CODEBOOKS TO OPTIMISE SPEAKER RECOGNITION

J.S. Mason, J. Oglesby and L. Xu

Department of Electrical and Electronic Engineering
University College, Swansea, UK.

ABSTRACT

A recent approach to speaker identification is based on personalised codebooks. The algorithm compares incoming test features with a set of N codebooks, one for each valid member of the user population, and the codebook which gives rise to the smallest accumulated distance for the full test feature sequence is assumed to identify the speaker.

Results from this inherently text-independent approach have highlighted the performance variations for different test utterances: the spoken digit 'nine' is good, while 'six' is bad. This observation has lead to the idea of classifying speech, via a text and speaker-independent codebook, according to empirical discriminating properties in the recognition task.

Such a classifier is developed here, and experimental results show that 10% or more of speech acts as little more than noise, interfering in the task of speaker recognition.

INTRODUCTION

In recent years interest in speaker recognition has been renewed as a result of the growth in remote access applications. Most of the early work in this area centered on text-dependent dynamic-time warping schemes. Recently however an inherently, text-independent approach using vector quantization and codebooks (VQCB) was proposed by Soong [1] the merits of which are simplicity, both in concept and implementation, Figure 1. Emphasis is placed directly on speaker characteristics and the representative features themselves, rather than their time sequence. Speaker identification (SI) decisions are made on accumulated distances measured from test features to the locally best vector within each individual codebook. Thus the time order of the test feature sequence and its matching with the codebooks is wholly unimportant, eliminating from the measurement any short, time-dependent inconsistencies between the training and the test-utterances. Such inconsistencies are inevitable in other modelling schemes. The approach thus serves as a very good vehicle for feature and distance measure comparisons, benefiting from the absence of these time factor.

Here we propose the use of VQCB's as a speech classifier for use in speaker recognition. The aim is to automatically label those parts of speech with good speaker discriminating properties so that some form of bias might be applied in the subsequent identification or verification process. To illustrate this idea Figure 2 shows some intra- and inter-speaker distributions for individual codebook entries and the corresponding frequency response of these vectors. It is very clear that the separations vary markedly: measurements for single entries from a typical digit-independent codebook give ratios of intrainter mean distances ranging from 0.15 to 0.85. Of course the precise nature of these distributions depends strongly on the the proximity of a given bin with others, the resulting popularity of each bin, the reference features, and the algorithm for the codebook generation. Nonetheless it is still a reasonable hypothesis that a test feature vector that 'hits' a given bin will have discriminating properties related to the distribution of all vectors that 'hit' that bin, examples of which are shown in Figure 2. The greater the overlap of codebook intra- and inter-distributions the more likely a decision error is to occur.

Figure 1. A VQ-based speaker identification system.
Figure 2. The frequency response and intra- and inter-speaker distributions for a voiced sound, vector A, and an unvoiced sound, vector B.

EXPERIMENTS

A series of experiments is aimed at highlighting those parts (or classes) of speech which, on average, give good speaker discrimination. The basic SI codebook scheme is that shown in Figure 1, but added to this is a classifier through which incoming test features are passed, Figure 3. Only if a feature belongs to a pre-chosen class, dictated by one of the entries in the classifier codebook, does this given feature pass below to the SI codebooks, otherwise it is discarded. A standard codebook generation algorithm is used for the classifier codebook (CCB). Single utterances of the digit set from each of the 10 speakers known to the system is used to train the CCB, 100 utterances in all. For the experiments describe below the SI codebooks were trained with 5 utterances of the digit set from the relevant speakers and recognition results are from 5 different utterances of the digit set, 500 test utterances with almost 2000 feature vectors producing one experimental result.

Obviously with a CCB of size 1 all test features pass through, and these results are used as the standard or control, Table 1a. With a CCB of size 2 approximately one half of the speech would be discarded and the other half passed through for testing. First entry number one in the CCB is used, then the experiment is re-run using entry number 2; this interchanges the 2 sets of speech features used for the recognition: the set discarded in the first run is now used for testing and vice versa. The effects of using these two different groups of testing data are illustrated by measures from the SI process, namely the distance to the nearest neighbour and the average distance to all the other persons codebooks. The measurements are made at the end of each utterance, and then averaged over all 500 utterances, Table 1b. The experiment is repeated for larger sizes of classifying book. The associated spectra for each entry of CCB size 4 are shown in Figure 4. Table 1c gives the results for this CCB, and it is clear that there are marked differences in the discriminating properties of the 4 classes. Next the size of the CCB is increased to 32, corresponding to the model size found sufficient to characterise speakers in the standard VQCB scheme. Results of the 32 associated trials are shown in Table 1d. The spectra of the size 32 CCB is shown in Figure 5 given in order of the nearest neighbour mean distance, and as expected the end with the better discriminating classes characterised by low-frequency dominant entries associated with vowel sounds; the other end contains more high-frequency dominant entries associated with unvoiced sounds.
In the above experiments just one class or CCB entry is used to gain statistics for that given class. Now the number of classes used is increased, one-by-one, according to the nearest neighbour merit order given in Table 4, so that the apparent best discriminating classes are combined first, entries 27, 31, 15, etc., through to CCB entry 5 which has a nearest neighbour distance of -0.74. This is an intuitive combination order, and by no means is suggested as the optimum. However the corresponding performance for these combined classes given in Figure 6 serves to illustrate the principle that certain types of feature vectors do not assist the task of speaker recognition.

Figure 5. The frequency response of the entries in the size 32 classifier codebook in nearest neighbour merit order.

Figure 6. The nearest neighbour distance as increasing numbers of classes of speech are let through the classifier. The distance is first normalised by the number of test vectors in a given utterance passed by the CCB, then averaged over all utterances of all speakers.

Figure 7. The speaker identification error rate as increasing numbers of classes of speech are let through the classifier.
Using the same combined classes as used for Figure 6 to filter the incoming speech, we measure the SI recognition rates Figure 7. The result showing that speech corresponding to entries 12 and 24 actually hinder the SI process reducing the recognition performance, the optimum being when 30 of the 32 classes of speech are used.

**CONCLUSIONS**

It is demonstrated that different classes of speech make differing contributions in the task of speaker recognition. Experimental results show that actually discarding some parts of speech can improve recognition rates. In total 500 test utterances corresponding to almost 20,000 feature vectors were used in the experiments. When discarding two chosen speech classes, accounting for some 1600 of the vectors, recognition performance improved. Little improvement is seen when more than 80% of the speech is used implying some 20% of the speech vectors have little or no speaker specific information. This process has not yet been optimised, and it is likely that further judicious discarding would improve the results still further.

**REFERENCES**


