Automatic Speaker Recognition: An Application of Machine Learning

Brett Squires
School of Computer Science and Engineering
University of New South Wales
Sydney Australia 2052
bretts@cse.unsw.edu.au

Claude Sammut*
School of Computer Science and Engineering
University of New South Wales
Sydney Australia 2052
claude@cse.unsw.edu.au

Abstract

Speaker recognition is the identification of a speaker from features of his or her speech. This paper describes the use of decision tree induction techniques to induce classification rules that automatically identify speakers. In a population of 30 speakers, the method described has a recognition rate of 100% for both text dependent and text independent utterances. Training times scale linearly with the population size.

1 INTRODUCTION

Speaker recognition is the identification of a speaker from features of his or her speech. This paper describes the use of machine learning techniques to induce classification rules that automatically identify speakers.

The most common application for speaker identification systems is in access control, for example, access to a room or privileged information over the telephone. Usually the task is simplified to speaker verification, where the speaker makes an identity claim and then the claim is either verified or rejected. The task described in this paper is somewhat more difficult since the system must select the identity of a person from a known population. In access control applications we assume that the environment can be controlled by reducing noise and interference and the speaker is cooperative, i.e. he or she responds to instructions. We make the further assumptions that there is only one speaker at a time and the speaker does not deliberately try to disguise his or her voice.

The goal of speaker recognition is to be able to automate the process of building a recognition system that can identify a person by using measurements of his or her speech waveform. These measurements are usually highly redundant and one of the problems in building the recognition system is to filter out the redundant measurements from the truly informative ones. Furthermore, some measurements must be processed to generate trends that are often more informative than the raw measurement. It is not easy to build a speaker recognition system manually. Nor is it easy to maintain such a system over time as new speakers are introduced into the population or a speaker's characteristics change slightly. Therefore, machine learning is an obvious approach to try.

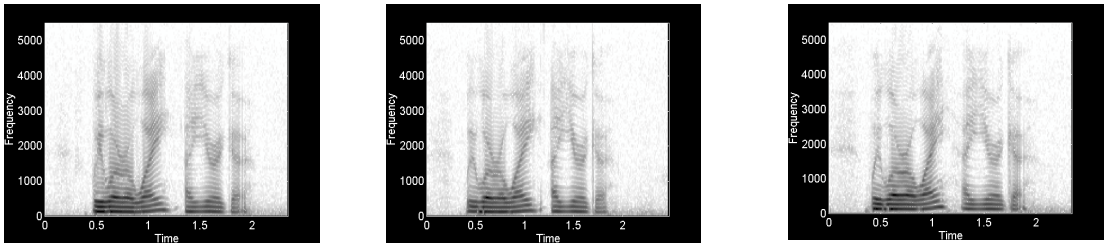
Previous attempts at automatic speaker recognition have used a variety of methods including nearest-neighbour matching, neural nets and statistical approaches.

Nearest-neighbour algorithms are used in two methods, namely, template matching and vector quantisation. In template matching, an instance of a spectral template for each speaker is stored as a reference. To identify a speaker, a nearest neighbour match is attempted on each template to find the best match. Some examples of speech templates are shown in Figure 1.

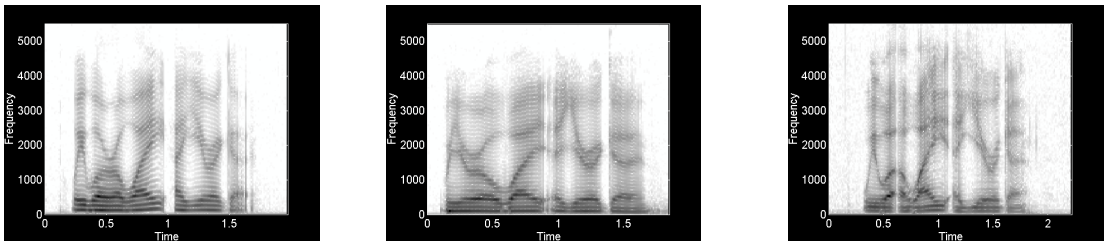
In vector quantisation (Burston, 1987; Matsui & Furui, 1992), a 'code book' stores spectral feature vectors as prototypic instances of the short-term characteristics of each speaker's voice. Unlike template matching, vector quantisation stores many feature vectors for each speaker to increase the likelihood of a correct match.

Hidden Markov models (HMM) have been used for text-dependent speaker recognition, that is, the speakers must repeat a particular utterance to be recognised (Rosenberg, Lee & Gocken, 1992). With HMM's, an utterance is characterised by a sequence of transitions from one speech event to another through a series of Markov states. The Markov states themselves are hidden but are indirectly observable from the sequence of spectral feature vectors output by the model. The parameters that describe the HMM are the transition probabilities between the states and the probabilities of observing spectral vectors in each state. In speaker recognition a HMM can be used as an

* Author to whom correspondance should be addressed.



Spectrograms of an utterance by the same speaker



Spectrograms of an utterance by three different speakers

Figure 1: Templates

adaptive template that statistically learns to fit the expected feature transitions across a word.

Rudasi and Zahorian (1992) applied neural nets to text independent speaker recognition with some success. They found that training a single net to perform N-way classification of N speakers became exponential with N. However, by training many separate nets to distinguish between pairs of speakers, the training time was order N^2.

These previous approaches have their limitations. Template and HMM speaker recognition systems have largely been limited to text-dependant utterances. Both template and vector quantisation methods have large memory requirements and are slow in recognising speakers. However, they are suitable for speaker verification since only one lookup of a template is required. Neural nets are slow to train as the population size increases and require a considerable amount of tuning to suit the application.

This paper describes the use of decision tree induction techniques to induce classification rules that automatically identify speakers. In a population of 30 speakers, this method has a recognition rate of 100% for both text dependent and text independent utterances. Training times scale linearly with the population size.

2 PREPROCESSING

A major part of the success of our method is due to the preprocessing that transforms the speech signals into the data for induction. The preprocessing stages are shown in Figure 2.

In the experiments reported here, speakers were asked to repeat the same utterance five times[1]. These were recorded using a microphone of moderately high quality in a quiet but otherwise normal environment. Four of the utterances were used for training. The fifth was used for testing.

Each utterance is broken into overlapping frames taken in 10 milliseconds increments each with a duration of 30 to 40 milliseconds. Signal processing is performed on each frame to collect measurements of short-term characteristics of the voice signal. Frames are then labelled according to the name of the speaker. However, only some frames are retained, namely, those frames that contain the vowel content of the speech. Each labelled frame will be used as a training example for input to C4.5 (Quinlan, 1993).

Next we briefly describe the signal processing that is performed on a frame.

The first function to be applied to the speaker sample is a simple silence classifier that scans the frames of speech to determine if a frame has enough energy to be considered part of the utterance spoken. Speech attributes gained as a part of the silence analysis include the silence/noise classification, the average magnitude of the frame and the number of zero crossings in each frame.

[1] In all of the experiments described in this paper, the utterance is "My name is .... My voice is my passport. Verify me." We stole it from the movie *Sneakers*.

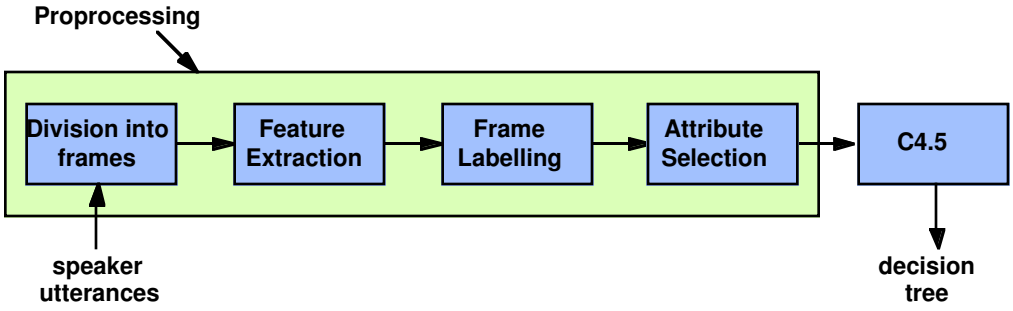


Figure 2: Stages in building the classifier

In order to speed up the measurement of pitch, voicing analysis is applied. This scans all noisy frames looking for a periodic pitch signal. Any noisy frames that contain a reasonably periodic signal are classified as *voiced* and all others are classified as *unvoiced*. Voiced frames correspond to vowels and sonorant consonants whereas unvoiced frames correspond to plosive sounds.

Following voicing analysis, the pitch of each voiced frame is calculated and stored as an attribute of the frame. Attributes were also derived from the pitch contour over each set of 15 consecutive frames (150ms). These measured the average pitch and the change in pitch over the 150ms interval.

The next stage of preprocessing calculates the linear predictive coefficients of a model of the vocal tract. This is an approximation to the true vocal tract transfer function that transforms the fundamental frequency (i.e. the pitch) into the complex waveform that radiates from the speaker's lips. Reflection coefficients are also calculated. These represent the structure of the different tubes in the vocal tract.

We also calculate the cepstral coefficients that capture the short-term trends of the spectral envelope and represent the transitional information between phonemes or vowels.

Next, we calculate the formant frequencies, bandwidths and relative energy. Formants are the main resonances of the vocal tract. There are three strong formants within the voice from the mouth cavity, the nose cavity and the throat cavity. Formants can be used the classify the vowel content of our speech.

Finally simple statistical analysis calculates trends and distributions for several short term measurements.

A selection of the features for each labelled frame is combined into an example case. The set of examples from a population of speakers is input to C4.5 to produce a classifier for the speakers. In section 4 we describe how features are selected.

3 SPEAKER CLASSIFICATION

Once a decision tree has been built, it is used as a component of the complete classification system. The classification process is shown in Figure 3.

As before, an utterance goes through preprocessing to produce a set of frames, each described by a set of attributes resulting from the feature extraction. Our software provides the ability to select a subset of attributes to pass to the decision tree produced by C4.5.

The decision tree is used to classify each frame as being spoken by a particular person. The identity of the speaker is finally decided by taking the majority class of the decisions for each frame. We will show that while the classification accuracy for individual frames is not very

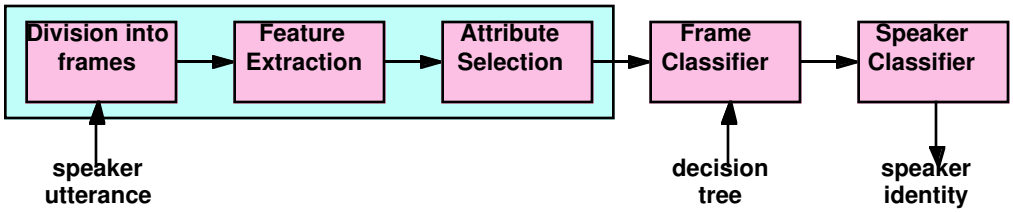


Figure 3: Speaker Classification

reliable, taking the majority class proves to be very reliable.

4 EXPERIMENTAL RESULTS

A series of experiments was conducted to investigate the best use of attributes and to discover how well the system would scale with population size.

4.1 GLOBAL MEASUREMENTS

The aim of the first experiment was to examine the usefulness of global measurements. Recall that these are attributes of the entire utterance, not just a single frame. The global measurements used to represent utterances were the mean, standard deviation and dispersion of the pitch, reflection coefficients and cepstrum coefficients. There are 12 reflection coefficients and 12 cepstrum coefficients, so there are 12 reflection coefficient means, 12 reflection coefficient standard deviations, etc.

The speakers, 10 female and 15 male, each made five repetitions of the same utterance for text-dependant speaker recognition. The measurements described above were then calculated from each utterance. Those taken from four of the five utterances of each speaker were used to induce a decision tree. The remaining measurements were reserved for testing.

When all the measurements were used as attributes, C4.5 obtained an actual error rate of 16%. It is possible that C4.5 erred in the selection of the most informative attributes. To check this, subsets of the attributes were systematically selected and tested. The best error rate obtained was 4% with the pitch mean and the cepstrum coefficient dispersion attributes.

It was clear that C4.5 suffered from a lack of data. There is a relatively small sample size and a large number of attributes. Unfortunately, it is impractical to obtain more data since a speaker recognition system should not require each speaker to provide more than a few samples. Furthermore, the statistics of speech attributes require at least 30 seconds to stabilise (Markel, 1972). Therefore,

Table 1: Induction with short-term attributes (10 speakers – 5 male, 5 female)

Attributes	Number of Attributes	Tree Size	Local Error	Misclassified speakers
Pitch contour, normalised utterance time	16	1761	35.8%	0
Pitch contour, normalised magnitude contour, ZCR contour	45	2127	38.9%	0
Formant contours: frequency, bandwidth, energy	48	2573	38.9%	0
Formant frequency contours:, formant bandwidth and energy	20	2521	44.7%	0
Cepstral coefficient contour	36	2147	40.5%	0
Cepstral coefficients, 1st order Δ-cepstral coefficients	24	2223	38.9%	0
Cepstral coefficients, 1st and 2nd order Δ−cepstral coefficients	36	2197	40.8%	0
Reflection coefficients, cepstral coefficients	24	2373	41.4%	0
LPC, reflection coefficients	24	2555	42.9%	0
LPC, reflection coefficients, cepstral coefficients	36	2381	41.4%	0
Pitch contour, formant frequency contour	35	1587	30.3%	0
Pitch contour, cepstral coefficients	27	1407	27.6%	0
Pitch contour, formant frequency contour, cepstral coefficients	47	1349	27.1%	0
Pitch contour, formant frequency contour, reflection coefficients , cepstral coefficients	54	1351	26.1%	0
Pitch contour, formant frequency contour, reflection coefficients , cepstral coefficients, normalised utterance time	55	1177	25.7%	0
all attributes	128	1231	26.7%	0

Table 2: Pruning the decision trees (20 speakers)

Minimum number of examples required for split	Tree Size	Local Error	Misclassified speakers
2	3461	39.0%	0
5	2201	39.0%	0
10	1337	42.6%	0
50	367	47.8%	0
100	193	53.0%	2

the next experiment went on to investigate the use of purely short-term attributes, i.e. measurements performed on each frame. Since each frame contributes one example and an utterance contains thousands of frames, the data sets are much larger.

4.2 USING SHORT-TERM ATTRIBUTES

The aim of this experiment was to determine if the short-term attributes are sufficient to identify speakers and to determine which of these attributes perform best. In these experiments, five male and five female speakers produce the same utterance five times, one of which is kept for test data. Many more training examples are available now because each frame contributes one example. The attributes are the short-term measurements described in section 2. Like the previous experiment, C4.5 was applied to data sets with all of the measured attributes and also to data sets in which only subsets of attributes were present. Our intention was to determine how effective C4.5 was in finding the best attributes and to determine if fewer attributes could be used to speed up learning times. Table 1 shows the results of these experiments.

The first column shows the different combination of attributes selected for each experiment, the last row showing the results of providing all the attributes to C4.5. The 'local error' is the error rate when trying to classify individual frames as to the identity of the speaker. This shows that the decision trees were not very good at recognising the speaker from a single 30-40ms frame. Suppose there are 1000 voiced frames in an utterance and we run the decision tree on each one of those frames. We then take frequency count for each class that resulted. For example, 732 frames were classified as speaker 'A' while 224 were classified as speaker 'B' and so on. If we chose the speaker with the majority class, we find that the correct speaker is identified consistently. The reason that this works is simple. As long as the local error rate is less than 50%, the majority class should always be the correct speaker. However, if the local error rate exceeds 50%, as long as the erroneous classes are evenly distributed amongst the misclassified frames, it is likely that the majority class will still be the correct speaker.

The second last trial shows the best results that could be obtained in terms of accuracy and tree size. It also shows which combination of attributes is the most useful. The tree output by C4.5, when all the attributes were input, had those attributes near the top of the tree. So C4.5 did a good job in finding the most informative attributes. However, the trees were large because of the highly variable short-term features used to make a classification. So, two questions remain: can the tree size be improved and how well does this method scale up to a larger population?

4.3 PRUNING

All the experiments described so far used C4.5's default pruning parameters. In the following experiments we forced C4.5 to prune more heavily in an attempt to find smaller trees that could still perform well. Two pruning parameters can be changed: the pruning confidence level and the stopping criterion, i.e. the minimum number of examples required in a node for a split. Experiments were performed with both parameters. It was found that varying the confidence level between the default 25% down to 1% made little difference to the error rate or the tree size. However, varying the stopping criterion reduced the tree size dramatically. Table 2 shows the results of the pruning experiments. Samples from 20 speakers were used in this set of experiments. As always, each speaker repeated the same utterance five times and one was reserved for testing.

Clearly, the tree size can be reduced significantly while still providing reliable classification for the population of 20 speakers. Note that there is a trade-off between accuracy and readability. While, the initial large trees are more accurate, the smaller trees are easier to interpret. However, as the local error rate increases, the 'vote' becomes unreliable.

4.4 CHANGES IN POPULATION SIZE

To observe the effects of increasing population size, a series of experiments was performed in which the population was increased from 5 to 30 speakers by adding two or three speakers at a time. The effect on the local decision error of increasing the population size is shown

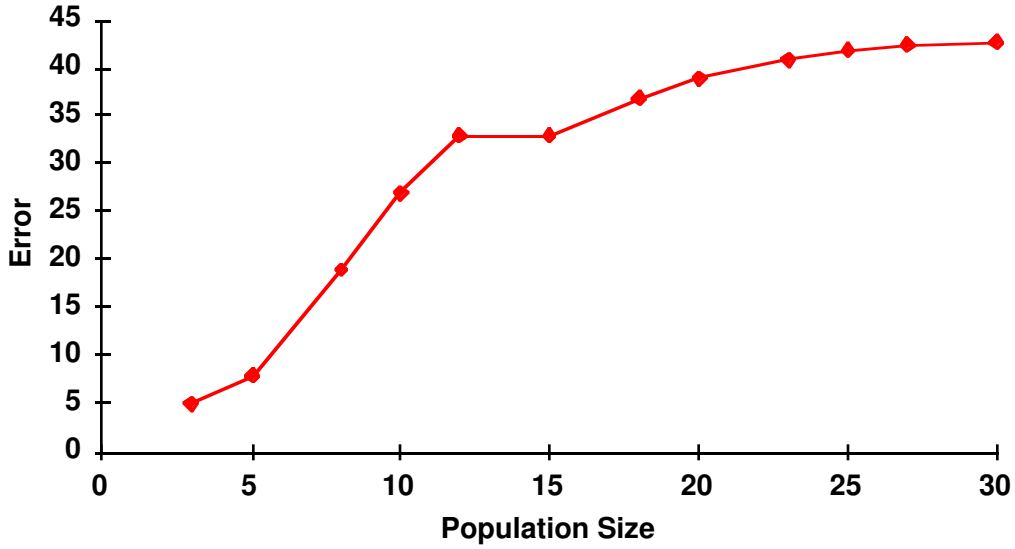


Figure 4: Local decision error as a function of population size

in Figure 4. No speakers were misclassified for any population size.

The local decision error increases monotonically, but appears to approach a limit between 45% and 50%. Provided that this error remains bounded below 50% the method should safely scale to larger populations. Of course, these data are not conclusive, but they are very promising. Furthermore, there are other ways of containing error rates.

So far, we have only described experiments in which a single, multi-class decision tree is built. However, we have also constructed two-class trees for each speaker. That is, for each speaker, we label the examples as positive for utterances spoken by that person and negative for all the others. Normally, a random sample is taken from the negative data so that the number of positive and negative examples is roughly the same. So if it is found that some large population size causes the induction of a single multi-class tree to 'blow up' there is at least one alternative for scaling up.

Further experiments with single multi-class trees show that the size of the decision tree grows linearly with population size. This is true for default pruning levels and for 'over pruning', although the rate of growth for over-pruned trees is considerably slower. Furthermore, training times are linear with population size. Details of these experiments are given by Squires (1994).

4.5 TEXT INDEPENDENT SPEAKER RECOGNITION

All the experiments so far have tested 'text-dependant' speaker recognition, that is, the same utterance was used for training and testing. In text-independent recognition, the training and test utterances need not be the same. Three experiments were conducted to test the suitability of our method for this kind of recognition.

A decision tree was trained using five repetitions of the same utterance from 30 speakers. The decision tree was used to classify five new utterances from the 30 speakers, a total of 150 utterances. Speakers were identified with 66% accuracy. However, the next experiment combined all five utterances from each speaker into a single, long utterance that resulted in 100% correct classification.

In the final experiment, involving six speakers, the utterance used for training was changed to a longer one lasting at least 90 seconds. When tested on the five short utterances as in the first experiment, 93% accuracy was achieved. Thus, reliable text-independent speaker recognition is feasible but requires longer test utterances or a broader range of training examples.

4.6 PERFORMANCE OVER TIME

All the test data used in the previous experiments were collected at the same time as the training data. Perhaps the strongest test for a speaker recognition system is to use test data collected some time after the initial training.

Training utterances from 30 speakers were collected in an initial recording session. The test utterances were recordings of the same utterance from 12 of the original speakers, 8 of whom repeated the utterance a second time. A total of 21 utterances were recorded between 3 days and 2 months after the initial recording session with each speaker. All 21 utterances were successfully recognised, however with lower confidence than in previous experiments. It was found that while the words of the utterance were the same in training and test cases, often the tone and attitude of the speakers changed significantly. In addition, the recording conditions were often different since the test cases were recorded in different locations. Thus, the decision trees proved to be quite robust with respect to natural variations to be expected in speaker recognition.

4.7 AN INFORMAL TEST

One of the most interesting tests of this method was an informal one, performed after the presentation of this work at a seminar at UNSW. A volunteer was asked to provide samples of his speech to demonstrate that the system really worked. As it happened, the volunteer was Ross Quinlan. The environment was the AI lab, a large, open-plan area housing about twenty desks and workstations. The speech samples were recorded in the middle of the lab and during the recording, a group of more than half a dozen attendees of the seminar were chatting in the background. To prevent the learning time from being too long, only the data of six other speakers were used. The system was able to distinguish Professor Quinlan's voice easily. While the small population size makes this demonstration less than a convincing test, the fact that the system did not get confused by the background speech is very encouraging for practical applications.

5 CONCLUSION

These experiments have shown that induction can successfully be applied to the problem of learning to automatically recognise a speaker using short-term attributes. Decision trees are used to classify a speaker by accumulating local decisions made on frames of speech over an entire utterance. The class that has the majority of local decisions is deemed to be the identity of the speaker. This method has proved to be successful for both text-dependant and text-independent speaker recognition.

Most of the preprocessing techniques used here are well known. The major advantage of the current work is the use of decision tree induction. The construction and application of decision trees are significantly faster than other training and classification methods. This is important in speaker recognition since populations and the characteristics of individual speakers change regularly and therefore the classifier must be retrained frequently. Furthermore, classification times are also important since we would like real-time recognition of speakers. With the availability of high-speed Digital Signal Processing chips, preprocessing can be done very quickly, thus the classifier contributes a significant component to the recognition time. Since decision trees can be implemented by simple if-statements, they are extremely fast classifiers.

We believe the methods described here hold great promise for practical application. A current project is aimed at installing an access control system for the AI lab at the University of New South Wales. While the size of population of the lab is roughly the same as the populations sizes reported here, a practical system faces many more problems. The system must operate in real time. The environment cannot highly controlled. There will be many speakers who are not in the sampled population, thus, for a secure area, the rejection rate is perhaps even more important than the recognition rate. A speaker's characteristics vary over time, thus, there is a requirement that the system should be incremental in that new samples may be needed to retrain the system to recognise altered speaker characteritics. And of course, it must be possilbe to deal with arge population sizes.

Despite these problems, we are confident that machine learning is a practical tool for speaker recognition.

Acknowledgments

We thank Donald Michie for suggesting this project and Ross Quinlan for making C4.5 available.

References

Burston, D.K. (1987). Text-dependant speaker verification using vector quantisation source coding. *IEEE Trans. Acoust. Speech and Signal Process.,* **AASP-35**, 133-143.

Markel, J.D. (1972). The SIFT algorithm for fundamental frequency estimation. *IEEE Trans. on Audio and Electroacoustics.*, **AU-20**(5), 367-377.

Matsui, T. and Furui, S. (1992). A text-independent speaker recognition method robust against utterance variations. *IEEE Int. Conf. Acoust. Speech Signal Process.*, **3**, 377-380.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kauffman.

Rosenberg, A.E., Lee, C-H. & Gocken, S. (1992). Connected word talker verification using whole word hidden Markov models. *IEEE Int. Conf. Acoust. Speech Signal Process.*, **3**, 381-384.

Rudasi, L. and Zahorian, S.A. (1992). Text-independent talker identification with neural network. *IEEE Int. Conf. Acoust. Speech Signal Process.*, **3**, 389-392.