A Similarity Measure for Automatic Audio Classification

Jonathan Foote
Institute of Systems Science
National University of Singapore
Singapore 119597
jtf@iss.nus.sg

Abstract

This paper presents recent results using statistics generated by a MMI-supervised vector quantizer as a measure of audio similarity. Such a measure has proved successful for talker identification, and the extension from speech to general audio, such as music, is straightforward. A classifier that distinguishes speech from music and non-vocal sounds is presented, as well as experimental results showing how perfect classification accuracy may be achieved on a small corpus using substantially less than two seconds per test audio file. The techniques a presented here may be extended to other applications and domains, such as audio retrieval-by-similarity, musical genre classification, and automatic segmentation of continuous audio.

Introduction

This paper presents a method of rapidly determining the characteristics of audio samples, using a supervised tree-based vector quantizer trained to maximise mutual information (MMI). Unlike other approaches based on perceptual criteria (Pfeiffer, Fischer, & Effelsberg 1996; Wold *et al.* 1996; Blum *et al.* 1996; Wyse & Smoliar 1995), this technique is purely data-driven and makes no attempt to extract subjectively "meaningful" acoustic parameters. Unlike hidden Markov modelling, this method is computationally inexpensive, yet is robust even with only a small amount of test data. Thus classification is rapid in terms of both computational cost and the small amount of test data needed to characterize the audio. Similar measures have proved successful for talker identification and clustering (Foote & Silverman 1994; Foote 1997). This paper presents a classifier that distinguishes speech from music and non-vocal sounds. This has immediate applications for speech recognition: in general, there is no guarantee that a given multimedia audio source contains speech, and it is important not to waste valuable resources attempting to perform speech recognition on music or non-speech audio.

The basic operation of the classifier is as follows. First, a suitable training corpus of audio examples must be accumulated and parameterized into *feature vectors*. The corpus must contain examples of the kinds (*classes*) of audio to be discriminated between, e.g. speech and music, or male and female talkers. Next, a tree-based quantizer is constructed using the methods of Section . This is a "supervised" operation and requires requires the training data to be *labeled*, i.e. designating to which *class* each training example belongs. This is all the human input required. The tree automatically partitions the feature space into regions which have maximally different class populations. Though this alone could be used as a classifier, it will not be robust as class distributions typically overlap a great deal, and are generally inseparable. Rather, the quantizer is used to generate a *template* that can be used as a reference. This is done by quantizing the test data for a particular class (by seeing in which partition the quantizer places each datum), and constructing a histogram of the resultant partition counts. A similar histogram may be computed for a test audio file, and some measure of "distance[1]" computed between the test histogram and each class reference. For a simple classification task, the unknown test audio is classified according to the most similar reference template.

Tree-based Quantization

Unlike the more common K-means vector quantization (VQ), the tree-based quantization is supervised, which means the feature space may be profitably discretized into many more regions than the conventional minimum-distortion vector quantizers. 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

[1] "Distance" is used here in a very loose sense, as the measures discussed may not be symmetric or satisfy the Triangle Inequality.

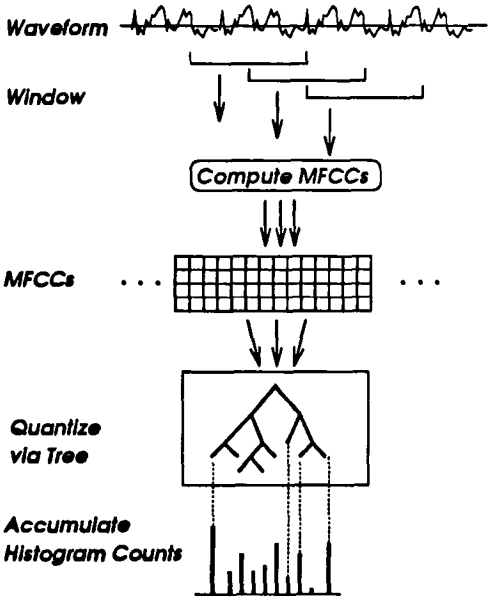

Figure 1: Audio template construction

amount of available enrollment 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 audio classification system works as follows. Both test and enrollment speech is first parameterized 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 parameterization 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 quantization tree is grown off-line, using as much training data as practicable. Such a tree is essentially a vector quantizer; discriminative training ensures that it attempts to label feature vectors from different classes with a different label. To generate a class template for subsequent identification, training data is quantized, 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 data may be compared.

To identify an unknown data, a pdf is computed from quantized test data in a similar manner. This test template can be compared with those from the

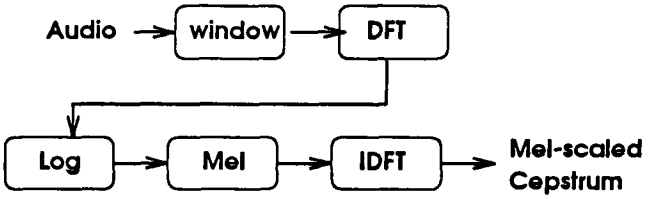

Figure 2: Mel frequency scaling (After Robinson 1995)

reference classes using one of any number of distance measures; the "closest" reference template then identifies the class of the test data. If the test data is not guaranteed to be in one of the reference classes, a distance threshold may be set to reject data that does not sufficiently resemble any reference model.

Supervised MMI Trees for quantization

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 class of each datum.

Tree construction

Because the construction of optimal decision trees is NP-hard, they are typically grown using a greedy strategy (Breiman *et al.* 1984). The first step of the greedy algorithm is to find the decision hyperplane that maximizes the mutual information metric. While other researchers have searched for the best general hyperplane using a gradient-ascent search (Anikst & others 1991), 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

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

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

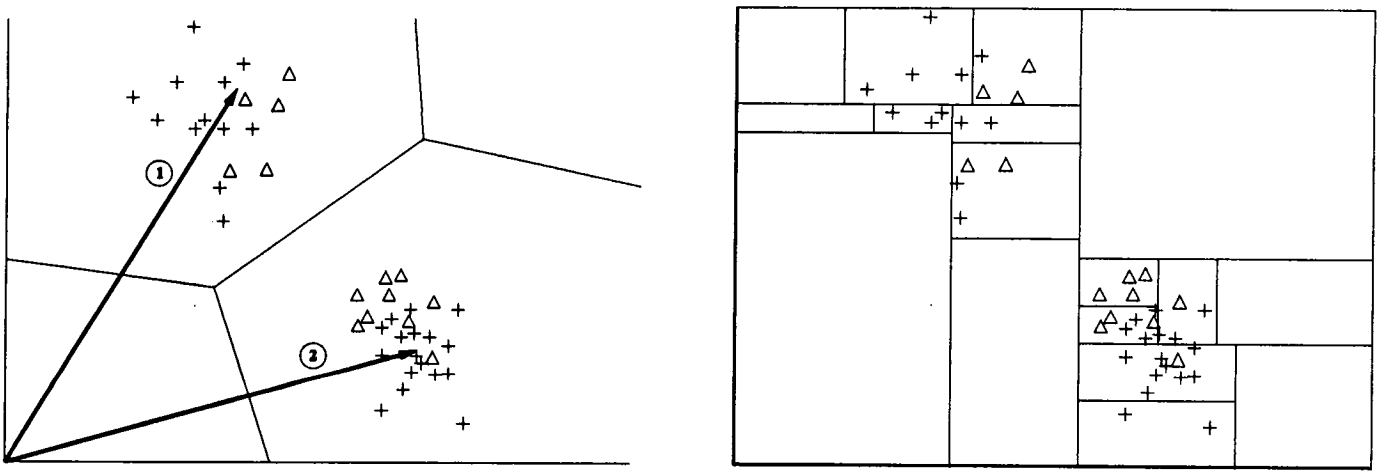

Figure 3: nearest-neighbor VQ (left) and MMI tree (right) feature space partitions

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 3.

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:

$$\begin{aligned} N_{ij} &= \text{Number of data points in cell } j \text{ from class } i \\ N_j &= \text{Total number of data points in cell } j \\ &= \sum_i N_{ij} \\ A_i &= \text{Number of data points from class } i : x_d \geq t_d \end{aligned}$$

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:

$$\Pr(c_i) \approx \frac{N_{ij}}{N_j}, \quad \Pr(a_i) \approx \frac{A_i}{N_{ij}}. \tag{3}$$

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):

$$I(X;C) = H(C) - H(C|X) \tag{4}$$

$$= -\sum_i \Pr(c_i) \log_2 \Pr(c_i) + \sum_i \Pr(c_i) H_2\left(\Pr(a_i)\right) \tag{5}$$

$$\approx -\sum_i \frac{N_{ij}}{N_j} \log_2 \frac{N_{ij}}{N_j} + \sum_i \frac{N_{ij}}{N_j} H_2\left(\frac{A_i}{N_{ij}}\right), \tag{6}$$

where H_2 is the binary entropy function

$$H_2(x) = -x \log_2(x) - (1-x) \log_2(1-x). \tag{7}$$

Equation 6 is a function of the (scalar) threshold t, and may be quickly optimised by either an exhaustive or region-contraction search.

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:

$$\text{stop}(l_j) = \left(\frac{N_j}{N}\right) I_j(X;C) \tag{8}$$

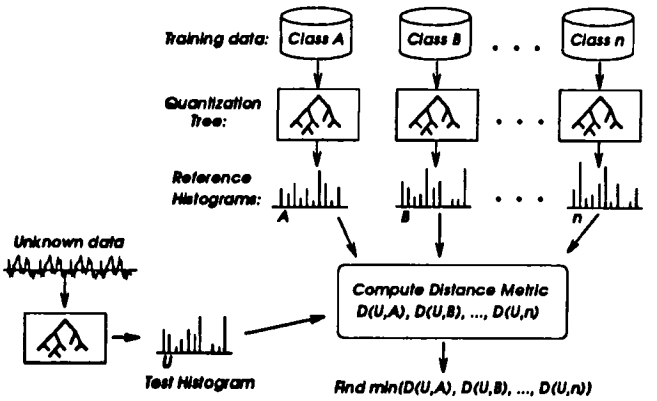

Figure 4: Audio classification using histogram templates

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.

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. Though the tree can be used as a classifier, by labeling each leaf with a particular class. When classifying a sufficient amount of class-labeled heterogeneous data, each leaf will get data from a number of classes "routed" to it. By choosing the most popular class, each leaf can be labeled with the the class whose data is most likely to end up there. Such a classifier will not be robust, as in general classes will overlap such that a typical leaf will contain data from many different classes. A better way to capture class attributes is to look at the the ensemble of leaf probabilities from the quantized class data. A second of data will result in 100 feature vectors (ignoring window 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 captures essential class qualities, and thus serves as a reference template against which other histograms may be compared.

Distance Metrics

Given data from an unknown source, a similar histogram may be estimated and compared with stored templates from the reference classes. The closest matching template then identifies the unknown data. Though it is not obvious how to choose an appropriate distance measure to compare the templates, simple approaches work well in practice. Several distance measures have been used in implementation. Given two histograms p and q of length N, denote the ith count of histogram p as $p(i)$. Assuming histograms are normalized $\sum_{i=1}^{N} p(i) = 1$ (and thus a true pdf), the following distance measures $D(p,q)$ may be used:

- *Euclidean distance*

$$D_E^2(p,q) = \sum_{i=1}^{N} [p(i) - q(i)]^2 \qquad (9)$$

This measure treats the histograms as vectors in N-dimensional space, and computes the L2 (Euclidean) distance between them. Though there is no true probabilistic justification for this measure, it is closely related to to the χ^2 measure, and was used successfully for speaker ID in Foote 1997.

- *Symmetric relative entropy*

$$(p||q) \triangleq \sum_{i=1}^{N} p(i) \log \frac{p(i)}{q(i)}.$$

In general, $(p||q) \neq (q||p)$, so

$$D_{RE}(p,q) \triangleq \frac{1}{2}\left[(p||q) + (q||p)\right].$$

This metric, (also called the information divergence or *Kullback-Liebler* distance), may exaggerate the difference between histograms because of the nonlinear dependence on the probability quotients. It also may not be robust to sparse histograms from small amounts of data. D_{RE} was used for speaker ID and clustering in Foote & Silverman 1994.

- *Correlation distance*

$$D_C(p,q) = \sum_{i=1}^{N} p(i)q(i) \qquad (10)$$

This is the distance metric used for the experiments of Section . This metric has the helpful property that zero histogram counts do not contribute to the measure, and thus it may be more robust to sparse histograms constructed from small data amounts. Note that if the histograms are considered vectors, this measure is similar to the "cosine distance" of information retrieval if the correlation is normalized by the product of the L2 norms of the histograms (Salton & Buckley 1987).

```
/sounds/bells/bellTower5.2notes.au
/sounds/bells/bellTower6.au
/sounds/bells/bellTower7.au
/sounds/crowds/crowd.au
/sounds/crowds/largeCrowd.au
/sounds/crowds/restaurant.au
/sounds/crowds/sidewalk.au
/sounds/laughter/laughterFemale1.au
/sounds/laughter/laughterFemale2.au
/sounds/laughter/laughterFemale3.au
/sounds/laughter/laughterYoungMale.au
/sounds/laughter/laughterYoungMale2.au
```

Table 1: example Muscle Fish sound classification hierarchy (after Wold 1996)

Class		*No. files*	*Tot. length (s)*
Male speech	(xm)	110	371.9
Female speech	(xf)	18	75.8
Non-speech	(fx)	43	101.7
Instrumental	(ix)	44	191.3
Rhythmic	(rx)	47	225.8

Table 3: Training data statistics

Because of the very small amount of test data used, many histogram counts will be zero. Additionally, the most populous histogram entries will often be those corresponding to silence. Because neither zero-count labels nor silence labels help to discriminate between classes, 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 5th through the 100th largest entries. All other entries are ignored in the distance measure. Note that an individual template has only 95 integer parameters, and is thus extremely compact.

Training data

Over 1000 ".au" files were collected from the Internet using the Lycos search engine, which has a provision for retrieving only audio files. A search request of "au" was used to ensure both relatively random file selection as well as the desired ".au" format. The URL of each file was automatically parsed from the Lycos search result and retrieved off-line using the `libwww-perl` package[2].

[2]The extent to which this infringes the intellectual property rights of the audio's creators/owners is not at all clear. It is assumed here that using this data for research falls squarely under the Fair Use clause of any copyright agreement and is thus permitted, however wide-reaching new legislation may change this; see `http://www.ari.net/dfc/`.

Class	*No. files*	*Avg. length (s)*
Music	34	1.98
Male speech	17	0.61
Female speech	35	0.52
Percussion	30	1.71

Table 4: Test data statistics

Of these files, 566 had the correct encoding, sample rate, and were unduplicated elsewhere in the set. Each file was auditioned and given a 2-letter code according to the scheme of Table 2. For the purposes of classification, it was desired to obtain samples of instrumental music (without singing, speech, or other vocalizations[3]), thus audio files designated "`rx`" and "`ix`" were selected as training examples of the "music" class, while files with "`xm`" and "`xf`" (there were insufficient `xc` files to warrant inclusion) served as exemplars of the "speech" class. Table 3 shows the resultant amounts of data, a little less than 1000 seconds in total.

Quantization trees were constructed using the data of table 3; each set of files corresponding to each of the five classes was labeled with the appropriate class. A single tree with 127 leaves was used for the classification experiments in the next Section. Note that more detailed trees could be grown and pruned, or existing trees could be pruned to have fewer leaves. This tree was used to generate two templates, one for speech (by generating a histogram from the `xf` and `xm` data) and one for music (using the `ix` and `rx` data).

Test data

Test data was obtained (with permission) from the Muscle Fish demonstration site (Wold 1996) because it had been conveniently pre-classified and thus needed no auditioning. Table 1 shows the Muscle Fish classification scheme. Three classes of audio files were used to test the distance measure automatic classification: plain speech from male and female talkers, music files (most containing sung vocals), and long-duration percussion samples (e.g. cymbal crashes) as examples of non-speech, non-music audio. Note that most of the test music data had sung vocals, in contrast to the training music data which was strictly instrumental. Music samples were truncated at two seconds, while the speech sample were single words of about 1/2 second duration. Table 4 shows the number and average length of the test files used.

[3]Like those audible in "heavy metal."

Music component:	Speech component:
i - music (not primarily rhythmic)	v - non-speech, non-laughter vocalization
r - rhythmic/percussive music	m - male speech
f - noise or non-vocal sound effect	f - female speech
p - processed, reverbed or noisy	c - child speech or baby
s - pitched singing or chanting	l - laughter
x - no music or other component (x implies clean speech only)	a - animal sounds
	x - no speech or singing

Table 2: Training data classification scheme

Experiments

Figure 5 presents a plot of the correlation distance D_C from the speech and music templates for each test audio sample. A larger value indicates a higher correlation, and thus more similarity. Though it is not visible in the plot, there is a male speech data point (×) in the cluster composed primarily of percussion samples (*). Some things to note in Figure 5 :

- Data points from the different classes are exceptionally well-clustered, especially considering the short duration (less than 2s) of the test samples.
- Male speakers and female speakers were reasonably well-distinguished, *even though the reference templates were constructed with speech from both genders*. This is perhaps due to the imbalance between male and female speech in the template training data: there was nearly five times as much male speech as female speech[4].
- Distance from the speech template alone did not discriminate well between the speech and music samples, undoubtedly because the music contained vocals (speech-like attributes). The outlier music sample near [0.6, 0.9] is a particular excerpt of Nat King Cole singing "Ain't Misbehavin' " that has only a faint string accompaniment, and is thus mostly vocal in nature.
- Distance from the music template, however, did an excellent job of discriminating music from speech and percussion samples. A simple threshold of $D_C = 0.5$ would correctly identify all test samples as music or non-music (speech/percussion).
- Except for the one male speech outlier previously mentioned, distance from the speech template would serve to discriminate speech from percussion samples. This is notable as no instances of percussion were used to train either the speech or music templates (though there were some percussive sounds in the `fx` data used to train the tree).

[4]The author realizes this is not gender-neutral but argues that this only reflects the existing bias on the Internet.

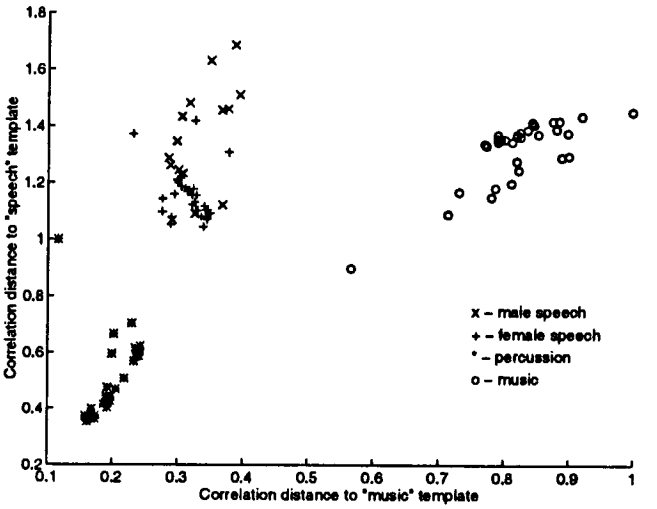

Figure 5: Similarity distances to speech and music templates

Conclusions and Future Directions

An effective method for audio classification has been presented, showing that useful identification can be performed with a surprisingly small amount of data. The distance measure as described here is useful as a general measure of audio similarity, and could be applied to retrieving audio documents by similarity: documents could be ranked by their distance to an example audio template. Given the very modest storage requirements of the templates, this would be practical for even extremely large archives.

This technique offers perhaps a way to measure subjective perceptual qualities of sounds, often described in terms like "brightness" and "harmonicity." Rather than defining and computing an actual measure of these relatively subjective terms, it is possible to train

a template with a number of example files deemed to have (or not have) the given quality. The resultant distance from that template may be used as a measure of the particular quality, *without having to explicitly define it.*

Another more difficult application is to automatically segment multimedia sources by audio *changes*, such as between different speakers (Wilcox, Chen, & Balasubramanian 1994; Roy & Malamud 1997), pauses, musical interludes, fade-outs, etc. Because the identification technique works well enough on a subsecond time scale, it could be used to detect these changes simply by looking at the histogram generated by a short running window over a longer audio stream. Comparing the window histogram with pretrained templates would allow detection and segmentation of speech, particular speakers, music, and silence. Another approach would be to compute the distance between the histogram of a short window with a longer window, which might yield a measure of audio novelty by the degree that short-term statistics differ from a longer-term average.

These would be the audio equivalents of scene or camera changes, cuts, fades and wipes. It should be possible to fuse data intelligently extracted from both the visual and aural modes, yielding more complete and robust information (about key frames, for example) than is available from either mode alone. A video corpus is being gathered at ISS for the purpose of segmentation research.

A large motivation for using MFCC parameterization for speech recognition is because the resulting features are reasonably uncorrelated. Because the tree quantizer can usefully model correlation, it may be possible to find parameterizations that better capture speaker-dependent features, especially when the importance of additional features can be judged by the tree. Additional features such as pitch or zero-crossing rate (as in Saunders 1996) would probably aid classification. An interesting possibility, yet unexplored, is to use compressed audio (for example MPEG encoded parameters) directly. This would eliminate the need for the parameterization step as well as decoding and would thus be extremely rapid.

Acknowledgments

This work was primarily funded by a J. William Fulbright research fellowship, administrated by the Committee for the International Exchange of Scholars. Thanks to and the staff at the Institute for Systems Science for additional support, and to Erling Wold for permission to use portions of the Muscle Fish audio corpus (Table 4).

References

Anikst, M., et al. 1991. The SSI large-vocabulary speaker-independent continuous-speech recognition system. In *Proc. ICASSP '91*, 337–340. IEEE.

Blum, T.; Keslar, D.; Wheaton, J.; and Wold, E. 1996. Audio analysis for content-based retrieval. Technical report, Muscle Fish LLC, 2550 Ninth St., Suite 207B, Berkeley, CA 94710, USA. `http://www.musclefish.com/cbr.html`.

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

Foote, J. T., and Silverman, H. F. 1994. A model distance measure for talker clustering and identification. In *Proc. ICASSP '94*, volume S1, 317–320. IEEE.

Foote, J. T. 1997. Rapid speaker ID using discrete MMI feature quantisation. In *Proc. Pacific Asian Conference on Expert Systems.*

Pfeiffer, S.; Fischer, S.; and Effelsberg, W. 1996. Automatic audio content analysis. Technical Report TR-96-008, University of Mannheim, D-68131 Mannheim, Germany. `ftp://pi4.informatik.uni-mannheim.de/pub/techreports/1996/TR-96-008.ps.gz`.

Roy, D., and Malamud, C. 1997. Speaker identification based text to audio alignment for an audio retrieval system. In *Proc. ICASSP '97.* Munich: IEEE.

Salton, G., and Buckley, C. 1987. Term weighting approaches in automatic text retrieval. Technical Report TR87-881, Department of Computer Science, Cornell University, Ithaca, New York 14853-7501. `http://cs-tr.cs.cornell.edu/Dienst/UI/2.0/Describe/ncstrl.cornell%2fTR%87-881`.

Saunders, J. 1996. Real-time discrimination of broadcast speech/music. In *Proc. ICASSP '96.* Atlanta: IEEE.

Wilcox, L.; Chen, F.; and Balasubramanian, V. 1994. Segmentation of speech using speaker identification. In *Proc. ICASSP 94*, volume S1, 161–164.

Wold, E.; Blum, T.; Keslar, D.; and Wheaton, J. 1996. Content-based classification, search, and retrieval of audio. *IEEE Multimedia* 27–36.

Wold, E. 1996. Muscle Fish Audio CBR Demo. `http://www.musclefish.com/cbrdemo.html`.

Wyse, L., and Smoliar, S. 1995. Toward content-based audio indexing and retrieval and a new speaker discrimination technique. In *Proc. ICJAI '95.*