

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<sup>18</sup> 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.<sup>19</sup> 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

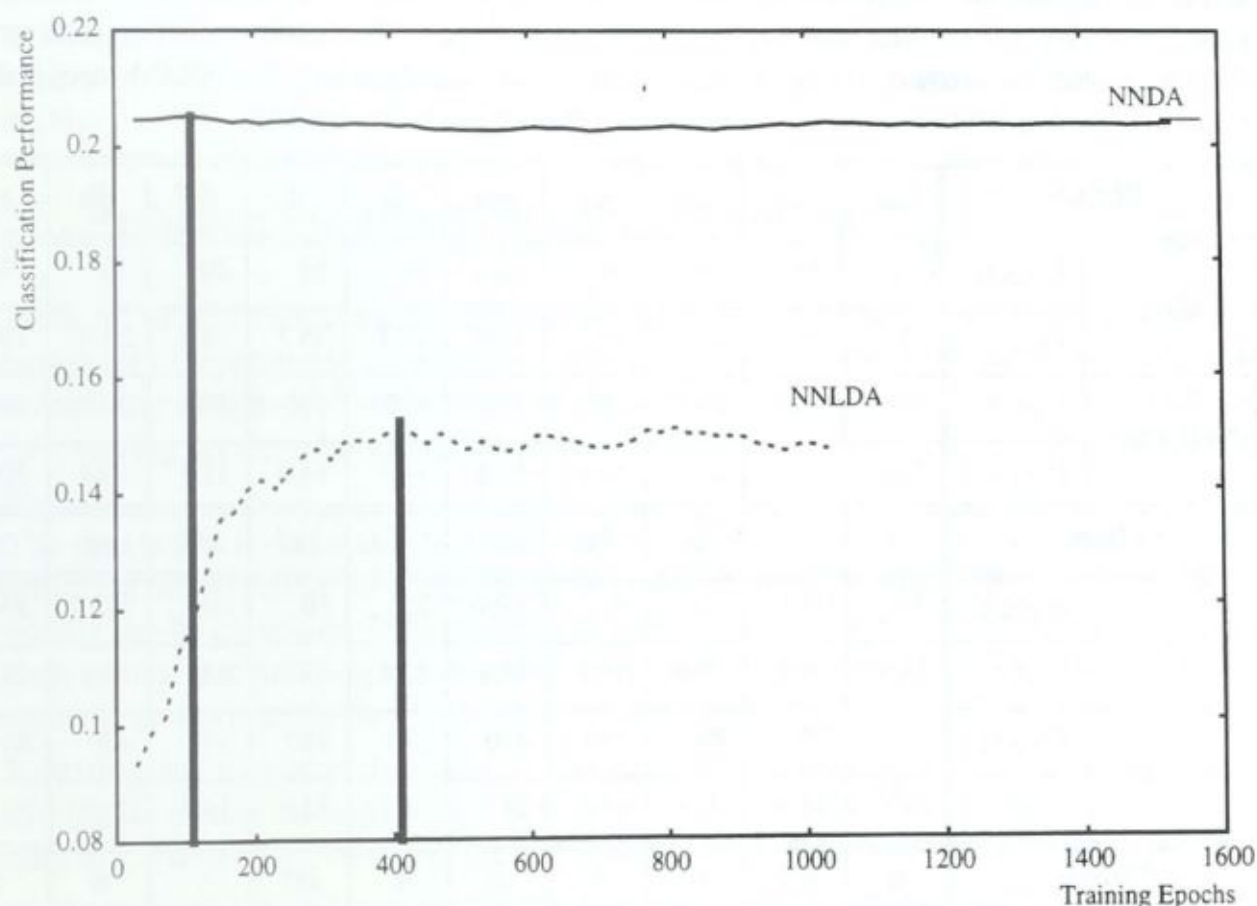
The hypothesis was that the neural network worked best because it failed to learn spurious features of the training set that would interfere with testing set performance. While this hypothesis would seem to have been supported for the J measure, when it came to classification the story was unclear. Although, 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<sup>20</sup>. 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.<sup>a</sup>**

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  $110(\pm 10)$  and  $410(\pm 10)$  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  $\pm 10$ .**

| 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<sup>21</sup> 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<sup>22</sup> 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.<sup>23</sup> 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.**

| $\hat{\mathbf{a}}$   | hidden layer | mean distance <sup>a</sup> from $\hat{\mathbf{a}}$ |      | Test Set Statistics |                      |
|----------------------|--------------|--|------|---------------------|----------------------|
|                      |              | Train  | Test | $t_{2416}$          | p-value <sup>b</sup> |
| 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?

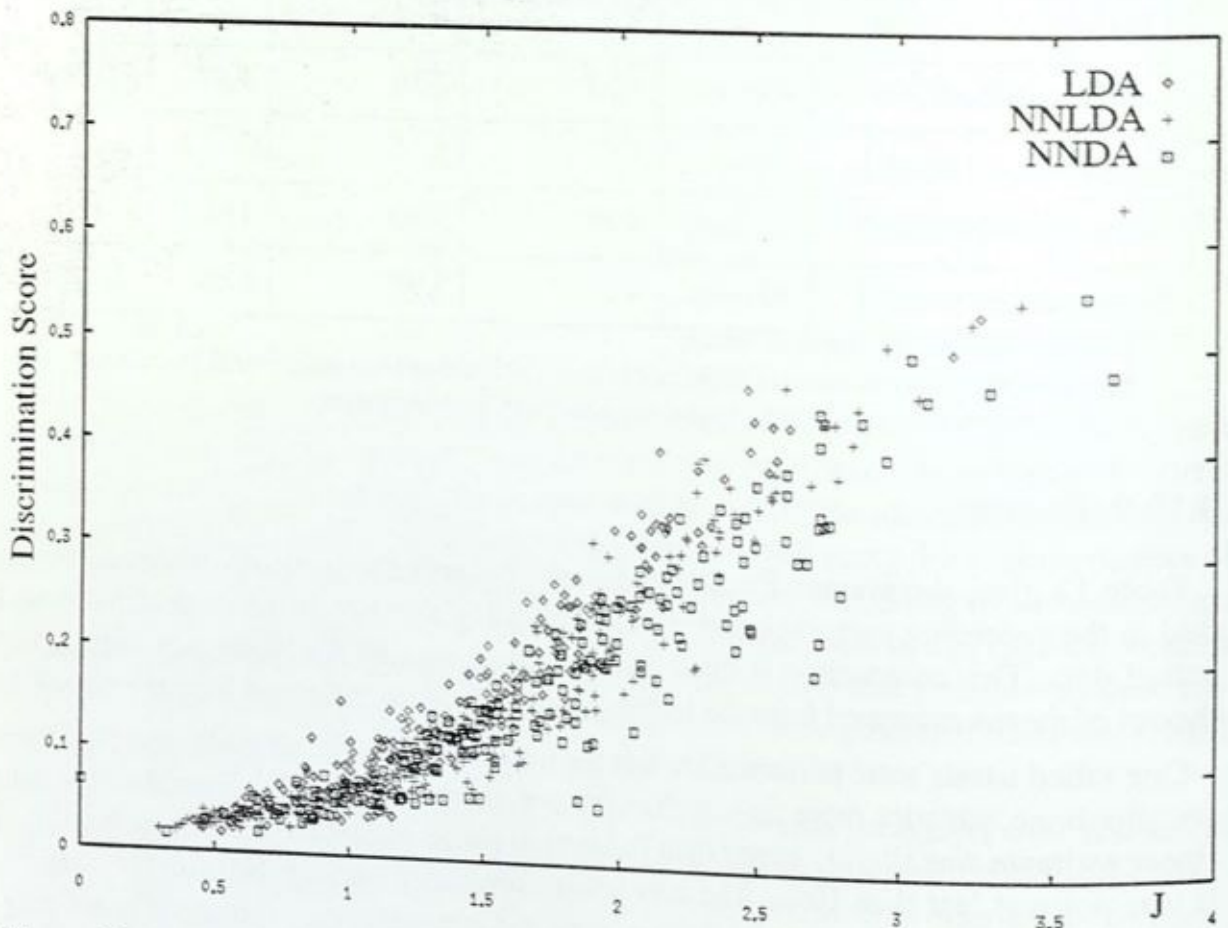


Figure 19: The relationship between the J measure and discrimination performance on the testing set for three phone models. Each point represents a particular combination of phoneme and PPC dimension. Values of J are on the ordinate and discriminant performance, measured as proportion of correct classifications,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

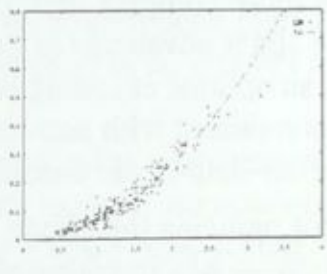
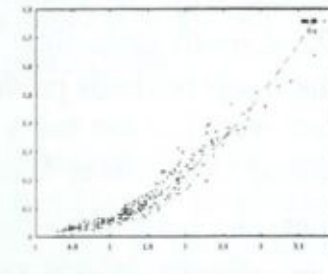
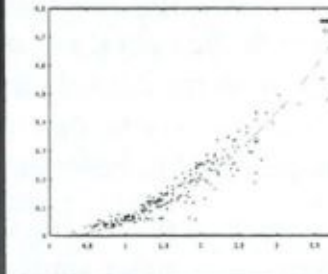
scores were gathered along with the J measure for the PPC candidates in this chapter, it was worth comparing the two measures.

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:

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

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

|          | LDA  | NNLDA   | NNDA   |
|----------|--|---|--|
|          |  |  |  |
| Equation | $disc = 0.066x^2 + \epsilon$   | $disc = 0.055x^2 + \epsilon$  | $disc = 0.048x^2 + \epsilon$   |
| 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,<sup>24</sup> 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<sup>1</sup> 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  $1.4 \times 10^{20}$ .

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

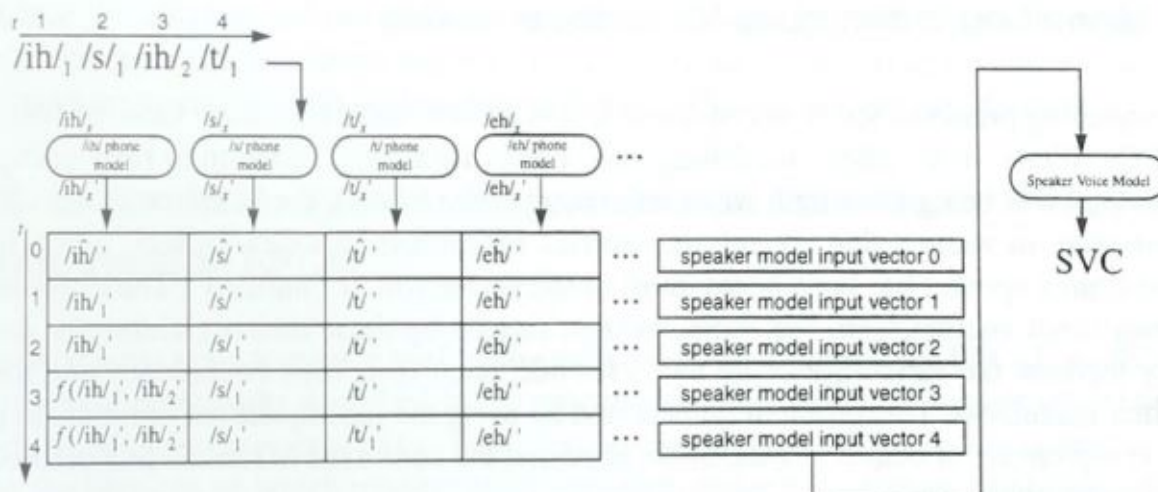
- *Compactness*: To permit the model, and applications that use it, to be used to generalise to new speakers, the space needs to be reasonably densely covered by training speakers. This can only be achieved if the model has low dimension.
- *Text independence*: Human listeners do not need to have the 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.<sup>2</sup> 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 a 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<sup>3</sup> input vector. The vector was reset to zero for each new speaker, equivalent to estimating unheard phone PPCs by their mean<sup>4</sup>. 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<sup>5</sup>.

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

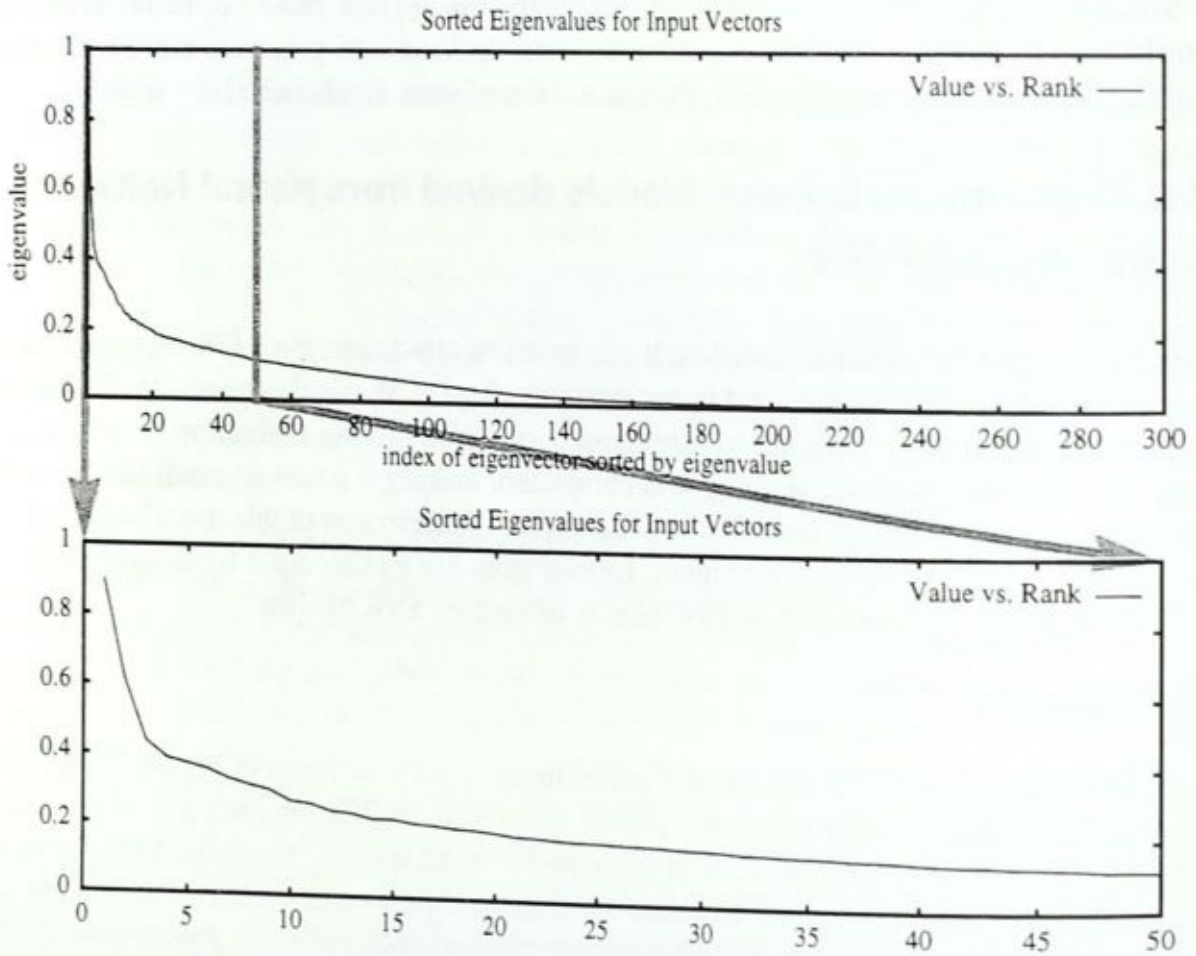
To estimate the dimensionality of the speaker information, sorted eigenvalues for the input vector covariance matrix were examined. These values are plotted in Figure 22. Most of the variation seemed to be contained in the first three dimensions. Although there was no sudden fall off in the eigenvalues, 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.





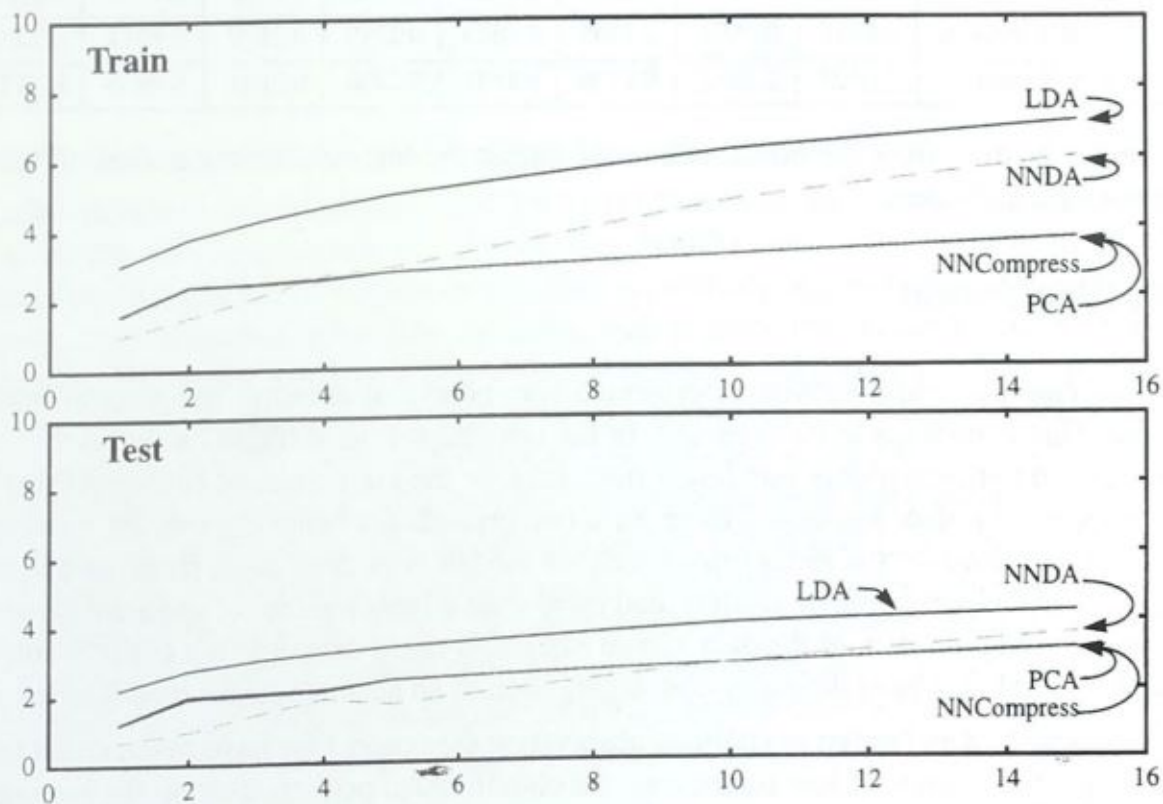
**Figure 22: Eigenvalues the input PPC vectors used to construct the SVC, sorted by size. The lower graph plots the same data as the upper graph, but the domain is limited to the first fifty eigenvalues to show detail.**

many reasonably large eigenvalues (up to about the 150<sup>th</sup> 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 speaker 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.

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

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

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

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<sup>6</sup>. 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  $\phi$  between the centres for even the best possible packing, is proportional to the following expression, where

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

( $o(1)$ ) 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].

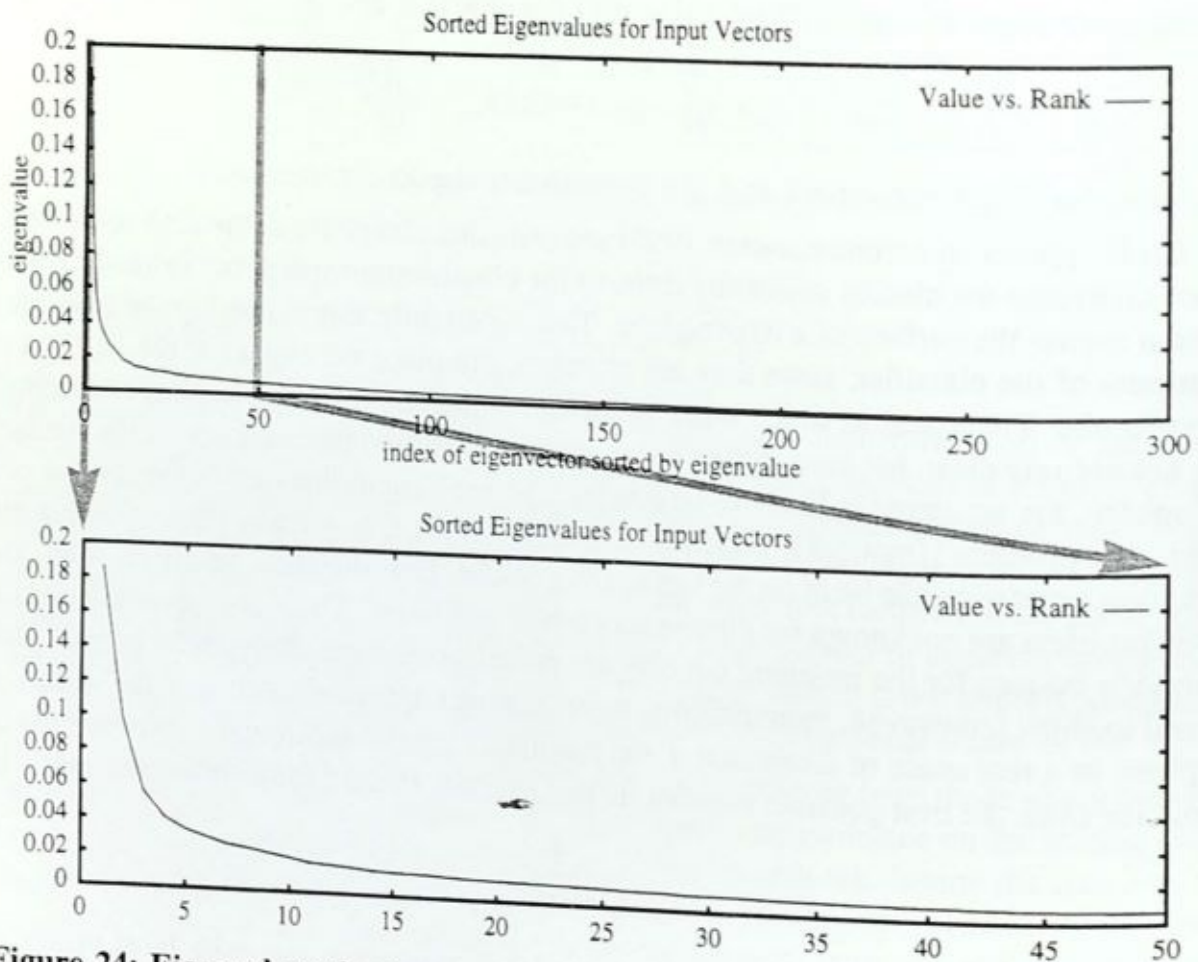
Since this dependence does not vary with  $k$ , we find, as before (working with angles instead of radii) that:

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

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 |

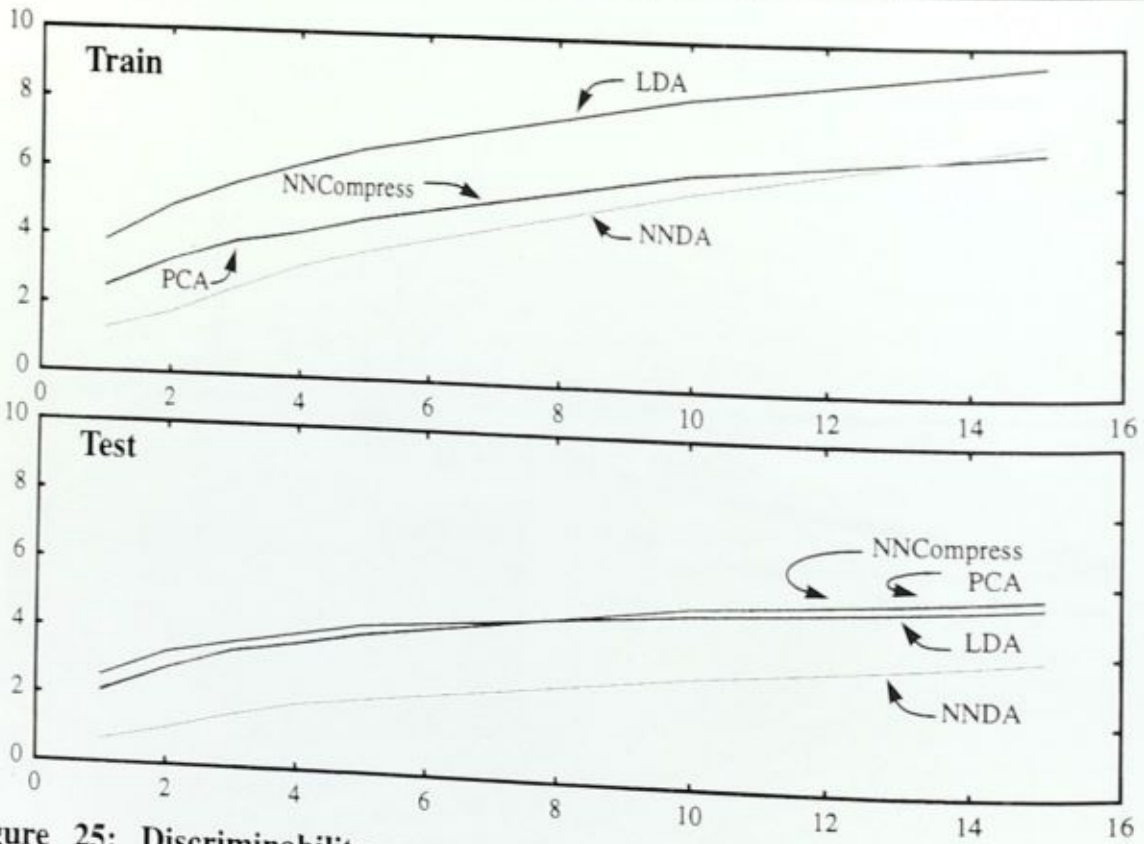


Figure 25: Discriminability measure (J) for speaker models. The PCA and NNComp results coincide, obscuring the plot. NNComp and PCA are nearly identical in both cases.

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 |
| <b>Output layer discrimination scores for neural nets</b> |        |           |        |        |        |        |        |        |        |
| 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 were available<sup>7,8</sup>. 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<sup>9</sup>. 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 NNDA 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<sup>10</sup>. 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.

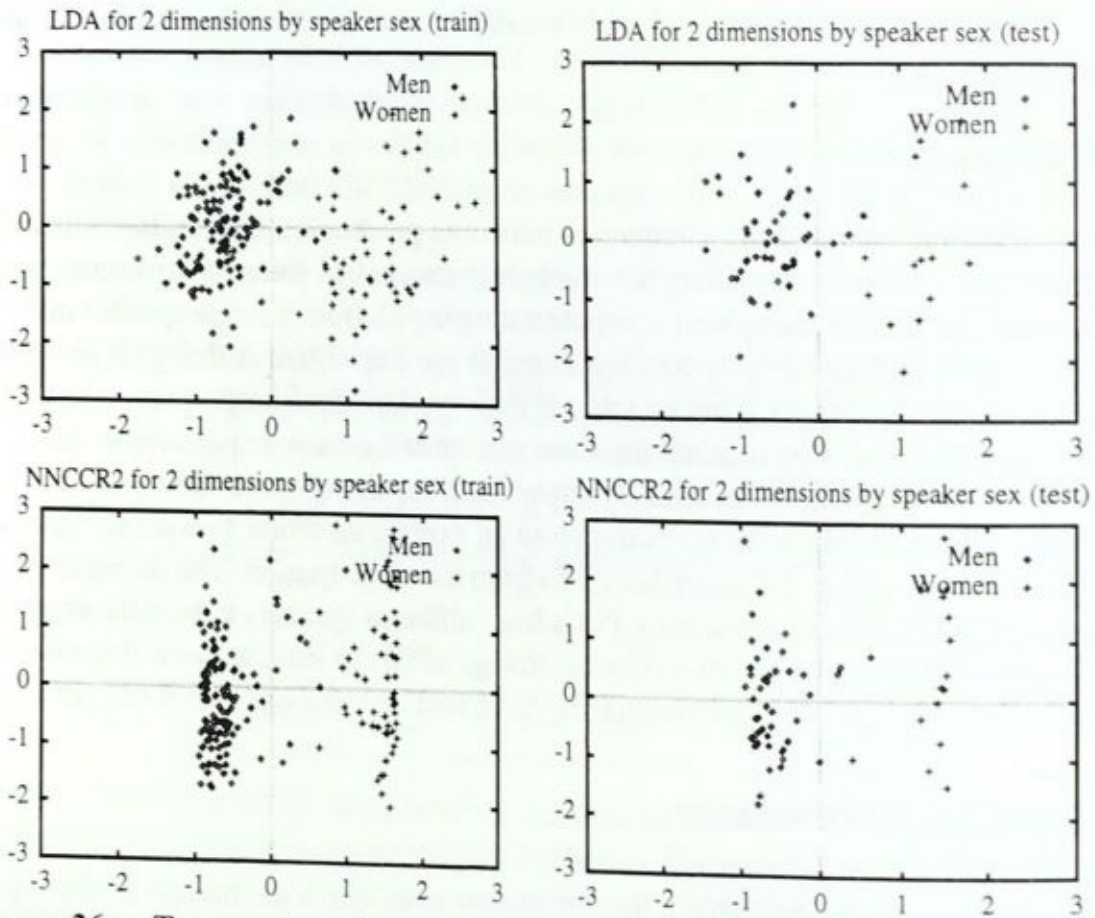


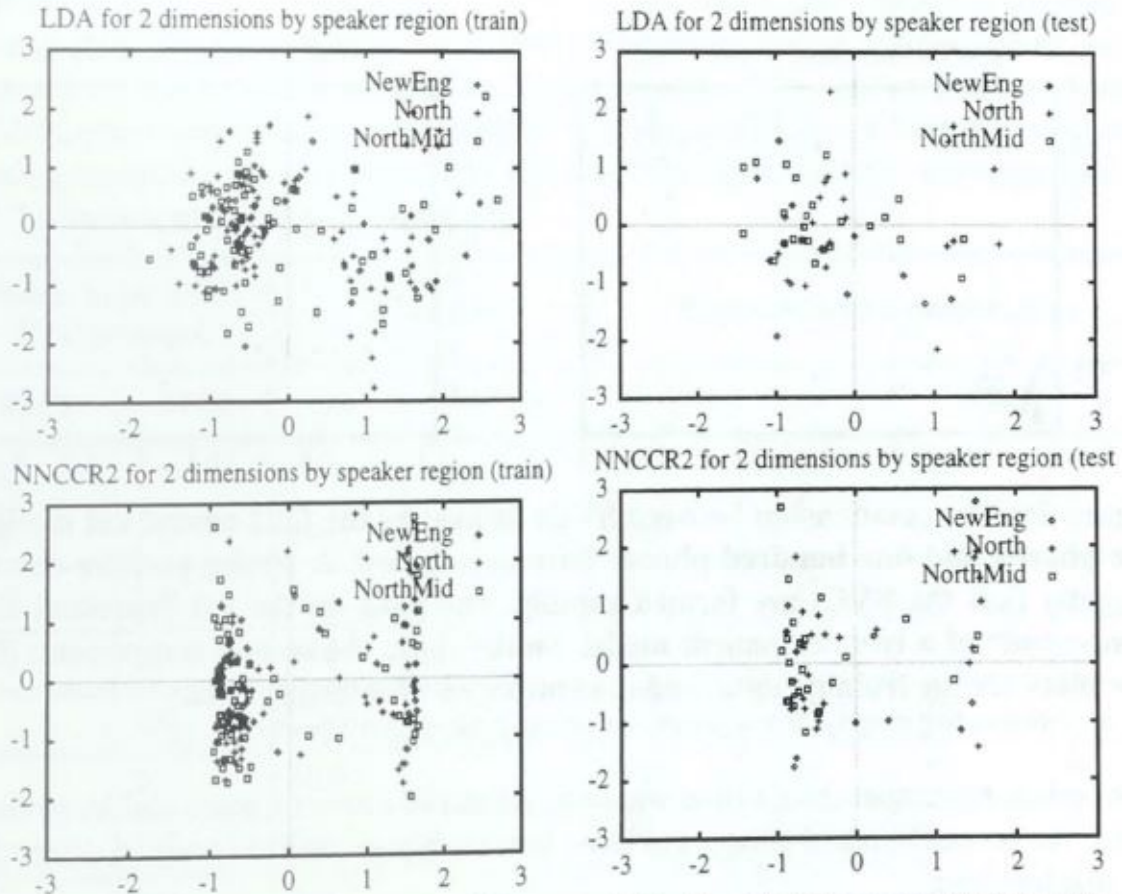
Figure 26: To examine the relationship between the models and perceptually relevant components of voice quality, two dimensional speaker models were plotted here labelled for speaker sex.

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.

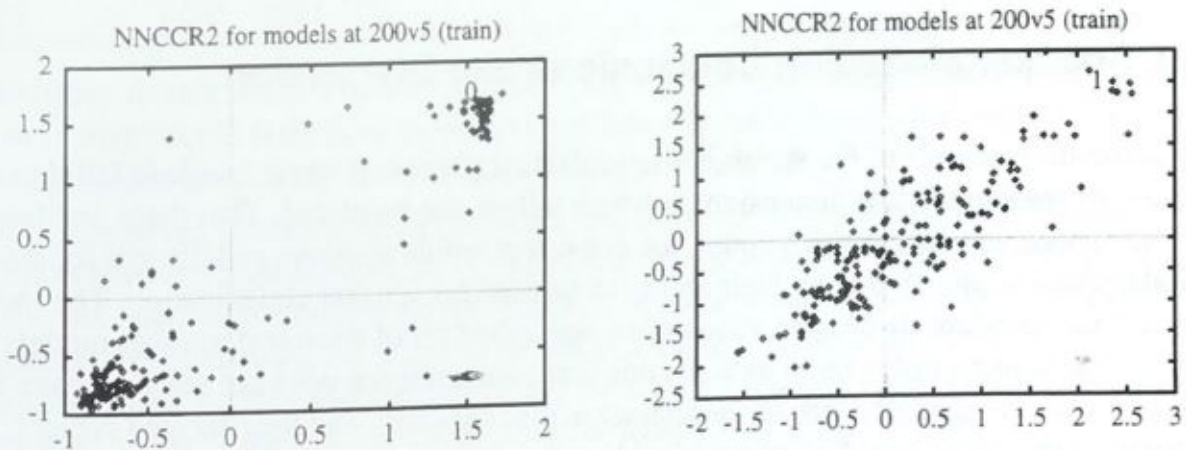
#### 4.8. Rapidity of formation

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.

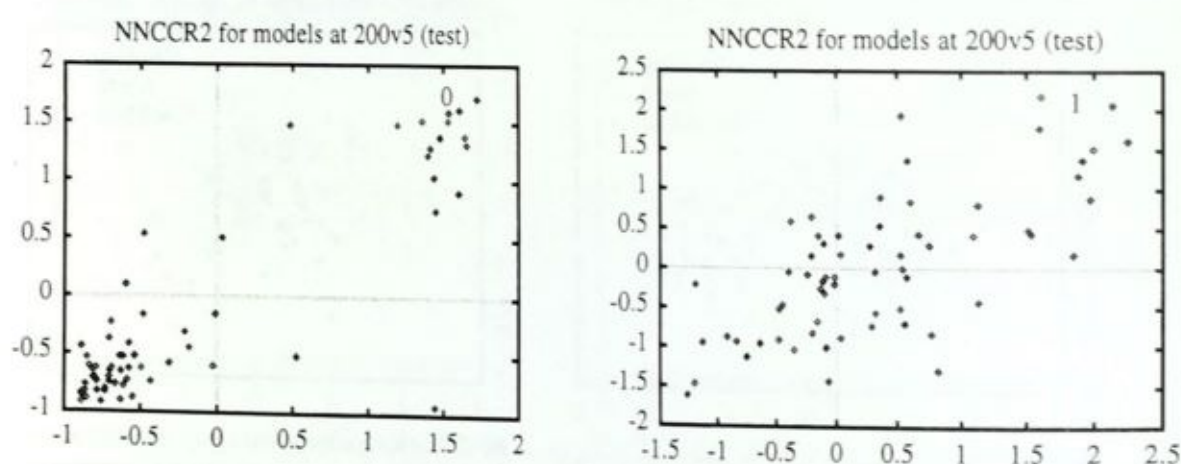
Although there is by no means a total agreement between the models at these times, there was a strong correlation between the SVCs produced at the two times for both SVC components. Table 19 enumerates significance tests for the correlation between SVCs formed after five and two-hundred phones had been 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<sup>a</sup> 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<sup>a</sup> 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-

tivity 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 been 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<sup>1</sup> [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<sup>2</sup>. 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 SVC 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<sup>3</sup>. 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.

<sup>3</sup>. 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- $n$ <sup>4</sup> 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<sup>5</sup> 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%<sup>6</sup> 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<sup>7</sup> 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  $\pm 1\%$  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

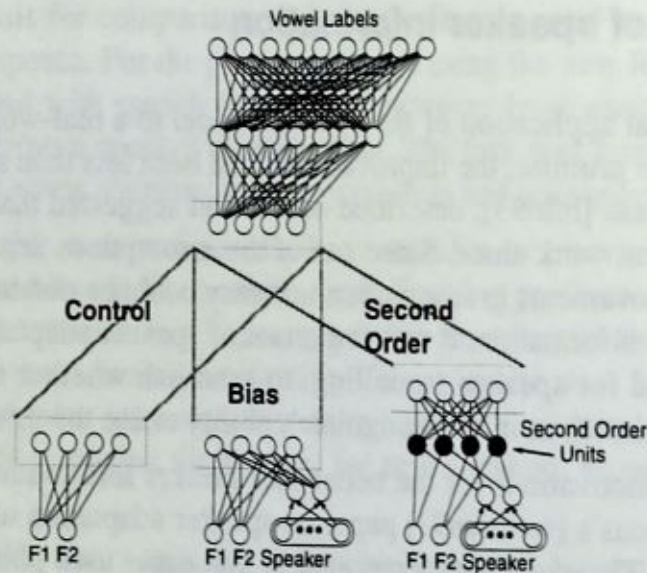
The Petersen and Barney (PB) vowel data [peterson52,watrous91] is a database consisting of formant values for two repetitions each of ten vowels spoken by seventy-six speakers (thirty-three men, twenty-eight women and fifteen children). Because it contains a substantial amount of speaker based variability 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<sup>8</sup> 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% <sup>a</sup>    |
| 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<br>(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<sup>9</sup>. 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.

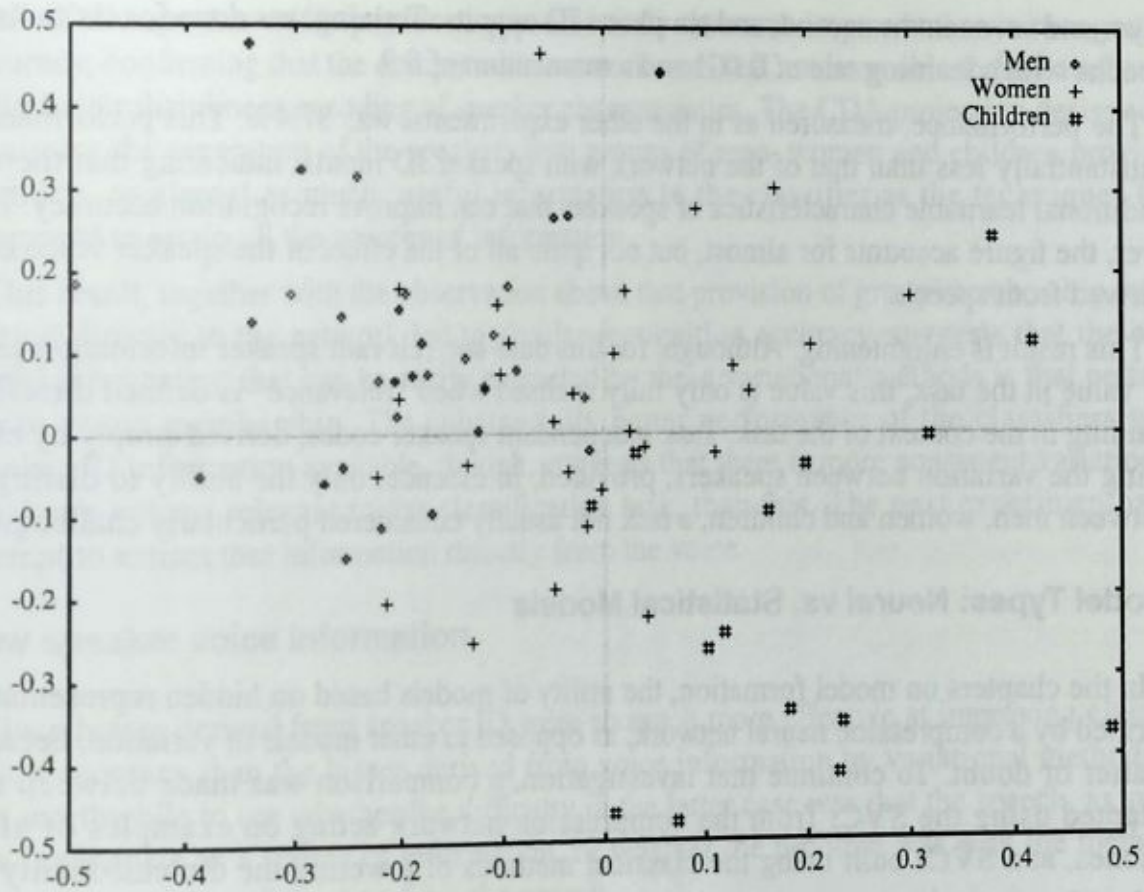
To test the usefulness of these speaker codes as a means for speaker identification, another net was trained to identify the speaker, given the (gender independent) speaker voice code. This was not, apparently, a simple task, since a number of attempts were required before a successful network architecture 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.<sup>10</sup> 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<sup>11</sup>.

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.<sup>12</sup>

| 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 than 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"<sup>13</sup> 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 *ID*s 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 RM Spell 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 <sup>a</sup> | 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 RMspeL 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 RM Spell 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-

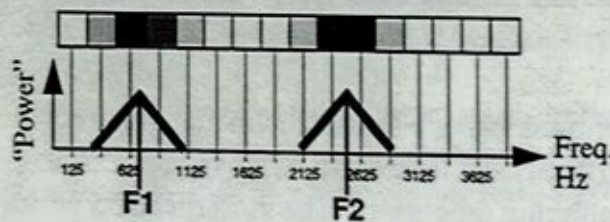


ble sources for the differences between the effect of speaker information on classification of data from the two data-sets included:

- 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,<sup>14</sup> 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 representation as with the raw formants, but the second-order biases were rendered ineffectual.**

Although this representation seems to have prevented the second order networks from making any use at all of the speaker ID information,<sup>15</sup> 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<sup>16</sup>, 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<sup>17</sup> 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.

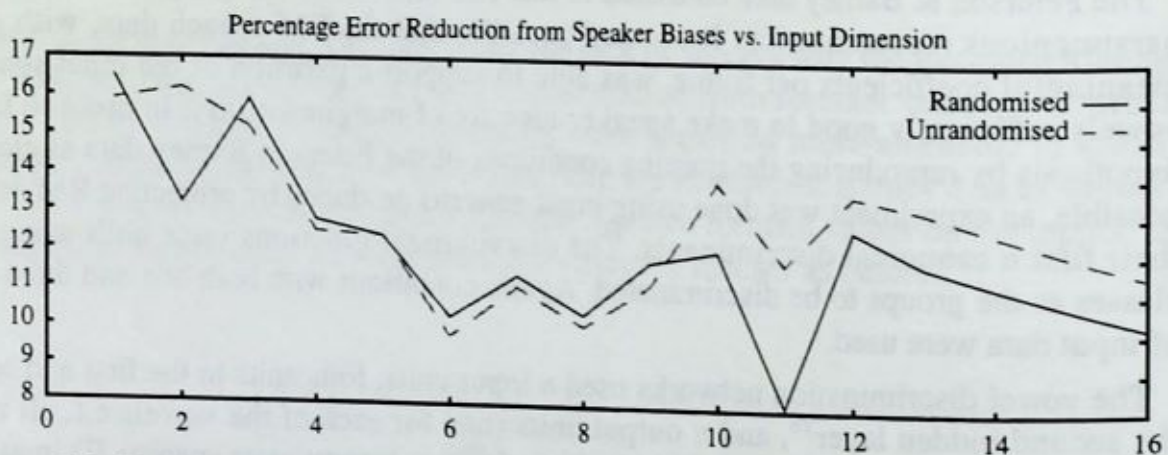
Table 30 gives classification accuracies for both the speaker-biased and unbiased networks for each dimension. The percentage error reduction produced from speaker ID is displayed

| Case <sup>a</sup> | Input Dimension |      |      |      |      |      |      |      |      |      |      |      |      | Raw (16) |
|-------------------|-----------------|------|------|------|------|------|------|------|------|------|------|------|------|----------|
|                   | 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<sup>18</sup> 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.

| Case <sup>a</sup> | Input Dimension |      |      |      |      |      |      |      |      |      |      |      |                   | Raw  |
|-------------------|-----------------|------|------|------|------|------|------|------|------|------|------|------|-------------------|------|
|                   | 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 <sup>b</sup> | 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 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-