /IY/ — was the case the experiment was repeated using a speaker code including information extracted using CDA from all phones. Again, the network was the same as the network that had used speaker ID, but with the twenty speaker ID units replaced by three speaker model inputs.

The speaker models were generated by iteratively time aligning whole phones to a three frame duration using the technique outlined in section 3.5.2. Canonical discriminant analysis was performed on all of the forty-eight element vectors resulting from this process, within each phone, and each vector was projected onto the first three canonical variates. These phone models were then inserted into an overall input vector in the order they appeared in the database, resulting in a sequence of eighty-one element vectors[21]. These

21. Three elements per phone, multiplied by twenty-seven phones.

vectors were, again, subject to a CDA by speaker, yielding a three element speaker code.

Case		Input Dimension						Raw
		1	2	4	6	9	13	48
Train	Bias	41.8	59.4	67.1	70.5	72.3	73.7	76.1
	Simple	41.7	58.9	66.5	69.7	71.8	73.8	76.2
Test	Bias	49.5	62.8	68.8	71.8	73.1	73.5	75.3
	Simple	49.3	62.5	68.8	70.9	73.2	74.2	74.0
% Error Reduction	Train	0.2%	1.2%	2.1%	2.6%	1.8%	-0.4%	-0.4%
	Test	0.4%	0.8%	0%	3.1%	-0.4%	-2.7%	5.0%

Table 40: Improvement in Recognition performance from Speaker Bias for RMSpell data for all phones projected to various dimensions. In the Bias condition, three units are used to present a CDA based SVC derived from previous speech.

that changed on every phone boundary. This code was fed into the three speaker model input units of the recogniser.

Recognition results for the training network are given in Table 30. There was no consistent gain in recognition performance from using the SVC, either for training or testing data. This is consistent with the prediction that, since the SVC is, almost certainly, only an imperfect approximation to speaker ID, which itself produced only a small change in recognition performance, any effect of the speaker model based biases would be difficult to detect.

5.10. Conclusion

Information about speaker identity, whether derived from a speech signal or supplied directly, can contribute to improving recognition performance, and these gains can be obtained even within relatively simple connectionist architectures. However, in recognisers using powerful classifiers and sufficiently rich input representations, the gains can be less than spectacular. When a classifier for all phonemes was applied to the RMSpell database, in fact, speaker information derived from speaker models had an effect that was too weak to be detected, if it had an effect at all.

The work described in this chapter to bridge the gap between the very strong effect of speaker information in previous work with the Peterson and Barney vowel database and the rather weak effect of similar information used in realistic tasks, suggested an important *caveat* for speech researchers: If simplified speech signals are used, care must be taken to ensure that speaker adaptation is providing information that cannot be obtained from the speech signal over short durations, rather than simply replacing information which has been lost from the speech signal by the chosen input coding.

These experiments also served to expose some weaknesses in speech recognition as a vehicle for comparing speaker models. Clearly, the major difficulty was the fact that building recognisers that can be adapted by biasing or other forms of modulation, as opposed to adaptation by retraining, is a difficult problem in itself, quite apart from the matter of whether or not that modulation is derived from previous speech, and how. Even if that problem were solved, though, and a suitable recogniser could be built, it would still be true that recognition is a rather opaque task. If the speaker models were successfully applied, the only evidence would be an increase in classification or transcription accuracy, perhaps broken down by speech unit. This would offer a poor vehicle for investigation of such questions as whether the speaker model is perceptually relevant.

To provide a more transparent application, where the influence of changes in speaker model would be directly visible (or, rather, audible), and to investigate another application area for voice models, a series of experiments in mimicry synthesis by voice transformation were carried out. These are described in the following chapter.

Chapter 6. Synthesis By Voice Transformation

Speech recognition had not proved to be as good a test bed for speaker modelling as one might have hoped, and, even if it had been, the effect of the speaker models would have been opaque. If a clear effect had been present, it would only have been visible as a change in a recognition score, making it difficult to understand how the speaker model was encoding speaker differences. For both these reasons, but chiefly the former, a new application where the effects of modelling would be more transparent was chosen. That application is the transformation of one voice into a set of other voices, with the target voice being described by the speaker model.

Although the main purpose here was to provide an environment for evaluating the speaker modelling system, and although none of them would be adequately served by the system as it stands, one can imagine practical uses for this sort of mimicry synthesis:

- In speech synthesis, it is important to provide a voice that the listener finds agreeable; It would be very convenient to be able to select the voice one wants one's computer to use simply by playing it a sample of that voice.
- In Speech-to-Speech translation systems such as Janus [waibel91], it would be desirable to produce the translated speech in the original speaker's voice, both for aesthetic reasons, and because of the obvious utility of being able to distinguish multiple speakers in conference calls and other meetings.
- Using an inverted form of voice transformation, it would be useful for automated voice-response systems (such as voice-mail prompts) to be able to utter user-recorded segments (such as the mailbox owner's name) in the same voice as the standardised prompts, rather than the voice of the original speaker.
- Works of interactive fiction would be enhanced if the characters inhabiting them could be given a variety of realistic voices.

6.1. Other voice transformation systems

Despite the existence of these potential applications, there has been relatively little work done in this area. The work that has been done has focussed entirely on the problem of transforming the speech of a single source speaker into that of a single target speaker, although voice normalisation, the attempt to transform the speech of a number of speakers into that of a single reference speaker, for use in speech recognition, could be considered closely related.

Transformation with manual intervention

One of the earliest approaches to the problem of voice transformation was directed more towards the central problem of this thesis, understanding the differences between speakers.

Childers et al. [childers85,childers89] looked at converting single sentences from a single pair of speakers, one male and one female, into the voice of the other, using an analysis-synthesis system. The system allowed them to modify the pitch, spectral expansion, and glottal pulse shape of the source speech to more closely match the target speech. Based on their hypothesis that it is the accurate rendering of transients that is important for intelligibility, and the accurate rendition of steady voiced segments that is chiefly responsible for voice personality, they performed the transformation entirely in these latter segments, simply copying the other segments from the source speaker. They found that by adjusting the pitch and the spectral expansion they could produce many of the characteristics of the target speaker, but that non-linear spectral expansion, different spectral expansions in different segments, and altering the shape of a glottal pulse produced by a parameterised model all improved voice quality.

It is difficult to evaluate this work as a practical technology, since it was aimed at discovering what factors in a voice were essential to its personality. What it does demonstrate is that one needs a flexible spectral transformation, rather than a fixed normalisation, to achieve good quality, and that ultimately, voice transformations — and speaker modelling systems that support them — are going to have to account for speaker differences in the excitation signal.

Codebook based transformations

One of the earliest papers describing a voice transformation technology [shikano86], was, in fact, directed not towards producing speech matching a particular speaker, but towards frame-wise normalisation of speech from multiple speakers, allowing it to be used as input to a speaker dependent speech recognition system. Since this system formed the basis for the largest coherent body of work in voice transformation [abe88, abe89, abe91a, abe91b, mizuno94], it is worth discussing in some detail, omitting details of the normalisation scheme that did not survive in later incarnations.

In this system, the speech from the target speaker and the source speech were LPC encoded, and the LPC spectral coefficients vector quantised[1], separately for each speaker. The speech was then aligned by Dynamic Time Warping alignment [nye84], and, for each source codebook entry, a histogram was made of target codebook entries that were aligned with it. These histograms were used to create new entries, one for each entry in the source speaker's codebook, that were linear combinations of the target codebook templates, weighted by the histogram counts. This new codebook was used to quantise the speech, and the process was iterated, several times. Each source codebook entry then had a corresponding "mapping" codebook entry that was simply an average of the speech frames that were aligned with that frame.

In the case of the recognition system they were investigating, a single target speaker was used, and mapping codebooks were used to move the speech of several source speakers

1. Vector quantising a signal involves choosing a fixed-size set of templates into which a signal can be decomposed. The signal can then be efficiently transmitted by sending these templates once, followed by a sequence of template indices from which it can be reconstructed. Vector quantisation is widely used in speech compression, and, more relevantly here, in HMM based speech recognisers.

towards the target, resulting in an improvement from 64.0% to 83.1% in recognition accuracy.[2]

The development of this system for voice transformation will be discussed in the following paragraphs, but it is worth pointing out at this point that the quality of the transformation that can be produced is highly dependent on how good an alignment can be achieved between the source and the target speech. Imperfect alignments will result in mapping codebook entries that are smeared over many different target frames, resulting in poor quality synthetic speech. Most, if not all, of the papers discussed here have chosen or selected data in such a way that good alignments are easier to achieve[3], a luxury of choice that this author would have liked to have had available, considering the efforts that had to be exerted towards improving the alignments used in this synthesis task.

After their initial success with voice transformation for recognition, Abe *et al.* applied the VQ transformation technique to speech synthesis, with the eventual goal, shared by the work in this thesis, of retaining voice personality during speech-to-speech translation [abe88]. In this case, the VQ codebooks were supplemented by scalar quantised codebooks for pitch and power. The training samples were one hundred Japanese words per professional speaker used. Each word was uttered in isolation, again, making alignment reasonably straightforward. Voice transformation was achieved by simply applying the three mapping codebooks for the speaker pair in question and resynthesising.

The performance was measured both objectively and subjectively. The objective measure used was the difference in vector distortion between the transformed source speech and the target, compared with the original source speech and the target, and in the absolute size of the average pitch difference between the transformed pitch and the target. On the first measure, the transformation reduced distortion by 27% for transformations between two female speakers, of 49% between two male speakers, and by 66% between a male and a female speaker. The transformation reduced the average pitch disparity to less than 15Hz. Extensive use will be made of distortion measures like this in reporting the results of this chapter.

The subjective measures used involved presentation of the original and transformed speech to human subjects who were asked to make similarity judgements. Two experimental setups were used. In the first, the original speech, the target speech, and three versions of converted speech — one of which was produced missing the pitch and one missing spectral conversion — for a male-female speaker pair were presented in a set of forty randomly assigned pairs. The listeners were asked to make similarity judgements on a scale of "similar", "slightly similar", "difficult to decide", "slightly dissimilar' and "dissimilar". A multidimensional scaling technique [hayashi85] was used to display the similarity relations. This technique suggested that not only did the full conversion move the male speaker perceptually towards the female speaker, but that pitch and spectral alterations both made an important, independent contribution towards the perceived similarity of voices.

The second subjective measure was used to assess conversion between two male speakers. Words A and B from the two speakers M and N respectively were played, followed by a word X resulting from an M->N or N->M conversion. Listeners were asked to judge which

2. While we shall not dwell overly on matters we left behind in a previous chapter, it should be noted that the recogniser was trained in Speaker Dependent mode and used in Multiple speaker mode.

3. The Shikano et al paper [shikano86] for example, rejects training sentences for which the alignment cost is high. Rejecting outliers this way is an excellent practice, but is only possible if one has more data than one needs from each speaker.

token A or B the voice in token X most closely resembled. The best of the voice conversion efforts explored later in this chapter was assessed using this second listening test.

The main purpose in reviewing this work in VQ transformation was to introduce the problems of alignment and to explain techniques for assessing the quality of voice conversions. For the sake of completeness, however, and to preview some of the techniques that are applied here in different contexts, it is worth mentioning more recent versions of the system.

Since the system was intended for use in a speech-to-speech translation system, an attempt was made to determine whether there were consistent differences between speech in two languages that should be modelled during conversion [abe90, abe91a]. Codebooks were created for speech from a single, bilingual speaker speaking Japanese and English, and both codebooks were used to encode speech from both languages. Although differences were found, they were insignificant compared to interspeaker differences. It should be noted, however, that although the speaker was, apparently, at "native speaker" proficiency in both languages, there is a possible confound stemming from the use of a single speaker. Bilingual speakers tend to move their pronunciation of both languages towards an intermediate phonology [flege94]. Differences between languages may therefore be greater than those measured.

In the same papers [abe90, abe91a], Abe *et al.* introduced an application of voice transformation to speech synthesis, by applying it to the output of the MITalk [allen87]. speech synthesiser. This is similar to the technique adopted in this thesis. The synthesiser produced isolated Japanese words matching those uttered by a Japanese speaker. The phoneme string to be uttered was chosen by hand. MITalk synthesised it using American English phonological rules, except for duration rules, which were modified to more closely match those for Japanese. Mapping codebooks were derived from alignments of the MITalk "Japanese" words with the human speech, and used on the output of MITalk speaking American English, to produce English in the Japanese speaker's voice. Although the authors expressed some reservations about the quality of the speech produced, it is clear that applying target-speaker based transformations to synthetic speech is a useful path to take, if voice conversion is to be of significant technological import.

In [abe91b] this work was taken further. Instead of converting the speech frame by frame, whole segments were transformed. In this manner, it was possible for the system to convert both the static spectral qualities of the speech and the within-segment dynamics. This technique reduced the spectral distortion between the converted and target speech for a pair of males to one third of the distance between the original voices, and improved speaker ID accuracy by 20% over that of a frame by frame conversion.

A final VQ based paper [mizuno94] is worth discussing because it suggests why applying a universal function approximator, such as a neural net, to the problem of voice conversion, is a good idea. While this is not what Mizuno *et al.* did, their use of piecewise linear models of voice quality is a step in this direction. In this voice conversion system, each codebook entry for the source speaker's voice was analysed for formant frequency and spectral tilt. A linear model converting these values for members of the codebook entry into those of the target speaker was derived. During conversion, this model was used to specify formant frequencies and spectral tilt for the target speaker, and minimal spectral distortion (MSD) search was used to find the nearest matching frame for the target speaker. Target speech was

then resynthesised from this frame. Although the speaker ID accuracy for the resulting speech was actually slightly less than for VQ converted speech, perceived naturalness of the voice was higher. This suggests that smooth spectral transitions, of the sort that functional approximators can generate, are important for natural speech. If a neural network can be trained to produce the target spectrum with adequate fidelity, its smooth interpolation between frames should result in speech that is more natural than that from a system using VQ coding.

Neural net methods

In fact, some recent work has sought to take advantage of this power, at least with respect to the source speech. Nam and Savic [nam90][savic91] constructed a system that used a neural network *classifier* to select a target codebook entry given a frame of LPC coefficients of the source speaker. The training data was generated by doing a forced alignment, assisted by using voicing decisions, in the LPC cepstrum parameter space. During synthesis, the LPC coefficients from the target codebook were used either with a pulse train or with a frequency shifted version of the source excitation signal to produce the transformed speech. The authors reported that the modified source excitation signal produced higher quality speech.

It is not clear why the authors used the network as a classifier, rather than as a function approximator, although one might imagine that it was in an effort to reduce smear in the output LPC coefficients[4]. Besides this difference, the system that will be developed in this chapter could be seen as an extension of Nam and Savik's system to the use of synthetic source speech and plurispeaker synthesis using speaker model input.

A more sophisticated synthesis scheme

Valbret *et al.* [valbret92b] propose a scheme that treats the two main contributors to segmental variation, the voicing source and the spectral characteristics, separately. Their work was also motivated by the notion that a "speaker can be characterized by a 'spectral print' in some parameter space", although they do not try to construct such a model explicitly. In their review of the work of Abe *et al.* and Nam *et al.*, they claim that the voice quality in those systems was limited by the use of the LPC vocoder. In their system, the authors decomposed the input signal into the excitation signal, and a series of pitch synchronous LPC spectral envelopes. The spectral envelopes were divided into disjoint acoustic classes and either a dynamic frequency warping (DFW) or linear multivariate regression (LMR) based transformation was learned between the source and target vectors in these classes. The target speakers's excitation signal was used together with the transformed spectral envelopes were used to produce a target waveform for each pitch period, and the PSOLA synthesis technique was applied to combine these waveforms into the target speech signal. Because prosodic modelling was considered to be beyond the scope of the paper, the actual pitch and timing of the target signal were applied to the transformed speech[5].

4. Although this is a reasonable hypothesis, a preliminary attempt to do the same in the voice transformation system developed here produced no improvement in the synthesised speech compared to direct transformation. Since time was limited and a detailed comparison of voice transformation systems was somewhat peripheral to the problem of model based synthesis, this attempt is not reported in detail here.

Training was performed using specially recorded CVC logatoms, which were used because the authors were unable to obtain sufficiently accurate time alignments for training speech consisting of simple noun phrases[6]. This difficulty is mentioned here because it is likely that alignment problems were partially responsible for the comparatively low quality of the transformed speech produced in the experiments reported in this thesis.

Of the two techniques, DFW and LMR, the authors reported that LMR produced speech that more closely matched the voice of the target speaker. This is promising for the neural network techniques used here, since, as has been pointed out in great detail earlier, the use of a neural network as a functional approximator is exactly a non-linear multivariate regression and often degenerates to being exactly an LMR model. In the current system, however, instead of dividing the acoustic space into non-linear regions and hoping for piecewise linearity within those regions, the system relies on the possibility of non-linear mapping by the network to build a unified transformation of the entire acoustic space.

6.2. Introduction to the Experimental work

The remainder of this chapter is a description of experiments in which a voice transformation system with multiple target speakers was developed, modified to use the SVCs described in chapter 4, and then evaluated by human listeners.

First, though, Section 6.3 briefly describes an attempt to produce the target speech directly from the speaker models, rather than by transforming another voice. Since this attempt was unsuccessful, the system was built by producing a neural network function approximator that converted the output of a commercial text-to-speech system into the target voice. This allowed the transformation network to concentrate on altering only the spectral characteristics of the speech, about which the voice code could be expected to contain useful information.

The voice transformation network was rather similar to the speaker dependent recognition networks described in the previous chapter, but trained to produce an LPC frame in the target voice instead of a phoneme label. Frames output by this network could be synthesised by the LPC10 vocoder. Details of the network, and of the alignment between the input and the target speech, are given in Section 6.5 to 6.7. Initial evaluation of the transformation network (Section 6.8) showed that, while the quality of the speech produced was poor, the neural network was able to move both the pitch and the filter characteristics of the input speech closer to those of a single target speaker (Section 6.8.2). In the experiment described in Section 6.8.3, the transformation network was given *1-from-n* speaker IDs, and was able to move the speech closer to that of any chosen target speaker from a set of five.

Section 6.8.4 describes an initial attempt to drive the transformation from a speaker code. For this experiment, the code was generated by averaging eight-element PCA-based phone codes for each speaker, and then compressing the concatenated average phone codes in a bottleneck network. This yielded one four element speaker code per speaker. Comparison of the output of a transformation network trained to use this code with the natural speech of a

5. Given the importance that pitch and timing seems to play in determining speaker identity, one would expect this to improve the perception of target speaker identity in the transformed speech markedly.
6. These noun phrases consisted solely of an article, an adjective and a noun.

set of speakers, including the target speaker, showed that the network produced speech that was, on average closer to that of the target speaker than to that of other speakers.

Finally, a voice transformation network was built that was designed to use the actual speaker codes from one of the speaker models described in Chapter 4. This, network, whose construction is detailed in Section 6.9., could produce speech intended to mimic that of any speaker in the TIMIT speaker set. Testing, again by measuring the best alignment distance between the output speech and natural speech from a large group of speakers, including the target speaker, showed that the transformation was moving the speech towards the specified target on average. This system produced the best quality output speech, presumably because the amount of training speech was much larger than for the other transformations, but the quality was still poor. Because voice transformation *per se* was not the goal of the thesis, it was not possible to invest the work that would have been required to produce high quality speech from the basic transformation.

After the model-based transformation system had been built, a final series of experiments was run to see whether human listeners could identify the output speech with the voice of one of a pair of speakers. ABX designs were used in all cases. In the first of these experiments, described in detail in Section 6.11, speech was produced at every stage of the conversion process between natural and DECtalk speech to asses how each step affected voice personality. Even LPC coding the natural speech and time aligning it with the synthetic speech significantly affected the perception of voice personality. For the speech output by the transformation network, the target speaker could be identified only in the case where the other speaker choice was not the same gender. This finding was consistent with the earlier indication that most of the useful information in the task independent speaker models concerned speaker gender. What remained to be seen was whether this distinction was supported entirely by the network's ability to change the pitch of the DECtalk speech. In Section 6.11, the subjects compared the output of the transformation with DECtalk speech whose pitch contour had been replaced with that of the target speaker or a different speaker. Subjects were able to distinguish the speakers when their sex differed, using no other information than pitch. In Section 6.13, an attempt was made to see whether, given that pitch was sufficient to account for the amount of voice personality in the transformed speech, it was also necessary. The output of the speech transformation, leaving pitch unconverted, was compared with human speech that had been time aligned with the DECtalk utterance, and that had had its pitch contour replaced with that from DECtalk. Deprived of pitch information, subjects were unable to identify the correct speaker, even when the speaker gender differed.

Although the voice transformation achieved was not of high quality, and served mainly to confirm the difficulty of forming task-independent speakers models of general utility, it is likely that it could be improved. With sufficient engineering work, some suggestions for which are given in the conclusions for this chapter, it seems likely that a plurispeaker voice transformation system could be built with quality comparable to voice transformation systems reported in the literature.

The following sections describe experiments in more detail. A reader wishing to gain only an overview may wish to skip forward to the conclusions in Section 6.14 on page 166.

6.3. Initial work - direct synthesis from models

Since the aim of working with synthetic speech was primarily to provide a test-bed for the speaker modelling work, some initial experiments were done using the rather radical approach of trying to synthesise speech directly from a speaker model. The speaker models that are built using compression are invertible, in the sense that they can be used to estimate the input that produced the low dimensional representation representing the speaker's voice. If they are based on a representation, such as LPC coding, allowing convenient resynthesis, they can be used "in reverse" to produce speech. Although these experiments were done using earlier, pilot versions of the speaker models, and might be improved by the use of the final speaker model, neither of these methods produced even marginally acceptable speech quality, and they are chiefly included here to as an explanation for the move to voice transformation. Two methods for producing speech directly from speaker models were tried:

6.3.1. Inverting the model all the way through

First, the overall speaker model was expanded to produce estimates of each of the phone codes. Then the phone codes were each expanded to produce fixed length vectors in the same vector space as the original speech. These vectors were simply concatenated to produce the desired phone sequence, and the result resynthesised. The resulting incoherent babble discouraged pursuit of this method of synthesis.

6.3.2. Concatenating segments

Given the recent success of systems doing concatenative synthesis [sorin91, hauptmann94], it seemed reasonable to expect that the speech quality could be improved if, instead of using the inverted model speech directly, this speech was replaced by that section of actual speech from the database most closely matching the speech produced from the model. The speech was additionally constrained to match the phonetic context desired, so long as sufficiently many choices were available with that context. For this experiment, the PSOLA technique, which has enabled much of the recent success of concatenative synthesis, was not applied. It didn't seem warranted for the initial experiment, and as it turned out, the speech produced by segment concatenation was not noticeably better than that produced by direct model inversion. It is unlikely that PSOLA would have improved it significantly.

6.4. Voice transformation.

It had become clear at this point that, even if the speech produced by direct model inversion was matching speaker voice characteristics at a segmental level, the underlying *synthesis* technology of naive concatenation would never produce a convincing demonstration that speaker modelling was a useful technology for practical purposes. However, as mentioned in the introduction to this chapter, other researchers had claimed good performance from systems that transformed the voice of one speaker into that of another. It is clear why doing this transformation should be relatively straightforward, compared to the task of synthesising a new voice whole. One is not forced to reproduce every characteristic of the speech — the

information bearing properties, including the sequence and articulation of phones, and the pitch and amplitude contour can be taken from the original voice and altered as necessary. They do not need to be created anew starting only with a string of phone labels, they only have to be augmented with the voice characteristics of the target speaker. Abe et al [abe91a] used similar reasoning, pointing out that in attempting to produce English speech in a Japanese voice, it was necessary to retain dynamic characteristics of MITalk speech, which help the speech sound like English, while changing the spectrum into that of the target speaker, since static spectral characteristics provide clues to speaker identity.

Although the goal of producing any modelled speaker's voice as the target was somewhat more ambitious than the usual single speaker pairs used in the literature, the underlying ideas that had been developed to support such transformations were easy to apply.

6.5. Transformation Method

The general technique used was relatively straightforward, and amounted to a non-discrete version of Savic and Nam's [nam90, savic91] codebook based voice transformation network which was described above, with, of course, additional inputs for the speaker model.

A connectionist network was used to perform a non-linear transformation from a combination of a number of LPC frames of input speech from the source speaker and the speaker model representing the target speaker, into a frame of speech in something approximating the target voice. An example of such a network is shown in Figure 38.

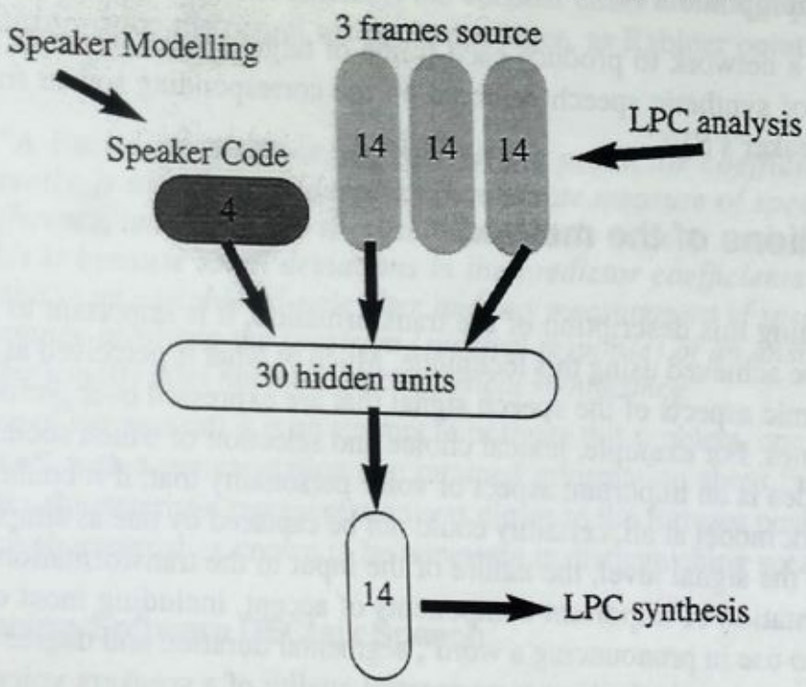


Figure 38: The voice transformation network combines source speech with a target speaker model to produce target speech. One example network is shown, but variations with different numbers of input frames, speaker code widths and number and size of hidden layers were also used.

Obviously, since the speaker modelling experiments were done with TIMIT speakers, it was desirable to use these speakers as the target set of voices for synthesis. To train a voice transformation, though, one needs to have available corresponding speech from two different speakers. Since it was intended that the system should achieve transformations to multiple target speakers, it was necessary to have a single source speaker produce all the training utterances for all the target speakers. The author's voice, since its pronunciation of words differs dramatically from the TIMIT speakers', could not be used. Even if its use had been possible, uttering of the order of a thousand sentences for this purpose would have been rather taxing. Fortunately, just as this section of the work was beginning, Digital Equipment Corp. (DEC) released Software DECtalk [dectalk94] a software-only version of the well known speech synthesis product. Software DECtalk provides a consistent source voice that will say more or less whatever one pleases. Moreover, since the program can be persuaded to output phoneme boundary information as it speaks, the experimenter is freed from the need to label the source speech by hand.

Training a voice transformation was done in four stages, which will be described in more detail in following sections:

- Producing Software DECtalk Speech for utterances corresponding to the training speech available for each TIMIT speaker.
- Time aligning the target speech with the corresponding Software DECtalk speech.
- Locating the target speaker in speaker space by using the speaker models to generate the appropriate SVC.
- Training a network to produce each frame of target speech, given as input both a window of synthetic speech centered on the corresponding source frame and the target speaker's SVC.

6.6. Limitations of the method

Before continuing this description of the transformation, it is important to recognise what can and can't be achieved using this technique. Much of what is perceived as voice quality is based on dynamic aspects of the speech signal that are expressed over periods much longer than a few frames. For example, lexical choice and selection of which social register to use to express an idea is an important aspect of voice personality that, if it could be captured by a neural network model at all, certainly could not be captured by one as simple as those used here. Closer to the signal level, the nature of the input to the transformation used here precludes representation of important components of accent, including most of the choice in which phones to use in pronouncing a word[7], segmental duration and degree of stress. What one can hope to capture is the long term spectral quality of a speakers voice, and, perhaps, specific qualities of spectral transitions that occupy only a few frames.

Although these omissions impose significant constraints on the system's ability to achieve convincing mimicry, and it would certainly be desirable to address them in future work, they are limitations that are shared by all the other voice personality transformation systems

7. Although if a speaker made utterly consistent substitutions, the network might be able to capture some of them.

reviewed. Although one hopes that these longer term effects can eventually be addressed by voice transformation systems, it is currently worth pursuing the goal of obtaining good transformations of the short term components of voice personality.

6.7. Details of the transformation

6.7.1. The speech representation

In all these experiments, the speech was encoded using the standard LPC10 encoder [tremain82], with the quantisation component defeated, and the LPC reflection coefficients converted into the Log Area Ratios described in earlier chapters [rabiner93, Bridle 1994 personal communication][8]. The encoder was also modified to eliminate a three frame delay that had prevented direct use of the TIMIT label files with the LPC10 encoded speech. The speech representation consisted, therefore, of a series of 22.5ms non-overlapping frames each containing fourteen values: two half-frame voicing decisions, pitch, power, and ten LAR LPC coefficients.

No claims are made here that the LPC10 coder is the best, or even a good, choice for doing this work. It was simply available, and easy to modify. Future versions of the system would almost certainly be improved by choosing a better parameterisation of the speech signal. Besides the distortion inherent in a low frame rate coder with pitch and half frame voiced/unvoiced decisions as its only source of voicing information, there is the additional difficulty that the LPC reflection coefficients the vocoder emits are not good candidates for the kind of transformation the neural net performs, since, as Rabiner points out [rabiner93 p191]:

> *"A Euclidean distance, defined on the predictor coefficients directly, is usually considered an inadequate measure of spectral difference, unless the two spectra are extremely close to each other. This is because small deviations in the predictor coefficients can result in an unstable all-pole filter, and any measurement of spectral distance involving the spectrum (spectral response) of an unstable filter usually does not have much physical significance"*

While the use of log area ratios is an attempt to mitigate this problem, one would undoubtedly be better off with a representation that retained information about the glottal source, and whose filter characteristic representation was closer to the formant positions and width, and other voice characteristics known to be important in distinguishing speakers.

6.7.2. Producing Software DECtalk Speech

The TIMIT database used contained both word level and a phoneme level transcriptions of the speech. If a human source speaker had been used, they would have been instructed to produce an utterance matching the word level transcription. With Software DECtalk one can do slightly better than this; it can be instructed to produce a phoneme sequence nearly iden-

8. Log Area ratios can be calculated from LPC reflection coefficients. They are intended to approximate the log ratio of the areas of successive equally spaced approximately cylindrical sections through the vocal tract.

tical to that used by the original speaker. Since the phoneme labels used in the TIMIT transcriptions, and those required by Software DECtalk differ somewhat, a rather *ad hoc* conversion was done, specified in Appendix A.3.

The Software DECtalk software was used to convert each phoneme string into a digital audio file of the speech and to produce a corresponding phonetic transcription[9]. Each speech file was encoded, using the LPC10 encoder, in the same way as the target speech, and the timing information in the transcription was converted so that timings were given in terms of LPC frame indices.

6.7.3. Aligning the speech

In order to learn a mapping between the source and target speakers' speech, it was necessary to find a correspondence between the Software DECtalk speech generated for each utterance and each training speaker, and the natural speech that was the target. It has been noted, above, that doing this well is not a trivial affair when two human speakers are involved, and, unfortunately, it has been shown [hunt84] that aligning MITalk[10] synthesised speech with human speech, approximating what is being done here, is much harder yet. Some effort had to be invested in the alignment process, and even so, the alignments produced were not always as accurate as one might have wished.

The TIMIT speech was aligned with the Software DECtalk speech by subtracting overall means from the frames in each file before alignment and then finding an path minimising the total frame distance $d = (0.1v_1)^2 + (0.1v_2)^2 + (0.005p)^2 + 0.01e^2 + \sum c_i^2$, where v_1, v_2 were the differences between the two frame voicing decisions, p was the pitch difference between the two frames, e was the RMS energy difference, and c_i was the difference between the ith LAR coefficients. An additional constraint was applied to ensure that a single frame of speech took up no more than five frames after warping. The alignment was set up so that the target speech was distorted to match the timing of the source speech, enabling arbitrary source speech to be used during testing.

Initial experiments with this, whole sentence, alignment suggested that the alignment process did not produce very good speech. Plotting just the energy contour for the original, dectalk, and aligned speech made it clear that simply minimising a simple frame distance did not produce the precise alignment needed to learn the best possible mapping between source and target speech. In fact, it did not come close.

The reason for this was not clear, but the choice of spectral representation seemed to be a possibility. Although Rabiner and Juang [rabiner93, p191-2] suggested that the LAR measure used ought to have been a reasonable choice for use with a Euclidean distance metric, it seemed prudent to try using the more popular cepstral distance. Although a definitive set of experiments was not done, comparing the alignments produced by the two measures suggested that, if anything, the LAR measure was performing better[11].

---

9. This transcription did not always correspond exactly to the phoneme sequence that was input. It appears that Software DECtalk contains some obligatory phonological rules.
10. Or, in this case, Software DECtalk.
11. It should be emphasised that much more testing and parameter tweaking was done for the LAR measure, so little should be read into this, except that the LAR measure appeared to be a reasonable one to use.

In order to improve the alignment by using more information, the labelling information available for the two speech samples was used to identify areas of correspondence. A similar alignment procedure to that described above was used, but now the path could be constrained so that frames known to correspond in fact, could be forced to coincide in the alignment. At first, the intention was to align the starts of words within the sentences, but, promises in the manual notwithstanding, Software DECtalk could not be persuaded to output whole word timing information. Instead, it was necessary to use phoneme based constraints. While this produced a more constrained, and, one would hope, more accurate alignment, it was more difficult to achieve, since the phonemes and phoneme labels used by the TIMIT database labelling and those produced by Software DECtalk do not exactly coincide.

For this reason, alignment paths for the LPC coded speech were forced to coincide at a subset of phoneme starts, themselves chosen by a lexical alignment of the phoneme labels for the two strings. The distance between phoneme labels was set to zero if they were identical, to 0.5 if they shared a first letter[12], and to one otherwise. Only exact coincidences (zero distance) were used to constrain the path, and then, only if they were separated by more than one frame. Figure 39 shows the energy contour of speech aligned using this phoneme constrained alignment, compared with the original speech and the Software DECtalk speech for an example sentence from the database.

6.8. Testing the basic voice transformation method

Although the neural net converter chosen was not entirely dissimilar to those reported in the literature, especially in [abe91b, mizuno94 and nam90], it was important to establish single speaker performance as a baseline against which multi-speaker and model-driven voice conversion could be measured. The first goal was to establish whether Software DECtalk speech could be transformed into something more closely resembling the voice of a single human speaker.

6.8.1. Evaluation technique

After the networks had been trained, the quality of the transformation could be evaluated by simply measuring the average frame distortion (d_c)[13] for a DTW alignment between the transformed synthetic speech, and the aligned target speech for the same sentence. The smaller this distortion, the better the transformation achieved. This distortion was evaluated by comparing it with the original average frame distortion (d_o) between the unconverted synthetic speech and the target; any decrease indicated that the transformation had achieved some success. It was also useful to express the results as a percentage distortion decrease $100 \times (d_o - d_c) \div d_o$ relative to the unconverted speech distortion.

To compare the extent to which changes in pitch — the most obvious component of voice quality — as opposed to changes in the spectral shape represented by the ten LAR coefficients in the frames, contributed to the transformation achieved, percentage distortion reduc-

12. This is an odd criterion, but worked well for the TIMIT and dectalk-output phoneme labels.
13. Defined in section 6.7.3 on page 146.

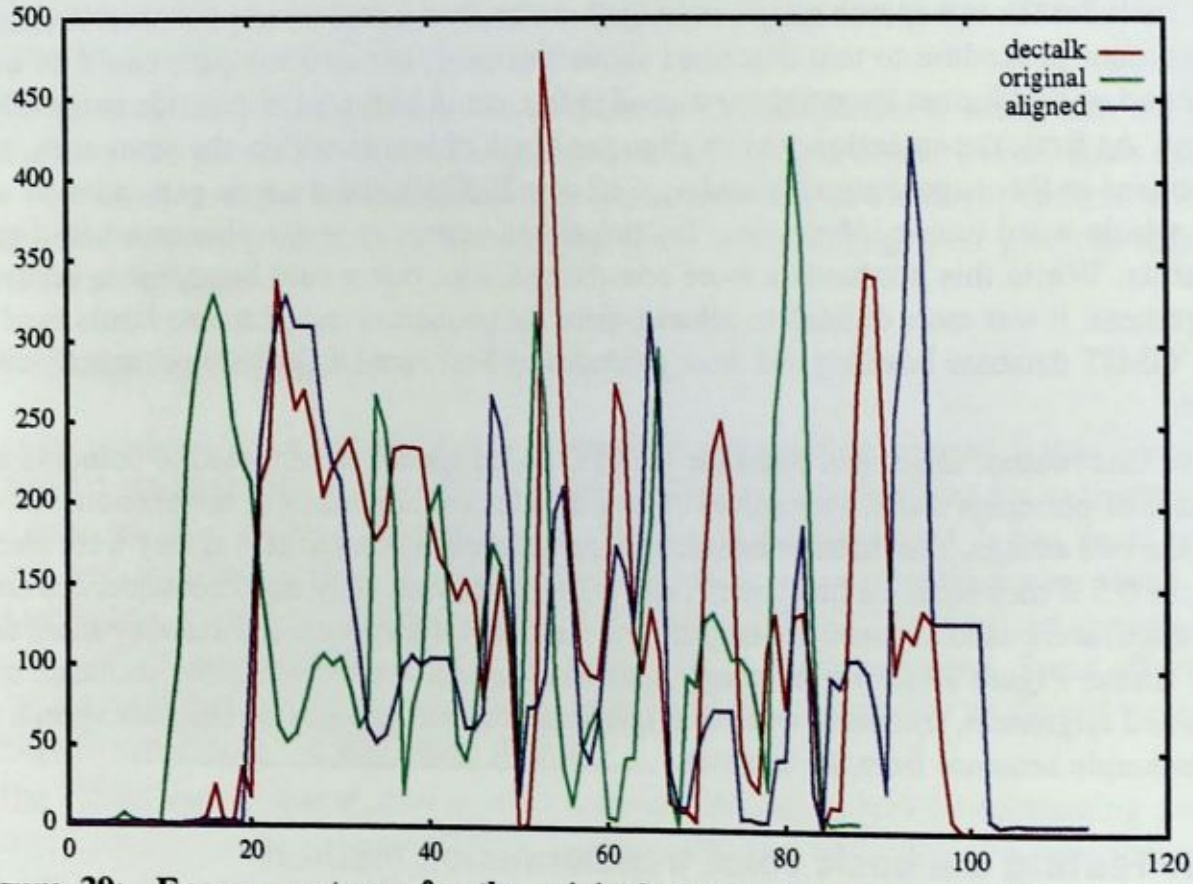


Figure 39: Energy contours for the original speech, the corresponding Software DECtalk speech, and the original speech aligned to the Software DECtalk speech using the techniques described in the text. The speech in question is the TIMIT sentence sx75 — *"The prowler wore a ski mask for disguise."* — spoken by training set speaker dr1-mgr10. Although the alignment improves the correspondence between the energy contours as it is designed to do, it is evident that there is still a considerable difference between the two speech signals and a great deal of opportunity for misalignment.

tions were also measured for alignments using *only* the pitch and spectral components respectively.

6.8.2. Experiment: Transforming Dectalk speech into single Human speaker

Table 41 gives these distortion measures for a variety of neural network architectures applied to the task of converting speech from Software DECtalk into the voice of the TIMIT speaker *fsjw0* from dialect region three. The performance figures are for the TIMIT sentence sa_1, which was not used in training the conversion. Training parameters common to the networks are given in Table B-1 on page 182 of the Appendices.

The average frame distance for an alignment between the target speech and the Software DECtalk input is given at the top of the table for purposes of comparison, together with the distances for an alignment using just the ten LPC-LAR coefficients, and an alignment using just pitch.[14] The remaining rows of the table give the corresponding distances after the Soft-

14. In this case, the distances given are the average difference between the unweighted pitch values from the aligned frames.

ware DECtalk speech had been converted using neural networks with between one and thirteen frames of input, and between thirty and one hundred hidden units, arranged in one or two hidden layers.

The closest match with the target speech was attained by the simplest network, which converted the speech in a single source frame at a time using only thirty hidden units.

Table 41: Effects of voice conversion of Software DECtalk speech with a single human speaker as the target. Distances of converted speech from the target are given for a variety of neural network architectures.

		Distance from Target Speech			% Improvement by Conversion		
		Whole Frames	Spectral	Pitch	Total	Spectral	Pitch
Raw Software DECtalk		3.24	1.44	24.2	0	0	0
Input Frames	Hidden Units	Converted (d_c)	Spectral	Pitch	Total	Spectral	Pitch
1	30	1.99	0.85	8.51	38.6%	41.0%	64.8%
5	30	2.18	1.09	9.03	32.7%	24.3%	62.7%
	60	2.14	1.06	8.98	34.0%	26.4%	62.9%
9	30	2.06	1.09	9.10	36.4%	24.3%	62.4%
13	30	3.39	1.94	8.30	-4.6%	-34.6%	65.7%
	60	2.55	1.12	10.0	21.3%	22.2%	58.8%
	30-30	4.11	2.36	9.95	-26.9%	-63.9%	58.8%
	50-50	3.70	2.18	11.8	-14.2%	-51.4%	51.2%

The networks with the widest input windows actually produced speech that differed from the target speech to a greater degree than the unaltered speech from Software DECtalk. Since there were only a total of nine training utterances containing 1 305 frames of speech available for this speaker, it is reasonable to suspect that the main reason for the superiour performance of the small network on testing data was that the larger networks were overfitting the training data. This hypothesis is supported by the observation that pitch conversion was moderately successful for all the networks, since even a small amount of training

speech ought to be enough to estimate reasonable linear model of the pitch change. Further

Table 42: Training and testing set errors for single speaker conversion networks. These errors are the usual Euclidean distance used with connectionist training.

Input Frames	1	5		9	13			
Hidden units	**30**	**30**	**60**	**30**	**30**	**60**	**30-30**	**50-50**
Mean Train Error	0.506	0.503	0.522	0.389	0.143	0.437	0.059	0.073
Mean Test Error (sa_1)	0.505	0.537	0.527	0.702	1.66	0.71	1.811	1.620

support is provided by the mean output unit errors measured during training, for the networks for training frames and for frames from sa_1, which are given in Table 42. The networks with more than five input frames showed clear signs of overfitting, although this noes not seem to have greatly harmed performance in the case of the network with nine frames of input. The differences between the performance of the single frame network and those with five frames may be due to over-fitting, or may simply be due to chance differences in the nature of the conversion function learned from the training data.

Conclusion

The effects of voice conversion were not just confined to a pitch normalisation, although such a normalisation was performed by the network. In almost all cases, the conversion network moved the spectral representation of the synthetic speech, contained in the ten LAR coefficients, substantially closer to that of the target speech.

Although the speech produced was not of high quality, this experiment had verified that a single neural net, acting as a function approximator, could successfully transform the speech of the Software DECtalk synthesiser into something more closely matching the speech of a target speaker.

6.8.3. Experiment: Plurispeaker synthesis, using perfect speaker information.

If the transformation was to be driven by the speaker model, as intended, the transformation network would have to produce speech from more than one speaker. Following the pattern of the recognition experiments, a *one-from-N* representation was used as an idealised speaker model.

To this end, separate transformation networks were trained for speech from five individual speakers. The task was to convert a single frame of input speech into the corresponding target speech, using the additional information supplied on five speaker ID units. Three varieties of networks were used, the shared parameters for which are given in Table C-2 in Appendix C. The first was a linear network, with no hidden units, that could compute only linear transformations of the input units - this network was capable of only a simple spectral, power, and pitch normalisation, and was included to act as a baseline for comparison of the performance of the non-linear networks with greater powers of functional approximation [hertz91]. The other two networks both had thirty hidden units. In one network, the hidden

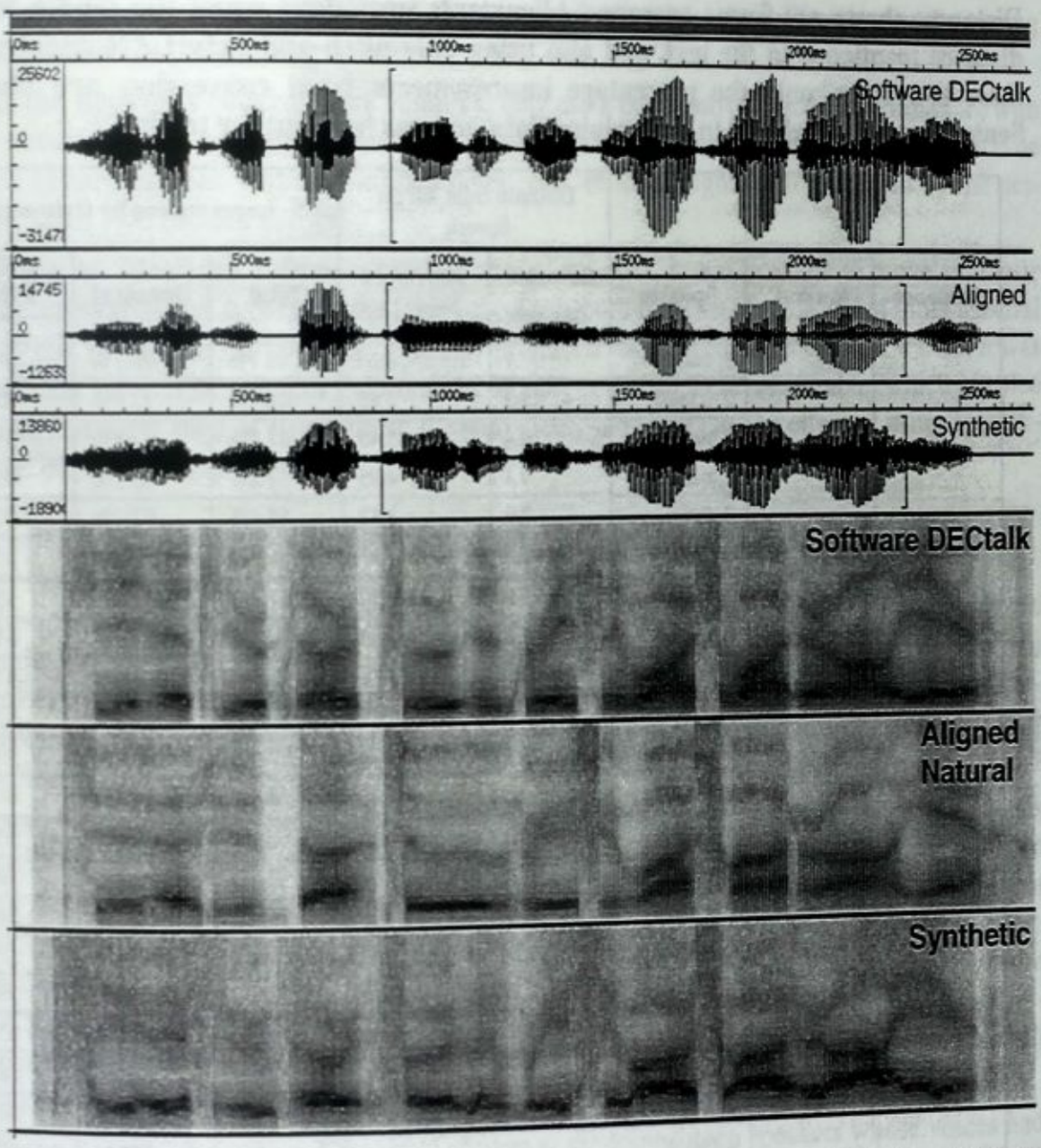


Figure 40: Software DECtalk speech, natural speech aligned to the Software DECtalk speech, and synthetic speech produced by the transform network, both as waveforms and as spectrograms. The transformation network used was one using a *1-from-n* input encoding to select among a set of training speakers. The synthetic (converted) speech seems to share qualities of both the Software DECtalk and target speech, but shows substantially more spectral smear than either. This smearing is a likely contributor to the poor quality of the synthetic speech after transformation.

units were arranged in a single layer, and in the other they were arranged in two layers of fifteen units each.

The following table (Table 43) gives distortion measurements for the three types of networks, both for a sentence included in the training data (sa_2) and one that was held out for testing (sa_1). With the exception only of the linear network applied to test set speech for two speakers and training speech for one speaker, the transformation networks altered the synthetic speech so that it more closely matched speech from the target speaker. As one would

Table 43: Speaker dependent Voice Transformation for five randomly chosen speakers. Distances shown are frame averages. Alignments were done using the whole frame distance mentioned in the text, and also using only pitch and only LAR spectra. In these two cases, only the percentage improvements from conversion are shown. Sentence sa_2 was included in the training data, sa_1 was held out for testing.

			Distance from Target Speech		% Improvement by Conversion		
Sentence	Network	Speaker	Software DECtalk	Converted (d_c)	Total	Spectral	Pitch
sa_1	Linear	3_fntb0	9.11	4.67	48.7%	69.1%	48.2%
		2_fcmm0	12.3	6.83	44.4%	93.9%	48.6%
		2_mmag0	10.4	5.45	47.8%	8.48%	39.6%
		1_mklw0	9.3	4.35	53.2%	15.1%	45.4%
		3_fsjw0	7.9	3.52	55.4%	82.7%	57.1%
		mean	9.81	4.97	61.3%	50.1%	55.5%
	TwoHid	3_fntb0	9.11	4.44	51.3%	67%	53.3%
		2_fcmm0	12.3	4.67	62%	80.9%	61.4%
		2_mmag0	10.4	4.76	54.5%	16.6%	47.1%
		1_mklw0	9.3	3.83	58.8%	22.9%	53.5%
		3_fsjw0	7.9	3.36	57.5%	78.4%	59.5%
		mean	9.81	4.21	70.8%	47.2%	64.4%
	OneHid	3_fntb0	9.11	4.28	53%	65.5%	53.7%
		2_fcmm0	12.3	4.97	59.6%	87.4%	58.5%
		2_mmag0	10.4	4.1	60.7%	2.32%	56.3%
		1_mklw0	9.3	3.8	59.1%	9.54%	53.4%
		3_fsjw0	7.9	3.36	57.5%	77.5%	59.9%
		mean	9.81	4.1	72.2%	46.2%	65.9%
sa_2	Linear	3_fntb0	10.7	3.05	71.6%	89.4%	68.8%
		2_fcmm0	10.6	4.53	57.1%	86.9%	58.9%
		2_mmag0	8.27	3.14	62%	1.23%	57.6%
		1_mklw0	9.1	3.54	61.1%	9.17%	54%
		3_fsjw0	8.28	3.01	63.7%	82%	62.7%
		mean	9.39	3.45	71.7%	51.9%	68%
	TwoHid	3_fntb0	10.7	2.79	74%	77.8%	71.6%
		2_fcmm0	10.6	3.59	66%	83.7%	68.8%
		2_mmag0	8.27	2.85	65.5%	-19%	62.3%
		1_mklw0	9.1	2.89	68.2%	-19.7%	62.5%
		3_fsjw0	8.28	2.79	66.4%	80.7%	65.2%
		mean	9.39	2.98	77.4%	46.1%	74.4%
	OneHid	3_fntb0	10.7	2.75	74.4%	81.7%	72.4%
		2_fcmm0	10.6	3.37	68%	84.6%	70%
		2_mmag0	8.27	2.8	66.2%	-0.537%	62.3%
		1_mklw0	9.1	2.82	69%	-3.33%	62.3%
		3_fsjw0	8.28	3.01	63.7%	77.3%	61.7%
		mean	9.39	2.95	77.8%	48.3%	74.2%

expect, the transformation matched the speech (sa_2) from the training set more closely than held-out speech. There was about a third less reduction in overall distortion for the testing speech than for the training speech.

As the networks became more complex, they did a better job of voice transformation, with the network with a single hidden layer substantially outperforming the linear network. The three layer network with one hidden layer, was, in turn, slightly outperformed by the network with two hidden layers.

Using an idealised *1-from-n* speaker model as input, it was possible to move synthetic speech towards the speech of the target speakers, both in terms of pitch and of other spectral features. While the output speech quality produced was inadequate for practical uses, it was clear that effects of speaker information on output speech were measurable, which enabled taking the next step of using those measurements to investigate the effect of using speaker-space based speaker models.

6.8.4. Experiment: Plurispeaker synthesis, mean speaker models.

Having established that neural networks could use speaker information to move synthetic speech towards that of a target speaker, the next goal was to determine whether positions in speaker space, automatically derived from speech, could be used to train the voice transformation, and whether, after that had been done, the transformation was useful for new speaker targets.

The speaker model used in this experiment was generated by first linearly warping phones to ten fourteen-element frames each, and then projecting these 140-element vectors onto their first eight principal components, computed within each phone. These eight element phone codes were averaged for each speaker, and concatenated into a 488 element vector. The one hundred and eighty-nine such vectors corresponding to the training speakers were then used to train a neural network compressor[15]. This network was used to produce four-element speaker codes for all one hundred and eighty-nine training and sixty-two testing speakers.

Three sets of these speakers were chosen at random: fifteen speakers whose voices had been used to train both the speaker model and the transformation network (*training*), fifteen test speakers who had been used to train the speaker models but who were not used to train the voice transformation (*test*), and fifteen test speakers who had not been used previously (*true test*).

A voice transformation network was trained in a similar manner to that in the preceding experiment except that the four component mean speaker voice code for each speaker was used to replace the binary speaker ID used in the previous experiment, and that three frames of speech, centred around the target frame, were used as input to the transformation.[16] The transformation network had thirty hidden units in a single hidden layer, and additional bypass connections were present directly connecting the input and output layer.

---

15. The network had 488 inputs and outputs, and twenty units in the first, four in the second, and thirty in the third of three hidden layers. No bypass connections were used. Speaker codes were extracted from the four-unit bottleneck layer. The network was trained for 8 000 epochs with a learning rate of 0.001 and momentum of 0.8.

16. Details of the training parameters are given in Table C-3 on page 184 in the Appendices.

Testing the model based transformation

A difficulty with these voice transformation experiments is determining whether one has succeeded in producing the modelled voice. In recognition experiments, the matter is straightforward: if recognition accuracy is better with the speaker information than without it, then one can be said to have succeeded, although the details of how this success was achieved may be difficult to discern. In synthesis, no such simple scoring criterion is available. What must be done is to compare the synthetic speech with actual speech from the target speaker, and see whether it matches that speaker more closely than other speakers. Fortunately, this is possible for the TIMIT database, which provides two particular sentences which are said by every speaker. One of these, sa_1 (*"She had your dark suit in greasy wash-water all year"*) was selected for use in testing.

Within each group of fifteen speakers, synthetic sa_1 speech s_i was produced for each-speaker. Natural speech, N_i was also available for each speaker. An average alignment distance D_{ij} could be computed between the synthetic speech from any speaker i and natural speech for any speaker j by measuring the average distortion of a frame on the best alignment path. These distortion measures form a 15×15 matrix, with the distance between natural and synthetic speech for each speaker lying along the diagonal, and the distances between the synthetic speech for a speaker, and natural speech from other speakers, off the diagonal. If the speaker model is successful, then the diagonal elements should be row minima. To measure whether this was the case, the diagonal elements were subtracted from each row, and the elements of the matrix summed. The result was then normalised by dividing by the number of elements in the matrix, in this case 225.

A positive value of this measure is an indicator of success. Table 44 gives this measure for each of the three speaker groups used in this experiment.

Table 44: The effect of speaker model input on the match between synthetic and natural speech is shown here for training speakers, speakers used to train the speaker model, but not the voice transformation, and completely untrained speakers. The measure, described in the text, compares the match between synthesised speech for a particular speaker and that speaker's natural speech with the match between the synthesised speech and natural speech from other speakers. Larger values represent better conversion.

Speaker Group	Measure
Train	0.397
Test	0.119
True Test	0.105

The speaker model clearly moved the synthesised speech in the direction of the target speaker, on average, with the strongest effect being seen on speakers within the training set. As one would expect, the effect of the speaker model was less pronounced for speakers who

had been used to train the speaker modelling network, but not used to train the voice transformation, and still less pronounced for speakers who had not been used in training at all.

Using a real speaker voice code generated by a network modelling speaker variation, it was possible train a voice transformation to convert synthetic speech into something, resembling, on the average, the target speaker more closely than other speakers, even for speakers who had never been encountered in training. Although a significance test for this effect was not readily available, there was evidence that speaker spaces could be used to model speaker variation in a way that the voice transformation could use, and in a way that generalised to new speakers.

6.9. Voice Transformation using a speaker model for the whole database

Having established that the simplified speaker models of the last two experiments could affect the voice transformation in an appropriate, if insufficiently accurate, way, the final step in the development of the system was to use one of the fully developed speaker models from Chapter 4 to build a transformation network that could cover the complete set of speakers in the TIMIT database.

6.9.1. A neural net speaker model used for voice transformation.

The speaker model used for this purpose was the NNCCR2 (Neural Network Compression with Pattern completion and Recirculation of outputs to inputs) model with fifteen speaker model units as described in Section 4.6.

During training, fifteen-dimensional speaker codes were extracted from this model after all the available speech had been presented to the speaker modelling network[17], and these codes were presented as speaker input to a transformation network with a five frame input window.[18] The training data for the transformation itself consisted of all of the available speech from the training set speakers, except for that occurring in the utterance of sa_1 by each speaker, time aligned with the same utterance as spoken by Software DECtalk. After the network was trained, using the parameters given in Table C-4 of Appendix C., its performance was tested using the same technique outlined in section .

The speaker modelling networks were used to generate speaker codes for each speaker in the training or testing set at the point when two hundred phones had been heard, or after all the speech from the speaker[19] had been exhausted, whichever came first. These speaker codes were used, along with Software DECtalk's utterance of sa_1, to produce synthetic speech in every modelled voice. This modelled speech was compared with the actual utterances of sa_1 by the speakers, after they had been time aligned with the Software DECtalk

17. Actually, in order to reduce the effect of outliers, the average of the last five speaker codes produced was used. This five frame smoothing window was also used on models produced using less input speech whose use will be described in subsequent paragraphs.
18. Using this wider window seemed justified in this case since a great deal more training data — from all one hundred and eighty nine training speakers — was being used.
19. Except, of course, that from the sentence sa_1.

speech. As before, distances were calculated by aligning every "training set" synthetic utterance with every *actual* "training set" utterance. Since there are one hundred and ninety speakers in the training set, this involved more that thirty six thousand alignments.

For training set speakers, on average, the mean frame distance on the best alignment path between the synthetic speech and the real speech it was aimed at mimicking was 0.38 lower than the distance from speech from other speakers. For testing speakers, the effectiveness of the transformation was lower; the distance was 0.26 less, on average, for the target speaker than for other speakers.

The results of the test with a simpler speaker model and fewer speakers were confirmed. The speaker model was allowing the transformation network to move the synthetic speech in the direction of the target speaker's voice.

6.9.2. The time course of transformation quality

To get an idea of how rapidly the useful information became available in the speaker code, the same procedure was repeated, using speaker codes extracted after fewer phones had been heard. The results are summarised in Table 45. For training speakers, the effect of the

Table 45: The influence of amount of speech used to form the speaker code on the quality of the voice transformation achieved. The measure given is explained earlier in the chapter. Larger values suggest better conversion.

Number of phones used to form speaker model	5	15	50	100	200
Train	0.332	0.371	0.380	0.373	0.379
Test	0.228	0.242	0.283	0.270	0.265

speaker model was fairly stable for all models built with more than five phones. For the testing speakers, the fall off in quality was perhaps more gradual, with an apparent decline for the models formed from fifteen phones or fewer.

Although it is fairly clear that five phones were not enough input to allow the speaker model to reach the final speaker code, there was enough variation among codes formed with more speech to prevent any trend beyond that from being evident.

6.10. Validating the final system with human listeners.

Results presented above showed that transformations controlled by a speaker model could reduce the distance between the output of the synthesiser and the speech of the target speaker, when that distance is measured by spectral distortion.

It was also important to know whether this measured distance reduction corresponded to a reduction in perceptual distance. Did the transformed speech from Software DECtalk sound more similar to the voice of the target speaker than the original dectalk speech did? If so, did it sound more like the voice of the particular speaker it is intended to imitate than the voice

of some randomly chosen alternative speaker? The following experiment was intended to provide initial answers to these questions. The aim was to determine whether listeners can identify which of a pair of target speakers, an utterance generated on the output of the transformation is intended to mimic the utterances of a particular target speaker. This experimental design loosely follows the practice of the papers from Abe *et al.* described in this chapter's introduction.

6.11. Experiment: Human ability to discriminate transformed voices.

In this experiment, unaltered Software DECtalk speech was viewed as being at one end of a continuum and the speech from the target speaker was viewed as being at the other end, with stages in the transformation of the former into the latter lying between them. The aim was to measure the degree to which each step in the conversion process affected the perceived voice personality of the speech. These steps are illustrated in Figure 41. Measure-

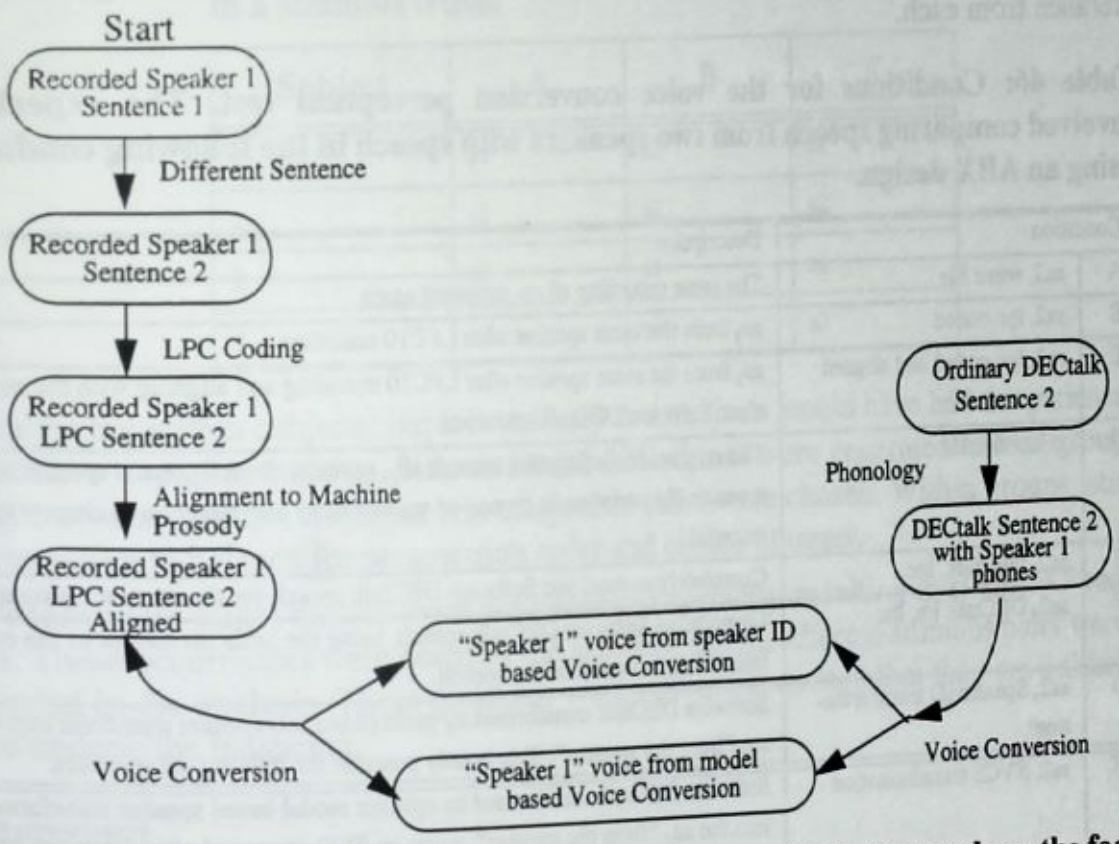


Figure 41: Transformation stages for voice conversion. At the left, we have the factors that increase the distance between the actual target utterance and the output of the voice conversion network. At the right, the factors that increase the distance between the input we would like to have to the voice conversion, and that we do have

ments were made of the ability of listeners to identify which of two speakers, introduced with samples raw speech — labelled "start" in the diagramme — was the target speaker of the transformation. Measurements were made of subjects' ability to identify the target speaker from the speech output from every step in the conversion, in an effort to get an indication where the greatest changes in speaker personality occurred.

6.11.1. Method

Materials

Using the transformation steps shown in Figure 41 as a guide, utterances for a variety of processing conditions, listed in Table 46, were produced for four sets of thirty six target speakers. These sets of thirty-six were composed of nine male and nine female speakers, chosen at random from each of the training and testing speaker sets. Speakers could be reused between sets of thirty six speakers, but not within. Within sets, the training speakers were divided into three groups of three pairs of speakers: three pairs of males, three pairs of females, and three pairs of whom one was male and one was female. Test speakers were similarly divided.

The aim of the experiment was to determine the rates at which speakers in each of the six sets of three pairs could be distinguished from each other, for each of the listed processing conditions, to measure the degree to which voice personality was retained by the conversion process, and, as a base line, the rates at which speakers could be distinguished given a short utterance from each.

Table 46: Conditions for the voice conversion perceptual test. The experiment involved comparing speech from two speakers with speech in the following conditions, using an ABX design.

Condition		Description
A	sa2, wave file	The same recording of sa_2 repeated again.
B	sa2, lpc coded	sa_2 from the same speaker after LPC10 encoding
C	sa2, lpc coded and aligned	sa_2 from the same speaker after LPC10 encoding and aligning with the equivalent Software DECtalk utterance.
D	sa1, wave file	The original recording of a second, sa_1, sentence from the same speaker, to measure the variation in perceived voice quality due merely to a change in material.
E	sa2, DECtalk, lpc	Completely unmodified Software DECtalk speech for the same sa_2 utterance
F	sa2, DECtalk Ph, lpc	Unmodified Software DECtalk speech, using the same phonemes as the original speaker to produce the speech.
G	sa2, SpeakerID transformation[a]	Software DECtalk transformed by multi (fixed set) speaker transform into the sa_2 "from the speaker". This is only possible for training set speakers.
H_1	sa2, SVC5 transformation	Software DECtalk transformed by speaker model based speaker transform into the sa_2 "from the speaker", using an SVC generated after 5 phones had been heard, and smoothed over a 5 phoneme window starting from that point.
H_2	sa2. SVC15 transformation	*ibid*, but after 15 phones heard
H_3	sa2. SVC50 transformation	*ibid*, but after 50phones heard
H_4	sa2. SVC100 transformation	*ibid*, but after 100phones heard
H_5	sa2. SVC200 transformation	*ibid*, but after 200 phones heard
H_n	Novel, SVC200 transformation	and, a completely novel sentence outside the set used in the TIMIT database, transformed using the SVC produced after 200 phones had been heard.

a. Since speaker ID inputs were not, of course, trained for the testing group speakers, this utterances in this condition were only produced for the eighteen training speakers.

Within each of the four speaker sets, two hundred and twenty five ABX stimulus triples were generated, using techniques covered in detail earlier, from the eighteen speaker pairs. In each case the A and B stimulus were down-sampled[20] versions of the original (16 bit, 16kHz) recording of sa_2, *"Don't ask me to carry an oily rag like that"*, from two speakers A and B, and the third sample, X, was generated under one of the conditions of Table 46. Which of the two speakers in a pair were assigned to be A or B, and which of A or B would be used to generate the "matching" stimulus X was chosen randomly for each stimulus. Even the unmodified Software DECtalk speech had one of the target speakers randomly assigned as a "correct" match. To minimise the effect of any bias from the particular choice of X to be deemed correct, and of a listener bias in favour of the second utterance, stimuli were generated for subjects in groups of four, with the only difference being that the presentation order within each triple and which of the reference speakers "A" and "B" would be used to generate the test stimulus, as shown in Table 47. It would be better to do this counterbalancing within subjects, but it is unlikely that subjects would have had the patience for a six hour experimental run. Sixteen sets of 225 stimuli were generated in four groups of four. Between groups, different sets of speaker pairs were chosen. Within groups, stimuli were chosen to balance for presentation order and choice of target.

Table 47: Permuting the triples across subjects. Entries are target speakers used or each of the four subjects in a quad, for each position in a stimulus triple.

Subject	A	B	X
1	s1	s2	s1
2	s1	s2	s2
3	s2	s1	s1
4	s2	s1	s2

Occasionally target stimuli were not generated correctly, resulting in silent "X" recordings. These occurrences were noted by the subjects, and the affected stimulus pairs were not included in the analysis. These problems were infrequent enough that they are unlikely to have undone the balancing.

Equipment

Stimuli were presented monaurally through the left ear using identical headsets plugged into the headset outlets of Digital Equipment Corporation Alpha work stations. Playback was managed using the freely available "Audiofile" audio presentation system [ref]. Raw recordings were down-sampled to 8kHz immediately before playback, and LPC-LAR recordings were decoded immediately before playback by the modified LPC-10 coder used throughout the thesis. Stimuli were presented to subjects, and subjects' judgements

20. To 8kHz so that it could be presented using "Audiofile".

recorded, using the user interface shown in Figure 42. This user interface was written in the

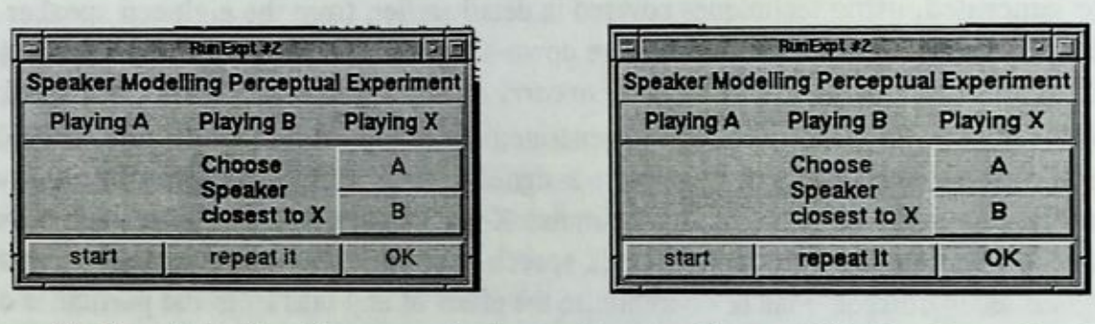


Figure 42: User interface for perceptual experiments, shown while the first stimulus in a triple is playing, and after the speaker has made an initial match X=A.

Tk/Tcl language.

Subjects

Sixteen volunteer subjects in their mid-twenties to early thirties were used. Most were CMU computer science graduate students. Fourteen subjects were male, and two female. No special attempts were made to keep the purpose of the experiment from the subjects.

Procedure

Subjects were seated before one of three alpha work stations displaying the user interface in Figure 42. They were read the introductory passage given in Appendix E., and then asked to start the experiment.

For each stimulus triple, the subject would hit the "Start" button. The three stimuli were played in order, with the user interface indicating which stimulus, A, B, or X was being played by highlighting the appropriate label in green, as shown on the left side of Figure 42. After all three stimuli had played, it became possible to make a choice between A=X or B=X, as shown in the right hand side of the figure. This choice could be changed, or the stimulus triple represented, repeatedly until the "OK" button was presented. None the subjects had any difficulty completing the task.

6.11.2. Results

Table 48: Results of the first perceptual experiment. Each cell contains the percentage of correct identifications of stimulus X as matching the voice in stimulus A or B. These numbers are for three target speakers in each condition for sixteen subjects, giving 48 trials per cell[a]. Means in the bottom row are calculated over the SVC model based conversions indicated by background shading.

Condition		Train			Test		
		Both	Female	Male	Both	Female	Male
A	sa2, wave file	100.0	100.0	97.9	97.9	97.9	97.9
B	sa2, lpc coded	100.0	95.8	89.6	97.9	95.8	93.8
C	sa2, lpc coded,aligned	97.9	89.6	77.1	100.0	85.4	81.3
D	sa1, wave file	97.9	87.5	83.3	97.9	83.3	89.6
E	sa2, DECtalk, lpc	52.1	58.3	43.8	52.1	47.9	37.5
F	sa2, DECtalk Ph, lpc	45.8	60.4	47.8 (46)	60.9 (46)	38.6 (44)	56.3
G	sa2, SpeakerID transformation[b]	91.7	56.3	60.4	-	-	-
H_1	sa2, SVC5 transformation	66.7	47.9	43.8	81.3	41.7	56.3
H_2	sa2. SVC15 transformation	81.3	50.0	52.1	66.7	39.6	58.3
H_3	sa2. SVC50 transformation	79.2	52.1	56.3	77.1	43.8	60.4
H_4	sa2. SVC100 transformation	79.2	47.9	50.0 (46)	81.3	56.3	52.1
H_5	sa2. SVC200 transformation	76.1 (46)	48.9 (47)	64.6	68.8	56.3	56.3
H_n	Novel, SVC200 transformation	77.1	48.9 (47)	54.2	81.3	39.6	52.1
	Mean	76.5	49.4	53.4	75.0	47.5	56.7

a. As noted above, a few of the X targets were not properly produced, reducing the cells marked to the number of trials shown in brackets.

b. Since speaker ID inputs were not, of course, trained for the testing group speakers, this utterances in this condition were only produced for the eighteen training speakers.

Subject's selections of "A" or "B" as the match for "X" were compared with the correct assignment recorded during stimulus preparation, and the proportion of correct selections tabulated in Table 48. It is clear that the task is not an easy one; Even on the task of telling which of the original two utterances had been repeated exactly (row A in the table), subjects did not perform perfectly, and by the time the speech of the target speaker had subjected to lpc encoding and decoding (B) and then aligned with the Software DECtalk speech (C), they were having considerable difficulty telling speakers apart, choosing correctly in about 85% of the cases for pairs of female, and about 80% of the time for pairs of male speakers. The basic encoding used for speech in this transformation was having a substantial effect on the perception of voice personality differences, even when the voice transformation had not been applied.

The difficulty of making voice personality distinctions at all, given a single phrase from each speaker, is illustrated by condition D, where the speakers had to choose which of the two utterances of *"Don't ask me to carry an oily rag like that."* matched the voice that had

uttered *"She had your dark suit in greasy wash water all year."* Subjects made a great many errors telling speakers apart in pairs of matched gender, even though they were listening to clean speech.

It is worth noting that in almost all these conditions with untransformed (but, in some cases, altered) speech, subjects found it easier to distinguish pairs of male test speakers than to distinguish pairs of male training speakers, and that they found it easier to distinguish women than men. While one could easily imagine that there might be gender based differences in the degree of voice personality, the source of the difference between the training and test set remains somewhat mysterious.

Conditions E and F were included as controls. In condition E, raw lpc processed Software DECtalk speech, was presented in the X condition, using phonemes chosen to match those of the target speaker. In condition F, the utterance X was the completely unaltered lpc encoded Software DECtalk speech, regardless of who A and B were. One would expect performance in condition E to be almost completely random, and on F completely random. The fact that some cell values skate dangerously close to significance reminds one of the dangers of reading too much into any individual cell of a large table. Pooled across all conditions in E and F, however, there was no evidence that listeners could guess above chance (P(rate of correct guessing > 0.5, 568 trials) = 0.48).

Conditions $H_{1\text{-}7}$ and H_n were, of course, the focus of the experiment. To deal with the obvious observation first, the speaker model clearly imposed enough voice personality on the transformation to enable men and women to be told apart in many cases — correct identification rates for all the speaker model conditions for pairs of mixed gender were significant at or above the 2% level. What is perhaps more surprising, given the clear separation for gender of the speaker models when plotted in a previous chapter, is that gender could not be separated more reliably. Within pairs of the same gender, the effect of the model is subtle. There was clearly no information retained by the transformation that allowed subjects to tell women apart. For men, though, there was some evidence that the model-based voice transformation was imposing some personality. Although the evidence is not overwhelming for models generated from any of the particular amounts of speech (5, 15, 50, 100 or 200 phones), when results were pooled over all model transforms, within the male speaker pairs, the probability that subjects performing were performing at or below chance was less than 10% for the training speakers, and less than 3% for the test speakers. And, as noted above, the test males seemed to be intrinsically easier to distinguish.

For the sake of completeness note that, as one would expect, since they were played nearer in time, there was a slight but highly significant bias in favour of matching the second (B) utterance of ABX triple with X, (46.3%A 53.7%B). This bias was completely controlled for across speakers by the balanced design of experiment, except for a possible slight effect from the trials with missing X.

It is also worth noting, for the sake of informing the course of future experimentation, that many of the subjects in the experiment mentioned that they were using mainly prosodic cues to match utterances — cues which the current transformation could not possibly capture.

6.11.3. Discussion

It is clear that the speaker model tested included sufficient information about the speaker to specify their gender in many cases. Beyond this the evidence is less clear. It is possible that other salient speaker differences were represented form men, but not for women. Part of the difficulty in determining this is due to the fact that the voice transformation developed here was not as sensitive instrument for measuring the effects of speaker codes as one might have hoped. A discussion of possible reasons for this will be deferred to the end of the chapter.

Since there was clear evidence only that the speaker code allowed gender identification, and weaker evidence that it differentiated usefully between some male speakers, it seemed possible that it was simply affecting pitch. Now, it is true that it is important for such a model to capture pitch, as Valbret *et al* point out [valbret92b]:

> *"The average level of the fundamental frequency is a crucial factor [in voice personality]. Even on nonsense words, the average pitch-value seems to be the most important factor for speaker identification: spectral transformation without the correct pitch modification results in a voice that is not recognised as the target voice; on the other hand, pitch modification without any spectral transformation significantly improves the speaker recognition rate."*

However, one can measure average pitch by less involved means that the models investigated here. It was useful to investigate whether the voice transformations modulated by the speaker code were doing anything beyond affecting the average pitch.

6.12. Experiment: Is the effect of the speaker code accounted for by pitch changes?

6.12.1. Method

To investigate this question, another set of trials closely resembling condition H_4 in the previous experiment was run. The experimental materials were slightly different this time. Utterances A and B were derived from samples of speech from two different speakers uttering the sa_1 sentence. This time, though, this speech was LPC encoded and time aligned with Software DECtalk speech for the same sentence. After alignment, everything but the pitch of utterances A and B was replaced with data from the Software DECtalk speech, so that *only* pitch differences between the speakers could be used to select between them. As in condition H_4 above, the utterance X in the ABX design was Software DECtalk speech for another utterance (sa_2), transformed using the voice transformation system, using the speaker code for speaker A or speaker B. The same speaker model as before was used to generate the speaker code after one hundred phones had been heard from the speaker. If the transformation was affecting only pitch, subjects in this experiment should be able to match the speaker for utterances in A and B with the target speaker for the speaker in X as well as they had in the previous experiment.

There were three experimental conditions: comparison of two men's voices, comparison of two women's voices, and comparison of a woman's voice with a man's. All conditions were presented in both possible orderings of the voices used for A and B, and both choices of whether X would match A or match B.

Stimuli were presented using the same user interface as before, and the experiment was introduced using the same preamble with only the number of trials changed to eighteen. Subjects were twelve male members of the CMU Computer Science Department in their early twenties to thirties.

6.12.2. Results

Results for this experiment are given in Table 49. The first row of the table gives the rate at

Table 49: In this experiment, the effect of the ability of the speaker models to affect voice personality by altering pitch was investigated. The A and B stimuli were Software DECtalk speech with the pitch contour replaced with that from two human speakers, one of whom was the target of the voice transformation that produced stimulus X. Each cell contains the percentage of correct identifications of stimulus X as matching the voice in stimulus A or B. These numbers are for three target speakers for twelve subjects, giving 36 trials per cell.

		Train			Test	
	Both	Female	Male	Both	Female	Male
Percentage correct	77.8	47.2	63.9	69.4	50.0	41.7
Number of standard deviations from chance	3.33	-0.33	1.67	2.33	0.00	-1.00

which the subjects were able correctly to identify which of the speakers A and B the speech in X corresponded to. As in the previous experiment, the speaker models were clearly able to produce an effect on the synthetic voice that allowed subjects to identify the speaker's gender in many cases, with only the pitch of the reference speakers available as a cue to their identity. As one might expect, this effect was greater for the training set than for the test set. For the training set, there was also some evidence that the model was providing information that allowed male speakers to be distinguished on the basis of pitch, but, unlike the similar trend in the previous experiment, this effect did not generalise at all to the test set. It is possible that the model was learning to set parameters for particular male training speakers from the speaker model that made the speech match the targets better, but if it did so, it did not use the speaker models as positions in a speaker space into which test-set male speakers could usefully be placed. Where speaker identification was possible using these stimuli, it was performed at rates that were not dissimilar to those when spectral characteristics of the reference speakers were available for comparison with the output of the speaker model. If more information than pitch was being used in the previous experiment, it was only apparent in the case of male testing set speakers, and that evidence was very weak.

6.12.3. Discussion

As far as it was possible to tell, using the instrument provided by the voice transformation networks, the only perceptually salient information that was consistently encoded by the speaker models was information related to gender, and that information did not have a perceptible effect on anything but the pitch of the talker's voice.

6.13. Experiment: Speaker information apart from pitch?

6.13.1. Procedure

In a final attempt to see whether there were modelled voice qualities apart from pitch that could be used to distinguish speakers, an experiment was run using stimuli in which all differences in pitch had been removed. As before, stimuli were presented in an ABX setup, with each subject listening to nine pairs of women, nine pairs of men and nine pairs of men and women from each of the training and test set, for a total of fifty-four comparisons per subject. The order of presentation of the stimuli, and which corresponded to the X stimuli, were counterbalanced across the four listeners used.

The materials for the A and B stimuli were prepared by taking natural, LPC-coded, speech for sentence sa_1 and time aligning it to the same sentence spoken by Software DECtalk. After alignment, the pitch signal in the speech was entirely replaced by that from the Software DECtalk version, yielding samples all of which had identical pitch contours. The X stimuli was generated as in the previous two experiments, by producing the sa_2 sentence using the voice transformation network with the speaker code the selected speakers out of those who produced sample A and B. After this transformed utterance had been produced, its pitch contour was also replaced with that of the input Software DECtalk speech.

Materials were presented to subjects using the previously described interface, and the subjects were read the usual preamble, with the number of ABX triples replaced by "fifty-four".

6.13.2. Results

Results for this experiment are given in Table 50. The first row of the table gives the rate at

Table 50: In this experiment, the ability of the speaker models to affect voice characteristics other than pitch was investigated. Pitch contours for all three stimuli were replaced with that from Software DECtalk, forcing subjects to use other cues if possible. Each cell contains the percentage of correct identifications of stimulus X as matching the voice in stimulus A or B. These numbers are for nine target speakers for four subjects, giving 36 trials per cell.

		Train			Test	
	Both	Female	Male	Both	Female	Male
Percentage correct	50.0	41.7	63.9	58.3	44.4	44.4
Number of standard deviations from chance	0.00	-1.00	1.67	1.00	-0.67	-0.67

which the subjects were able correctly to identify which of the speakers A and B the speech in X corresponded to. With pitch differences removed, subjects reported that it was very difficult to tell speakers A and B apart, let alone to tell which of them corresponded to the target of X, and this is reflected in the Table. In none of the conditions were listeners able to tell which of speaker A and B was the target in X at rates significantly greater than chance, although in the case of male training set speakers they came close. If any information beyond pitch is contained in the speaker codes, it is either lost during the transformation, or the speakers are unable to use it once the pitch and timing components of voice personality has been destroyed.

There was some weak evidence that some voice personality beyond pitch was retained for male training set speakers, but if it is, future work on improved speaker models, and particularly on improved voice transformation networks will be required to demonstrate the fact conclusively.

6.14. General Conclusions from the Voice Transformation work.

Although the experiments in which spectral distortion for transformed speech were measures indicated that the speaker codes were allowing the transformation to move the Software DECtalk speech towards the target voice, human beings were not able to detect the effects of this movement except in as much as the pitch of the voice was concerned. Although pitch is surely an important component of voice personality it is important to extend the voice codes to include prosodic qualities such are relative segment duration, and to ensure that they accurately represent long term spectral characteristics beyond pitch. It is also important to improve the transformation so that it produces high quality synthetic speech, and so that it accurately expresses the information contained in the speaker code. Since there was considerable discussion of the quality of the speaker codes in earlier chapter, the discussion here concentrates on the voice transformation.

In [abe91a] the authors make an interesting observation that the codebook size of reproducing speech from two speakers accurately must be approximately twice as large as that for as single speaker. While this rate of increase may level off with more speakers, it suggests that the multi-speaker voice transformation problem is, in fact, much more difficult than the same problem with single speakers. Certainly that possibility is supported by the quality of the speech output by the voice transformation system used here, which presented a considerable barrier to its use in evaluating the speaker models used with it in perceptual experiments.

It is difficult to know how far the plurispeaker transformation has to be improved before it matches the quality of other systems in the literature. Savic and Nam said of their voice transformation system that *"Experimental results [not included in their paper] demonstrated that there was almost no difference between the target voice generated by the voice transformation system and the target voice output from the LPC Vector Quantisation Vocoder, which was used as a reference."* On the other hand, In the case of the system in the literature [abe90, abe91a] that had goals most similar to those of the current work, although the training data used was more friendly to alignment, the authors seemed to have reservations about quality:

> *"In terms of the converted voice quality, cross-language voice conversion is not as effective as voice conversion between Japanese speakers. One reason for this may well be that in the cross-language voice conversion experiment MITalk speech was used instead of human speech"* [abe91a].

It would have been very useful, when first building the system, to have used the same kind of neural network and speech representation to build a transformation between two human speakers and to do so using very short utterances as has been done in the literature. Since both the speaker models and the test applications were being developed together, there simply wasn't time to gather the data for this experiment and to train such a transformation. It should however be done. If the transformation in such a system proved to be of low quality, in contrast to those reported in the literature, the cause would plainly be due to the speech representation, or to the use of a neural network functional approximator instead of a codebook based-mapping or connectionist classifier. If the voice quality was high, the system would provide a gold standard from which one could proceed to replacing the source speaker with soft-talk, and thence to the synthesis of multiple target speakers.

Earlier it was pointed out that one of the greatest problems in building the transformation lay in generating a good alignment between the source and target speech. When the alignment is imperfect, the network it trained to transform an input frame into the mean of the target frames to which it is aligned, some of which will be completely inappropriate. The end result of this being "blurred" frames being fed into the resynthesis system, and the production of distorted speech. Obtaining a good alignment was made difficult both by the use of synthetic speech as one of the signals to be aligned, and by the fact that the system depended on aligning entire sentences. It may be possible to improve this alignment by borrowing an idea from the recirculating speaker models. If the transformation moves the source speech towards the target speech, the transformed source speech should be easier to align with the target utterance. By training the system, using this method to obtain a better

alignment, and retraining, iteratively, a sharper transformation should be able to be trained. Another possibility for improving the transformation may be to use a great deal more training data per speaker, obtained from a database other than TIMIT, and to simply discard sections for which poor alignments are obtained from the training data.

A further difficulty with the voice transformation, which is shared by those reviewed in the introduction, is that there are components of voice personality that it can't currently model. Instead of simply transforming the data on a frame by frame basis, future systems should cover all the components of voice personality, beginning with the choice of the correct phonetic realisation for the target speaker of the lexemes in the utterance, and ending with the adjustment of relative durations of the phones within those lexemes, or even of the pitch and loudness profile of the utterance as a whole. Although this is certainly an ambitious goal, it is also a necessary one. In the perceptual experiments here, which compared whole utterances, subjects often commented that they had used prosodic, rather than spectral, qualities of the utterances to match speakers.

Although the quality of the target voice representations and of the transformation used to express them was far from ideal in this initial implementation, it seems likely that both can be improved with further science, to make more of the variation explicit, and further engineering to express it. More will be said on those matters in the next chapter; this one is closed with sentiment the author would like to heartily endorse, with respect to the current work in multi-speaker synthesis with conversion:

> *"Because cross-language voice conversion is a very new idea, and also a very difficult problem, we would like to claim that we have at least shown the possibility of such conversion and demonstrated a possible method"* [abe91a]

Chapter 7. Conclusions and Future Work

Although none of the systems investigated in this thesis was a complete success, a good deal was learned about the speaker modelling enterprise itself, and about the prospects for applying such models in real world tasks. Perhaps the most important lesson was that doing work in this area is currently very challenging. It was necessary to build both the speaker models, and the mechanisms for testing them, and neither of these tasks were straightforward. If there had been a pre-existing speech recognition system or voice transformation system that was known to show clear performance improvements when told which of a large set of speakers it was dealing with, then a larger number of possible speaker models designed to distil that identity into a point in speaker space could have been developed and evaluated, increasing the likelihood of success. If there had been a body of work in developing free-standing models of speaker variation, then there would have been both established criteria for evaluating the current models, and a those models could have been applied to the chosen applications to provide a clear baseline of performance. Instead, the models and the applications had to be pull each other up by their bootstraps, a clumsy and imperfectly executed manoeuvre.

Despite the inadequacies of the models developed here, and despite the difficulty using them with systems that do useful work, the *idea* of developing speaker spaces and using them to help speech systems adjust to new voices seems as promising as it did at the beginning of this work.

There is a great deal of work to be done if this idea is to be realised. The following paragraphs will summarise the conclusions that can be drawn by the work reported here in the areas of speaker modelling generally, and the application areas in speech recognition and synthesis. They will also outline plans for future work that may be useful in approaching the goal of producing systems that can use their knowledge of the way voices vary to improve their performance in the face of the great variety of human voices.

7.1. Speaker models

The speaker models that were built satisfied many of the stated design requirements. They were compact, text independent and formed rapidly. They also captured important characteristics of the speakers, as demonstrated by the fact that speaker gender was visible in the code, and by the fact that they could reduce the distortion between the output of the voice transformation system and speech from the speaker represented by a speaker code.

The failure of the models to be useful in speech recognition was forgivable, since the full-scale recognisers were unable to make use of speaker identity at all. What was more disappointing was the lack of clear evidence from the voice transformation work that the speaker models had captured perceptually relevant variation that could not be accounted for by the pitch of a speaker's voice. Nevertheless, the fact that reasonably good speaker classification accuracies could be attained using the SVCs for nearest centroid classification and the high correlation between speaker models produced from different amounts of speech from the same speaker both support the suspicion that more information about was present in the

speaker models than the transformation application was capable of revealing. Ways of improving on the transformation to produce a more sensitive instrument for making explicit the content of speaker codes will be discussed below, but even if one accepts that the speaker models developed here have made a decent start at a representation of speaker variation, there are clear steps that should be taken to improve them.

7.1.1. Improving segmental models

In the Chapter 3, when the segmental models that combine to make the speaker models were discussed, there was some discussion of methods for normalising the duration of states within speech segments, so that a component of the input to the segmental models would correspond to a spectral channel and a state, rather than a time. Two methods of doing this were discussed: DTW alignment to a set of reference templates, and using states identified by a Markov model based speech recogniser. In the work here, no such normalisation was done — segments were reduced to identical size by linear time warping. While differences in the relative duration of states within phones may well be important to voice personality, it would be probably be better to model this explicitly, by including a vector of relative state duration, along with the set of state spectra, in the input to the segmental models. A comparison should be made between phoneme models produced this way with the current set should be made, to see to what extent explicit modelling of timing variation reduces the intra-speaker stability of the phoneme models.

The other component of the variation in phone models that should be made explicit is the variation due to phonetic context. This context has a strong effect on the way a phoneme is realised, but has nothing to do with speaker variation. Ideally, one would control for this by modelling speaker variation in every context-dependent phone separately, but lack of data and the difficulty of combing the results into and over all speaker model preclude this path. However, preliminary experiments suggested that neural networks could be used to estimate the effect of this variation on a phone within an additive model. This estimate could then be used to control for the context effect when measuring the differences between a phone uttered by different speakers. Phone models that include such a control for context effects ought, again, to improve the stability of phone models within speakers, and would be well worth constructing.

7.1.2. Improving overall speaker models

In general, the linear, statistical speaker models performed as well at forming speaker codes that distinguished speakers as the neural networks did, and the discriminative models, as one might expect, formed more distinctive codes that the "variational" or compression models The sole exception was the recirculating neural network model, which despite being a compression network, produced codes that distinguished speakers well.

If the phoneme codes provide more information about voice characteristics together than they do separately — if they are more than just linear combinations of each other — the neural networks ought to have been able to produce more compact speaker codes than the linear methods. It was certainly demonstrated, on toy problems, that the networks are capable, under ideal conditions, of producing much-better-than-linear encodings.

One reason for this promise not being realised might simply have lain in the fact that the phone models were very noisy and this noise may have masked the interphone correlations the networks needed to observe. If this was the case, simply improving the phoneme models might be sufficient to give a modelling advantage to the neural networks. In any case, to ensure that the neural networks do at least as well as the linear methods, and that training is not expended in learning an imperfect linear model, they should be pre-loaded with weights derived from a linear model before training begins.

Beyond whatever improvements can be made to the raw modelling technology, there is still the matter of perceptual relevance. It was not possible to demonstrate conclusively that perceptually relevant voice properties beyond speaker gender were retained by the current model. It should be possible to improve on this situation. If a very large number of judgements of the degree of similarity of pairs of human voices are gathered from human listeners, the technique of multidimensional scaling can be applied to place these voices within a space in which distances between voices correspond to human perceptual distance. Although the effort involved in collecting the large number of similarity judgements needed would be considerable, the speaker model produced would be valuable; it would provide a standard against which other speaker models could be compared, and the codes representing the position of a speaker in this space could be used as training targets for models, like the present ones, derived automatically from the speech itself.

7.2. Speaker models for speech recognition

Although the application of speaker models to speech recognition here was unsuccessful in almost every respect, this failure cannot be ascribed to the speaker models. In the recognisers that were able to use speaker information at all, namely the recognisers for the Peterson and Barney data, task independent speaker models provided about the same amount of information about sex and age. Of course, one would hope for more than that from a general model of speaker variation. The measures described in the last section intended to decrease the amount of irrelevant variation are likely to improve the quality of these general models, but even of that effort remains relatively unsuccessful, there remains reason for hope. When voice information about a speaker was made available to the Peterson and Barney recogniser, by making the formant values from other phonemes spoken by the same speaker available through a bottleneck, the classifier was able to use this information to greatly increase its recognition performance. In light of the results presented in this thesis, the most likely path to success in applying speaker models to recognition lies in building models that are general, in as much as that they work for new speakers, and do not require retraining, but which are trained in the context of the particular recogniser in which they will be used.

That said, the major problem with applying speaker modelling to speech recognition was that an attempt was made extend to use speaker information in connectionist classifiers to realistic recognition tasks met with very little success. Even recognisers given perfect information about speaker identity benefited little from that information, precluding the possibility of large gains from imperfect information derived from speech. Given that speaker specific recognisers still outperform speaker independent ones, and that adaptation schemes involving additional training of parts of a recogniser with a sample of speech from a new speaker are generally somewhat successful, it is important to explore why speaker ID was

not beneficial. A start was made here: part of the difficulty lies in ensuring that one is providing the recogniser information that it cannot obtain elsewhere. The connectionist recognisers used here, despite their imperfect classification performance, had the advantage of being able to use a wide input window. The experiments with reducing the amount of information visible through this window suggested that the much of the information that could have been derived from speaker ID was already available in the multi-frame input. It also appeared that part of the difficulty was due to the homogeneous nature of connectionist classifiers, in which the use of speaker information to improve vowel classification appeared to interfere with the same classifier's ability to correctly recognise consonants.

If rapid adaptation using speaker models, whether those models are task independent or recogniser specific, is to fulfil its promise, then it will be necessary to gain a far better understanding than we presently have of what recognisers actually do with speech with different speakers, and how this causes errors to be produced. Then it will be clearer what prior information about a speaker's voice could be used to prevent the errors, and when. It may be more productive to pursue such an investigation using a Markov model recogniser, such as Sphinx, where the model parameters have more transparent roles than the weights in a neural net. In Markov models, the distributions associated with inputs to be associated with a particular acoustic state are explicit. If there are many misrecognitions associated with particular examples of these states in speaker recognition mode, then one can compare the distributions for the individual speaker and see whether, for example moving the means of the reference distributions for the state would suffice. If so, and if these mean shifts were correlated across acoustic states for a speaker, one would have a good idea of what sort of speaker model — in this case a regression model between deviations from acoustic state means — is likely to be productive.

7.3. Speaker models for voice transformation

. Using a single connectionist network as a functional approximator, it was possible to transform frames of input speech into something more closely approximating the voice of a target speaker, reducing the pitch and spectral distortion between the synthetic source speech corresponding speech from the target speakers. This was the case both where speaker ID was used to select the target speaker, and when speaker codes derived from one of the speaker models was used. When listening tests were done, however, the only information about speaker identity that was imposed with any reliability on the target speech was speaker gender, and that imperfectly. Despite the fact that the speaker information was affecting other components of the speech signal that pitch, it appeared that audible changes in fundamental frequency of the transformed speech were sufficient to account for the effects of the models in the listening tests. Given that pitch is such an important component of voice personality, this may not be entirely surprising. When samples of natural speech had identical timing[1] and pitch information imposed on them, there was very little perceptible difference between the voices of different speakers, especially speakers of the same sex. If the spectral changes produced by the transformation were related to voice personality,

1. Or as nearly identical as possible, given the difficulty of doing whole-sentence forced alignments, guarantees about timing are difficult to give.

they may have been rendered undetectable by the generally poor quality of the synthetic speech.

Disappointingly, in the light of the claims made for the similar voice transformation systems reviewed from the literature, the synthetic speech produced was, at best, barely intelligible.

Given that the voice transformation did serve to make the effects of speaker codes on the speech produced observable, work to improve the basic transformation system is likely to be rewarding. The main difference between the system used here and those in the literature was the relatively small amount of speech available for a particular target speaker, and the use of synthetic speech as input. It was clear that the alignments produced between the source and target training set were imperfect, even after the work that was done to improve them, and that misalignments in training data are likely to decrease the quality of the transformation that can be learned. Although time consuming, the only obvious way to find out how much an improvement in alignment can improve the voice transformation is to do the alignment by hand, for a larger amount of speech from a single target speaker, matching the quality of training data used in the systems in the literature.

If high quality transformed speech can be produced from synthetic speech by improving the alignment in training, a series of experiments needs to be performed to determine exactly which components of the target speaker it is most important to produce to transmit voice personality. Speech in which only the pitch has been changed, or the relative segmental duration, or the phonetic realisation of the lexemes, or the LPC coefficients, or particular combinations of all of them, should be compared for its ability to transmit voice personality. Only then will it be clear what sort of voice models need to be built to support plurispeaker synthesis well. It may turn out, for example, that relative segmental duration, a feature that was not included at all in the speaker models developed here, is one of the main features of perceived voice personality.

7.4. General Conclusions

Although technological artifacts, in the form of a recogniser with an improved ability to handle speech from a variety of speakers, or a synthesis system producing clear speech in a variety of voices, were not produced in the course of this thesis, the work done here should contribute to their production in the future.

The idea of quantifying the dimensions along which speakers vary is an important one. The models built here captured the variation in some of those dimensions, and pointed to others sources of variation that contaminate the current models, and which themselves need to be modelled.

The work in applications for the models showed that there is a great deal to be learned about the applications themselves. It is not clear, for instance, why recognisers with speaker identity information provided do not perform as well as ones that are trained on specific speakers — what are the parameters set in the latter case that cannot be modulated in the former? What are the components of voice personality that a transformation system *should* learn to affect and that a speaker model should describe? Building explicit models of inter-

speaker variation that address these questions will improve our understanding both of the problems of these speech technologies and of speech itself.

It is hoped that both by improving the task independent models so that they capture more of what is truly distinctive about a speaker's voice, and are less contaminated by what is said and how, and by working in specific domains to discover exactly what *is* distinctive about speaker's voices, the difficult and often frustrating start made here can be turned into a first step leading to the sort of universal models of speaker variation that were hoped for when this work was begun.

References

[abe88] Abe, M., Nakamura, S., Shikano, K. and Kuwabara, H. Voice Conversion through Vector Quantization. *ICASSP 88, Proceedings of the 1985 IEEE International Conference on Acoustics, Speech, and Signal Processing*. pp 655-8.

[abe90] Abe, M., Shikano, K. and Kuwabara, H. Cross Language Voice Conversion., ICASSP 90, *Proceedings of the 1990 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol 1. pp 345-8.

[abe91a] Abe, M., Shikano, K., Statistical analysis of bilingual speaker's speech for cross-language voice conversion. 1990, *Journal of the Acoustical Society of America* Vol 90 No 1, July 1991, pp 76-82.

[abe91b] Abe, M. A Segment-based Approach to Voice Conversion, ICASSP 91, *Proceedings of the 1991 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol 2. pp 765-8.

[abe93] Abe, M., Statistical Analysis of the Acoustic and Prosodic Characteristics of Different Speaking Styles. In *Procedings of the 1993 European Conference on Speech Communication and Technology*, pp 2107-2110, 1993

[artières93] Artières, T., and Gallinari P., Neural Models for Extracting Speaker Characteristics in Speech Modelization Systems", In *Procedings of the 1993 European Conference on Speech Communication and Technology*, pp 2263-2266.

[asoh90] Asoh, H,. and Otsu, N., An Approximation of Nonlinear Discriminant Analysis by Multilayer Neural Networks , 1990, *IJCNN International Joint Conference on Neural Networks*, 1990. Vol 3, pp 211-216.

[assman82] Assmann, P. F., Nearey, T. M., and Hogan, J.T. Vowel Identification: Orthographics, perceptual, and acoustic aspects. *J. Acoust. Soc. Am.* **71(4)** April 1982, pp 975-989.

[ayer93] Ayer, C.M., Hunt, M.J. and Brookes, D.M., A Discriminatively Derived Linear Transform for Improved Speech Recognition. In *Procedings of the 1993 European Conference on Speech Communication and Technology*, pp 583-586, 1993

[becker88] Becker, R. A., Chambers, J. M. and Wilks, A. R. *The new S language*. Wadsworth & Brooks/Cole, Pacific Grove, California. 1988.

[bimbot92] Bimbot, F., Mathan, L., Lima, A., and Chollet, G. Standard and Target Driven AR-vector Models for Speech Analysis and Speaker Recognition. ICASSP 92, *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol 2. pp 5-8.

[blackburn93 Blackburn C., Vonwiller J. & King R. (1993) Automatic Accent Classification using Artificial Neural Networks In *Procedings of the 1993 European Conference on Speech Communication and Technology*, pp 1241-1244.

[blomberg89] Blomberg, Mats, Voice Source Adaptation of Synthetic Phoneme Spectra in Speech Recognition. In *Procedings of the 1989 European Conference on Speech Communication and Technology*,, pp621-624

[boulard88] Bourlard, H., and Kamp, Y. Auto-Association by Multilayer Perceptrons and Singular Value Decomposition. *Biological Cybernetics*, 59, pp 291-294, 1988.

[bridle91] Bridle, J.S. and Cox, S.J. RecNorm: Simultaneous Normalisation and Classification applied to Speech Recognition, in Lippmann, R.P., Moody, J.E. and Touretzky, D.S. Eds, *Neural Information Processing Systems 3*. 1991. Morgan Kauffman, San Mateo.

[chambers93] Chambers, J. M., and Hastie, T.J. *Statistical Models in S*, Chapman & Hall, London. 1993.

[childers85] Childers, D.G., Yegnanarayana, B. and Wu, Ke. 1985, Voice Conversion: Factors Responsible for Quality,. In ICASSP 85, *Proceedings of the 1985 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol 2. pp 748-751.

[childers89] Childers, D.G., Wu, Ke., Hicks, D.M., and Yegnanarayana, B., Voice Conversion, 1989. *Speech Communication* 8 (1989) 147-158.

[conway88] Conway, J.H. and Sloane, N.J.A., *Sphere Packings, Lattices and Groups*, 1988 Springer Verlag, Berlin

[cottrell90] Cottrell, G. W., Extracting features from faces using compression networks: Face, identity, emotion and gender recognition using Holons. In Touretzky, D., Elman, J., Sejnowski, T. and Hinton, G.E. (Eds) *Proceedings of the 1990 Connectionist Models Summer School.* San Mateo: Morgan Kaufman. pp328-337.

[cox89] Cox, S.J., and Bridle, J.S. Unsupervised Speaker Adaptaion by Probabilistic Spectrum Fitting. In *Proceedings of the 1989 IEEE Interna tional Conference on Acoustics, Speech, and Signal Processing.* pp 294-297, April 1989.

[cox90] Cox, S.J., and Bridle, J.S. Simultaneous speaker normalisation and utterance labelling using Bayesian/neural net techniques. 1990. In *Proceedings of the 1990 IEEE Interna tional Conference on Acoustics, Speech, and Signal Processing.* Vol. 1. pp 161-4.

[cox93[Cox, Stephen. Speaker Adaptation Using a Predictive Model. In *Procedings of the 1993 European Conference on Speech Communication and Technology*, pp 2283-2286, 1993

[creelman57] Case of the Unknown Talker. *J. Acoust. Soc. Am*, **29**, 1957 p 655.

[dectalk94] Software DECtalk, Digital Equipment Corporation Maynard, Massachusetts, 1994.

[demers93] DeMers, D., and Cottrellm G. Non-Linear Dimensionality Reduction. In Hanson, S.J., Cowan, J.D. and Giles, C.L. (Eds) NIPS 1993. *Advances in Neural Information Processing Systems.* pp 580-587. San Mateo: Morgan Kaufman. 1993.

[dennis91] Dennis, S and Phillips, S. Analysis tools for neural networks. *Technical report (University of Queensland. Key Centre for Software Technology)* ; no. 207. May 1991.

[duda73] Duda, R.O. and Hart, P.E. *Pattern Recognition and Scene Analysis*, 1973, Wiley, New York.

[eskenazi93] Eskénazi, M. "Trends in Speaking Styles Research", *EUROSPEECH 93, Proceedings of the 3rd European Conference on Speech Communication and Technology.* Berlin. 1993. pp501-509.