RAPID SPEAKER ID USING DISCRETE MMI FEATURE QUANTISATION

Dr. Jonathan T. Foote
Cambridge University Engineering Department
Trumpington Street, Cambridge, CB2 1PZ, United Kingdom
Email: jtf@eng.cam.ac.uk Phone: +44 1223 332800 Fax: +44 1223 332662

ABSTRACT

This paper presents a method of rapidly determining speaker identity from a small sample of speech, using a tree-based vector quantiser trained to maximise mutual information (MMI). The method is text-independent and new speakers may be rapidly enrolled. Unlike conventional hidden Markov model approaches, this method is computationally inexpensive, yet is robust even with only a small amount of test data. Thus speaker identification is rapid in terms of both computational cost and the small amount of test speech necessary to identify the speaker. This paper presents theoretical and experimental results, indicating that perfect ID accuracy may be achieved on a 15-speaker corpus using little more than one second of text-independent test speech.

1. INTRODUCTION

Automatically determining the identity of a speaker is an important application of speech recognition technology. Besides the obvious security and verification applications, speaker identification (ID) technology can improve speech recognition accuracy by selecting speaker-dependent models [1] and can also be used to segment audio and video for multimedia applications [2].

Most recent speaker identification work has centered on continuous-density hidden Markov models. This paper presents an alternative discrete method, which, because it is trained discriminatively, can capture the differences between talkers without the need for the Viterbi decoding stage of HMM methods.

Unlike K-means vector quantisation (VQ), the tree-based quantisation is supervised, which means the feature space may be profitably discretized into many more regions than the conventional minimum-distortion vector quantisers. In addition, the tree-based method is arguably more robust in high-dimensional feature space, and may be pruned to vary the number of free parameters to better reflect the amount of available enrolment data. Perhaps more importantly, MMI-constructed trees can arguably handle the "curse of dimensionality" better than a minimum-distortion VQ, in part because only one dimension is considered at each split. Dimensions that do not help class discrimination are ignored, in contrast to a distortion metric which is always computed across all dimensions.

In practice, the speaker identification system works as follows. Both test and enrolment speech is first parameterised into mel-scaled cepstral coefficients (MFCCs) plus an energy term. The speech waveform, sampled at 16 kHz, is thus transformed into a 13-dimensional feature vector (12 MFCC coefficients plus energy) at a 100-Hz frame rate. This parameterisation has been shown to be quite effective for speech recognition and speaker ID, even though some speaker-dependent characteristics (such as pitch) are discarded.

A quantisation tree is grown off-line, using training data from as many speakers as practicable. Such a tree is essentially a vector quantiser; discriminative training ensures that it attempts to label feature vectors from different speakers with a different label. To enrol a reference speaker for subsequent identification, enrolment data is quantised, and a probability density function (pdf) is estimated by counting the relative frequencies of each label. This pdf serves as a reference template with which unknown speakers may be compared.

To identify an unknown speaker, a pdf is computed from quantised test data in a similar manner. This test template can be compared with those from the reference speaker using one of any number of distance measures; the "closest" reference template then identifies the unknown speaker. For speaker verification tasks, (i.e. the speaker may not be in the training set), a distance threshold may be set to reject speakers that do not sufficiently resemble any reference model.

1.1. Supervised MMI Trees for Quantisation

The feature space is partitioned into a number of discrete regions (analogous to the Voronoi polygons surrounding VQ reference vectors) by a decision tree. Unlike K-means reference vector estimation, the tree is grown in a supervised fashion. Each decision in the tree involves comparing one element of the vector with a fixed threshold, and going to the left or right child depending on whether the value is greater or lesser. Each threshold is chosen to maximise the mutual information $I(X;C)$ between the data X and the associated class labels C that indicate the speaker that generated each datum.

1.2. Tree Construction

Because the construction of optimal decision trees is NP-hard, they are typically grown using a greedy strategy [3]. The first step of the greedy algorithm is to find the decision hyperplane that maximises the mutual information metric. While other researchers have searched for the best general hyperplane using a gradient-ascent search [4], the approach taken here is to consider only hyperplanes normal to the feature axes, and to find the maximum mutual information (MMI) hyperplane from the optimal one-dimensional split. This is computationally reasonable, easily optimised, and has the advantage that the search cost increases only linearly with dimension.

To build a tree, the best MMI split for all the training data is found by considering all possible thresholds in all possible dimensions. The MMI split threshold is a hyperplane parallel to all feature axes except dimension d, which it intercepts at value t. This hyperplane divides the set of N training vectors X into two sets $X = \{Xa, Xb\}$, such that

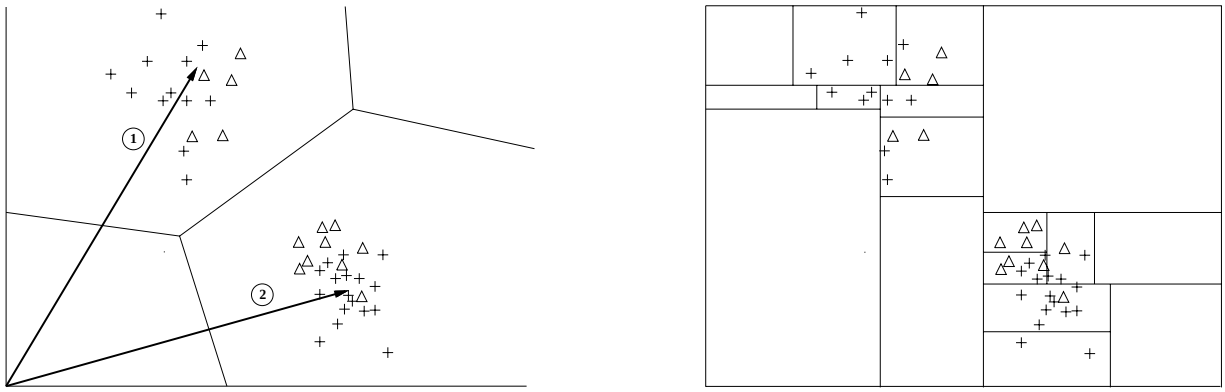

Figure 1. VQ (left) and MMI tree (right) feature space partitions.

$$Xa: \quad x_d \geq t_d \tag{1}$$

$$Xb: \quad x_d < t_d \tag{2}$$

This first split corresponds to the root node in the classification tree. The left child then inherits Xb, the set of training samples less than the threshold, while the right child inherits the complement, Xa. The splitting process is repeated recursively on each child, which results in further thresholds and nodes in the tree. Each node in the tree corresponds to a hyper-rectangular region or "cell" in the feature space, which is in turn subdivided by its descendants. Cells corresponding to the leaves of the tree completely partition the feature space into non-overlapping regions, as shown in Figure 1.

To calculate the mutual information $I(X;C)$ of a split, consider a threshold t in dimension d. The mutual information from the split is easily estimated from the training data in the following manner. Over the volume of the current cell, count the relative frequencies:

N_{ij} = Number of data points in cell j from class i

N_j = Total number of data points in cell j

$$= \sum_i N_{ij}$$

A_i = Number of data points from class $i : x_d \geq t_d$

In the region of cell j, define $Pr(c_i)$ to be the probability of class i and $Pr(a_i)$ as the probability that a member of class i is above the given threshold. These probabilities are easily estimated as follows:

$$\Pr(c_i) \approx \frac{N_{ij}}{N_j} \tag{3}$$

$$\Pr(a_i) \approx \frac{A_i}{N_{ij}} \tag{4}$$

With these probabilities, the mutual information given the threshold may be estimated in the following manner (for clarity of notation, conditioning on the threshold is not indicated):

$$I(X;C) = H(C) - H(C|X) \tag{5}$$

$$= -\sum_i \Pr(c_i) \log_2 \Pr(c_i) + \sum_i \Pr(c_i) H_2\left(\Pr(a_i)\right) \tag{6}$$

$$\approx -\sum_i \frac{N_{ij}}{N_j} \log_2 \frac{N_{ij}}{N_j} + \sum_i \frac{N_{ij}}{N_j} H_2\left(\frac{A_i}{N_{ij}}\right), \tag{7}$$

where H_2 is the binary entropy function

$$H_2(x) = -x \log_2(x) - (1-x) \log_2(1-x). \tag{8}$$

Equation 7 is a function of the (scalar) threshold t, and may be quickly optimised by a region-contraction search.

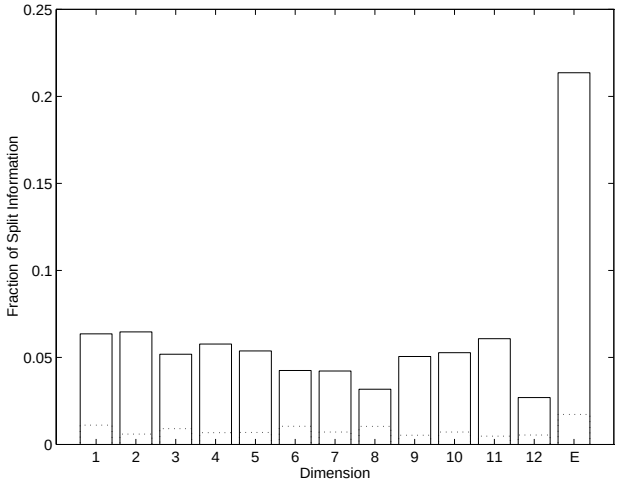

Figure 2. Fraction of mutual information by feature.

This splitting process is repeated recursively on each child, which results in further thresholds and nodes in the tree. At some point, a stopping rule decides that further splits are not worthwhile and the splitting process is stopped. The MMI criterion works well for finding good splits, but is a poor stopping condition because it is generally non-decreasing. (Imagine a tiny cell containing only two data points from different classes: any hyperplane between the points will yield an entire bit of mutual information. Bigger cells with overlapping distributions generally have less mutual information.) Also, if the number of training points in a cell is small, the probability estimates for that cell may be unreliable. This motivates a stopping metric where the best-split mutual information is weighted by the probability mass inside the cell l_j to be split:

$$\text{stop}(l_j) = \left(\frac{N_j}{N}\right) I_j(X;C) \tag{9}$$

where N is the total number of available training points. Further splits are not considered when this metric falls below some threshold. This mass-weighted MMI criterion thus insures that splitting is not continued if either the split criterion is small, or there is insufficient probability mass in the cell to reliably estimate the split threshold.

1.3. Dimensional Importance

An interesting side-effect of tree construction is that the relative importance of feature-space dimensions can be estimated. The maximum mutual information given by a split

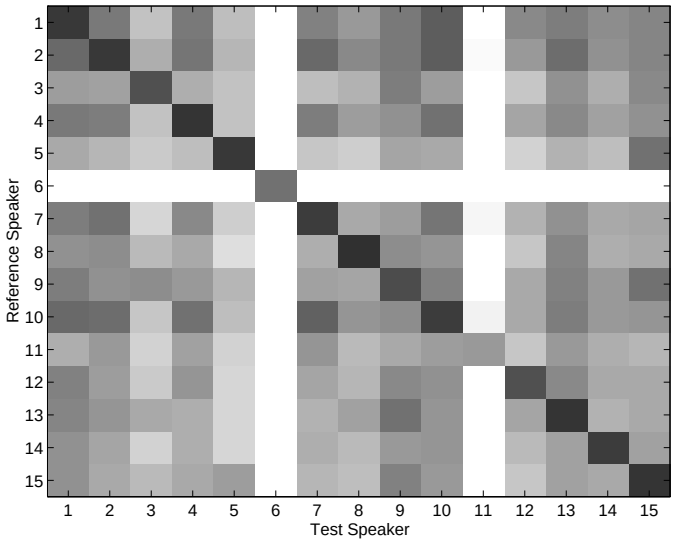

Figure 3. Speaker distance matrix (darker = closer)

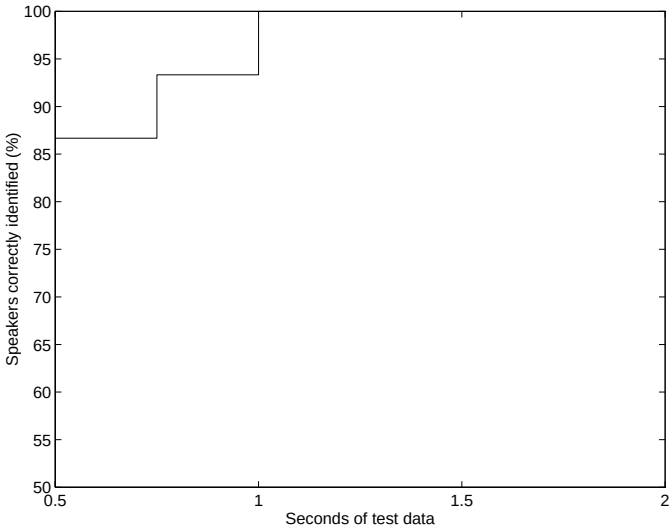

Figure 4. Speaker Identification Accuracy vs. Test Speech

in a particular dimension will depend on how well the feature values are correlated with the class labels. The relative "importance" of each feature may then be judged by looking at its contribution to the total mutual information. Figure 2 shows the fraction of mass-weighted mutual information given by each dimension for a tree grown on the speech data of Section 3. The relative importance of 12 mel-cepstral coefficients, log energy, and differences thereof are plotted, with the differential values indicated by a dotted horizontal line. Figure 2 shows clearly that the energy and primary cepstral features are most important. Unlike speech recognition, the higher-order cepstral features are important as they are dependent on fine spectral shape and thus vocal tract and pitch differences that characterise speakers. Though the differenced parameters are important for speech recognition, they are much less necessary for identification, and could probably be eliminated without affecting identification performance.

2. TREE-BASED TEMPLATE GENERATION

The tree partitions the feature space into L non-overlapping regions or "cells," each of which corresponds to a leaf of the tree. For speech recognition, the tree may be used as a vector quantiser front-end for a discrete HMM system [5]. For the speaker identification experiments described here, the system is used as a simple vector quantiser.

Given an amount of speech data from a particular speaker, the ensemble of leaf probabilities from the quantised data will characterise that speaker. A second of unknown speech will result in 100 feature vectors (ignoring edge effects), and thus 100 different leaf labels. If a histogram is kept of the leaf probabilities, such that if, say, 14 of the 100 unknown vectors are classified at leaf j then leaf j is given a value of 0.14 in the histogram. The resulting histogram uniquely classifies a speaker, regardless of whether the speaker was used for tree construction.

Given speech from an unknown speaker, a similar histogram may be estimated and compared with stored templates from the reference speaker. The closest matching template then identifies the unknown speaker. Though it is not obvious how to choose an appropriate distance measure to compare the templates, simple approaches work well. For the experiments presented here, the Euclidean distance between the two templates is used, as it is closely related to to the χ^2 measure. Related work has also used symmetric relative entropy as a distance measure [1].

Because of the large span of the template histogram (2024 entries), many counts will be zero, especially for the very short lengths of test data used in the experiments. Additionally, the most populous histogram entries will be those corresponding to silence. Since neither zero-count labels nor silence labels help to discriminate between talkers, the distance measure is computed only between moderately populated histogram entries. This is done by sorting the histogram computed by summing all the reference templates, and finding the third through the 303rd largest entries. All other entries are are ignored in the distance measure.

3. EXPERIMENTS

A subset of the VMR1 corpus was used for both enrolment and test data [6]. Each of the 15 speakers (11 male and 4 female) provided several read utterances There were fifteen speakers, of which 11 were male and 4 female. Data was recorded at 16 kHz from a Sennheiser HMD 414 close-talking microphone in an acoustically quiet room. Each speaker read several sentences from the TIMIT corpus, which was designed to be phonetically rich (though the amounts of data used here are far too small to cover all phone articulations).

Forty-five utterances (three sentences from each of the 15 speakers) were used to train the quantisation tree, a total of 195 seconds of data. Training data was labeled with silence and the particular speaker, using an existing HMM system [7]. The resulting tree had 2024 leaves (thus possible output labels).

Ten sentences from each speaker was used for enrolment data; an average of 41 seconds per speaker (of which a significant amount was silence). Reference templates were generated for each of the fifteen speakers using the tree and the methods of Section 2..

Ten different sentences from each speaker were used as test data, again, about 40 seconds per speaker. This resulted in ten test templates. The Euclidean distance was computed between each test and reference template; in all cases the distance to the same-speaker reference template was smaller than any intra-speaker distance. Thus the distance metric used for speaker identification results in 100% identification accuracy on this test set. Figure 3 displays the speaker distance matrix graphically. The intersection of row i and column j represents the distance between the

speaker i reference template and the speaker j template; the darker the element the more distant the speaker. The closest distances in a column are clearly on the diagonals, which are the distances between the test and reference models from the same speakers.

A useful speaker identification system should require only a small amount of test data. However, identification error will increase as the amount of test speech shrinks, because many relative frequencies will be zero, and others will be unreliable because of insufficient data. To investigate the identification accuracy as a function of available test data, the same 15-speaker experiment was performed but the amount of data available for the test models was varied from 0.5 to 2 seconds, as shown in Figure 4. Note that identification accuracy is perfect using substantially less than two seconds of test speech; even with one-half second of speech, only two speakers are incorrectly identified (random identification would result in fourteen misclassifications).

4. CONCLUSIONS

A rapid and effective method for speaker identification has been presented. Though this method shows promise, more work must be done. Experiments on a larger corpus (such as the YOHO corpus specifically designed for speaker ID work) would show that the method is robust to bigger user populations. In addition, experiments with speakers not in the enrolment set should be performed.

These experiments show that useful identification can be performed with a surprisingly small amount of data. Even with only half a second of test data, 13 of the 15 speakers were correctly identified. More sophisticated distance measures, especially rank-based metrics which may be more robust to sparse data, should allow rapid running speaker identification; for example to locate new speakers in an audio soundtrack.

A large motivation for using MFCC parameterisation for speech recognition is because the resulting features are reasonably uncorrelated. Because the tree quantiser can usefully model correlation, it may be possible to find parameterisations that better capture speaker-dependent features, especially when the importance of additional features can be judged by the tree.

REFERENCES

[1] J. T. Foote and H. F. Silverman. A model distance measure for talker clustering and identification. In *Proc. ICASSP 94*, volume S1, pages 317–320, April 1994.

[2] L. Wilcox, F. Chen, and V. Balasubramanian. Segmentation of speech using speaker identification. In *Proc. ICASSP 94*, volume S1, pages 161–164, Adelaide, SA, April 1994.

[3] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, Calif., 1984.

[4] M. Anikst et al. The SSI large-vocabulary speaker-independent continuous-speech recognition system. In *Proc. 1991 ICASSP*, pages 337–340, 1991.

[5] J. T. Foote. Discrete MMI probability models for HMM speech recognition. In *Proc. ICASSP 95*, volume 1, pages 461–463, Detroit, May 1995. IEEE.

[6] G. J. F. Jones, J. T. Foote, K. Spärck Jones, and S. J. Young. VMR report on keyword definition and data collection. Technical Report 335, Cambridge University Computer Laboratory, May 1994.

[7] J. T. Foote, G. J. F. Jones, K. Spärck Jones, and S. J. Young. Talker-independent keyword spotting for information retrieval. In *Proc. Eurospeech 95*, volume 3, pages 2145–2148, Madrid, 1995. ESCA.