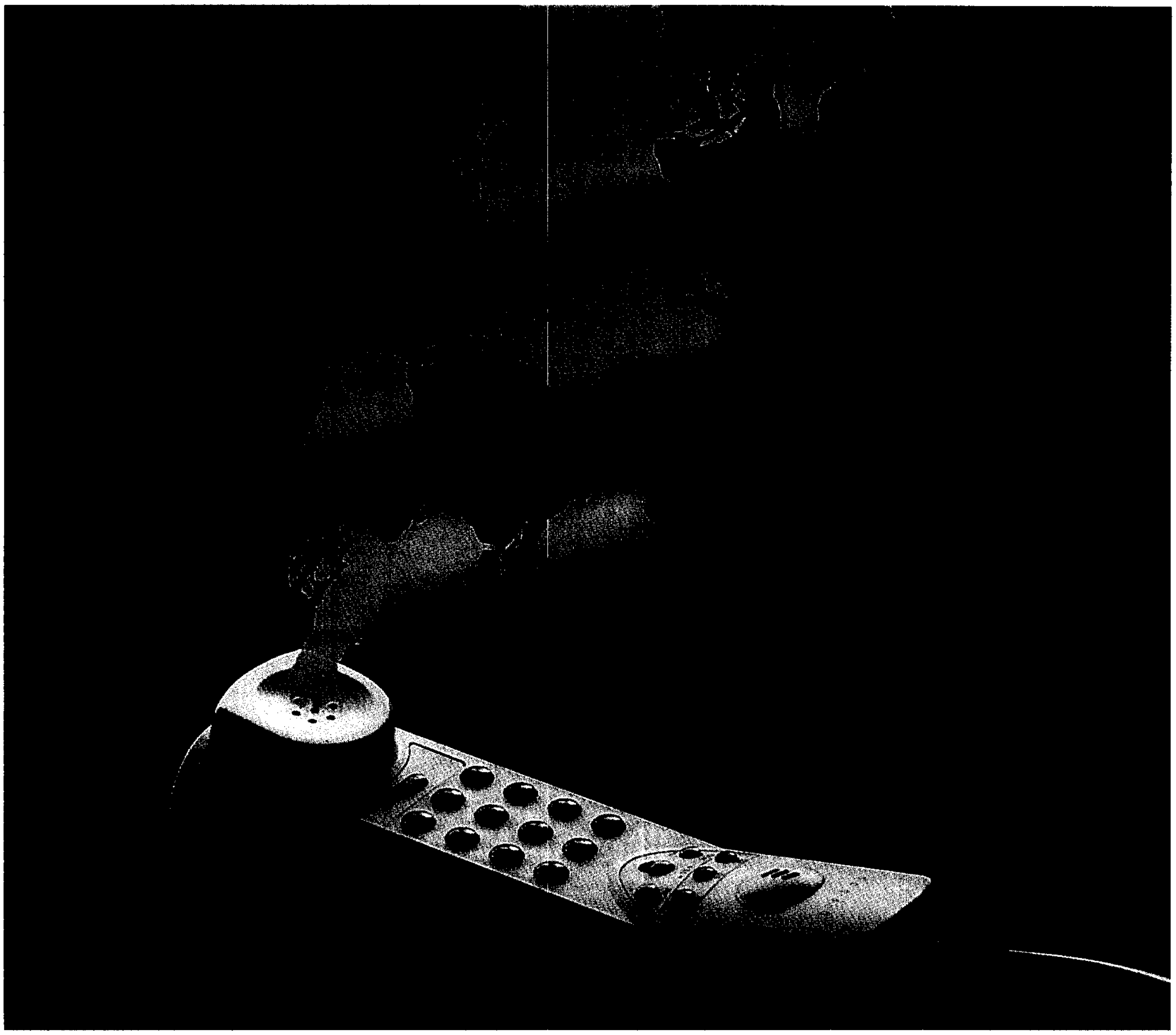


©Telegraph Colour Library/FPG International/Peter Scholey

Text-Independent Speaker Identification

HERBERT GISH and MICHAEL SCHMIDT

The task of automatic speaker identification is to determine the identity of a speaker by machine. In order for humans to recognize voices, the voices must be familiar; similarly for machines. The process of "getting to know" speakers is referred to as training and consists of collecting data from utterances of people to be identified. The second component of speaker identification is testing; namely the task of comparing an unidentified utterance to the training data and making the identification. The speaker of a test utterance is referred to as the *target speaker*. The terms *speaker identification* and *speaker recognition* are used interchangeably.

1053-5888/94/$4.00©1994IEEE

In this article, only text-independent recognition is considered. By *text-independent*, we mean that the identification procedure should work for any text in either training or testing. This is a different problem than *text-dependent* recognition, where the text in both training and testing is the same or is known. In this latter case, knowledge of the words or word sequence can be exploited to improve performance. In some cases text dependency is implicit, e.g., training and testing is done with digit strings although the digit strings may be different in training than in testing.

Speaker recognition can be subdivided into two further categories, closed-set and open-set problems. The closed-set problem is to identify a speaker from a group of N known speakers. Naturally, the larger N is, the more difficult the task. The speaker that scores best on the test utterance is identified. Alternatively, one may want to decide whether the speaker of a test utterance belongs to a group of N known speakers. This is called the open-set problem, since the speaker to be identified may not be one of the N known speakers. If a speaker scores well enough on the basis of a test utterance, then the target speaker is accepted as being known. Though the open-set task involves only a binary decision (accept or reject), it is not necessarily easier than the closed-set problem since it requires that a score be developed that has an absolute meaning; namely, a score that provides a calibrated measure of belief that the target speaker is known. The score is compared to a threshold for purposes of acceptance or rejection. The process of developing a calibrated score is referred to as *score normalization*. While this normalization process is not required for the closed-set problem, we will see that score normalization can play an important role in robust, closed-set, recognition procedures. In addition, this normalization enables the scores from the robust procedures to be used directly for the open set problem. *Speaker verification* is a special case of the open-set problem and refers to the task of deciding whether a speaker is who he or she claims to be. Often, however, speaker verification systems must not only verify the voice, but also the text with a speech recognizer in order to prevent imposters from using recordings. Now, we will focus attention on the closed-set problem.

The speaker recognition task falls under the general problem of pattern classification. Much of this discussion describes maximum *a posteriori* (MAP) probability techniques applied to speaker recognition. This article is not a complete survey of the speaker identification field, nor a research paper, but is intended rather as an introduction to some current work in the field, with particular emphasis on the robust methods developed by the authors. For other introductions to speaker recognition, the reader is referred to [1, 2].

The Technical Challenge

Perhaps the most important technical challenge for applications is dealing with the effects of the communication channel through which speech is received. In many applications, this is a telephone channel. The difficulties do not arise from the existence of a channel *per se*, but rather that in many situations the channel may vary from utterance to utterance. The changing channel effectively creates variability in a speaker's acoustics that far exceeds his/her normal variability. The variability essentially moves speakers about in feature space, and distorts their patterns, which leads to an increased uncertainty in identification. Variability also manifests itself as the occurrence of artifacts such as crosstalk and noise events. Robust methods that will be described can deal with such artifacts.

To date, there are few techniques that deal with variability produced by the filtering effects of the different channels. The usual approach is to employ features that exhibit some degree of invariance to the channel. Gish [3] treats the effect of the channel on a speaker's model probabilistically, leading to a technique which can be useful under some conditions. In this article, we deal with channel effects by utilizing the channel invariance properties of certain features.

Potential Applications

The potential for application of speaker recognition systems exists any time speakers are unknown and their identities are important. In meetings, conferences, or conversations, the technology makes machine identification of participants possible. If used in conjunction with continuous speech recognizers, automatic transcriptions could be produced containing a record of who said what. This capability can serve as the basis for information retrieval technologies from the vast quantities of audio information produced daily. In law enforcement, speaker recognition systems can be used to help identify suspects. Security applications abound. Access to cars, buildings, bank accounts and other services may be voice controlled in the future. Some existing applications use voice in conjunction with other security measures, perhaps a codeword, to provide an extra level of security. You may want to verify that the speaker you are talking to is in fact who he or she claims to be. Systems exist that place telephone calls to check that a speaker is where he or she is supposed to be. The technology has applications to human-machine interfaces, where intelligent machines would be programmed to adapt and respond to the current user. Speaker identification also has applications to other voice technologies. Speech-recognition systems can usefully employ speaker-recognition technology. Gender recognition, based on a variant of speaker-recognition techniques, is already in use in many speaker-independent speech recognizers to improve performance. The above list is by no means complete, but provides an indication of the types and variety of applications.

Feature Selection

Speech exhibits significant variation from instance to instance for the same speaker and text. From the point of view of text-independent speaker identification, a speaker produces a stream of speech features (features will be discussed in greater detail below). These features characterize both the speech as well as the speaker. For more than a few seconds of speech, we expect the features to fill feature space in a way that depends primarily on the speaker and not the particular text spoken. The assumption is that with sufficient speech, a good representation of the sounds that a speaker can create will be observed. The goal is to obtain descriptions or models of a speaker's patterns in feature space which can be used to identify the speaker of a test utterance.

An important step in the speaker identification process is to extract sufficient information for good discrimination, and, at the same time, to have captured the information in a form and size that is amenable to effective modeling. The amount of data generated by short utterances is quite large. Speech is usually digitized at a rate of 8 kHz or higher, using 8 bits or more per sample, requiring tens of thousands of bytes for a few seconds. Whereas these large amounts of information are needed to characterize the speech waveform, the essential characteristics of the speech process changes relatively slowly, permitting a representation requiring significantly less data. Speech signals can be parametrized over relatively long time periods of 10 to 25 ms called frames. If the speech from a 20 ms frame can be reduced to a 14 dimensional vector, say, then a data reduction ratio of 11.4 = 160/14 is achieved at the 8 kHz sampling rate. The process of reducing data while retaining classification information falls under the general heading of feature extraction. The vectors extracted are termed *features*. The *n*-dimensional feature space is referred to as *speaker space*.

Feature vectors produced by individual speakers are often assumed to be samples from a continuous density probability distribution. The distributions of different speakers overlap and share speaker space, but are ideally distinguishable from each other so that speaker identification can be achieved. Speaker identification is accomplished by determining how and where voices "spend their time" in speaker space. To simplify matters, it is usually assumed that the feature vectors are independent of one another, even though vectors from consecutive frames are correlated in reality.

Cepstral Features

Speech information is primarily conveyed by the short-time spectrum, the spectral information contained in about a 20 ms time period [4, 5]. While the short-term spectra does not completely characterize the speech production process, the information carried by it is basic to many speech processing activities, including both speaker recognition and speech recognition. There are a variety of methods for parameterizing short-term spectrum. The characterization of choice, also for a variety of applications, are mel-warped cepstra. We describe the basic elements of this computation below and also discuss some relevant properties.

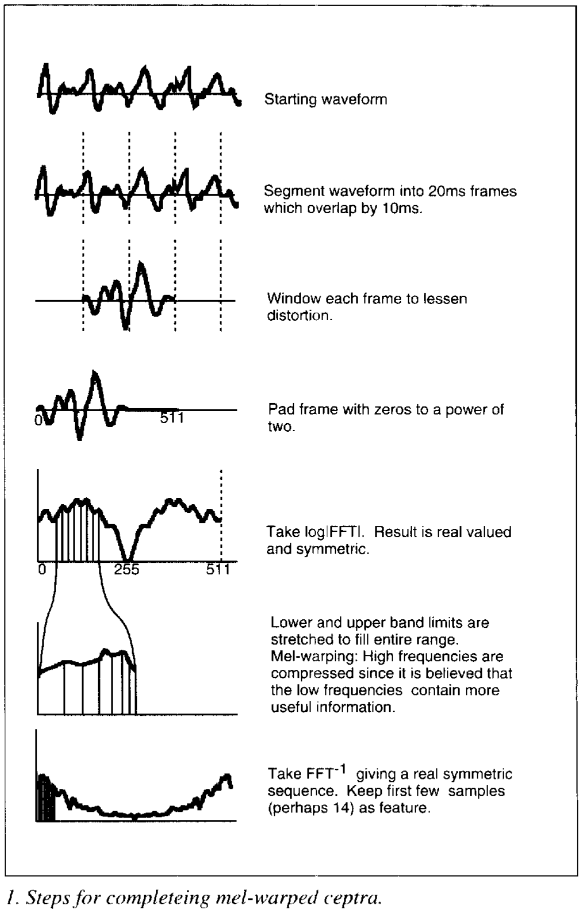
The first step of many feature generation algorithms is to move the speech signal to the frequency domain via a fast Fourier transform (FFT). The cepstrum is computed by taking the FFT inverse of the log magnitude of the FFT:

$$\text{cepstrum (frame)} = \text{FFT}^{-1} (\log |\text{FFT (frame)}|)$$

The inverse Fourier Transform and Fourier transform are identical to within a multiplicative constant since log|FFT|is real and symmetric; hence the cepstrum can be considered the spectrum of the log spectrum. The mel-warped cepstrum is obtained by inserting the intermediate step of transforming the frequency scale to place less emphasis on high frequencies before taking the inverse FFT. The mel scale is based on the non-linear human perception of the frequency of sounds [4, 6]. The first few, perhaps 14, low cepstral coefficients of the cepstrum are retained (Fig. 1.)

Some justification for the use of cepstra is in order. To start, cepstra are used to get at the shape of the vocal tract. A bare-bones view of voiced speech production is that air is forced through the vocal chords producing periodic pulses which are subsequently filtered by the shape of the vocal tract [4, 7]. Hence voiced speech is sometimes modeled as a quasi-periodic pulse (periodic over individual frames) followed by a linear time-invariant (LTI) filter. Let $s(t)$ denote the speech signal, $h(t)$ the impulse response of the LTI filter and v(t) the periodic pulse signal. Then in the frequency domain,

$$S(f) = V(f)H(f)$$


Starting waveform
Segment waveform into 20ms frames which overlap by 10ms.
Window each frame to lessen distortion.
0
511
Pad frame with zeros to a power of two.
0
255
511
Take log|FFT|. Result is real valued and symmetric.
Lower and upper band limits are stretched to fill entire range. Mel-warping: High frequencies are compressed since it is believed that the low frequencies contain more useful information.
Take FFT-1 giving a real symmetric sequence. Keep first few samples (perhaps 14) as feature.

1. Steps for completeing mel-warped ceptra.

where S, V, H denote the respective Fourier Transforms. By taking logarithms of both sides, a desirable separation of the periodic and vocal tract components is achieved,

$$\log S(f) = \log V(f) + \log H(f)$$

A further Fourier transform preserves the above sum. If f_0 is the frequency of the periodic pulse, the pitch period, then log V(f) has peaks at integer multiples of f_0. These peaks translate into a relatively steep bump in the cepstral domain. The overall shape of the log spectrum, i.e., the spectral envelope, on the other hand, is described by low cepstral values. The cepstrum is a vehicle for separating the much less informative pitch information from the more important vocal tract shape information [4, 7, 8]. The above procedure for separating the input signal from an LTI filter falls under the heading of "homomorphic deconvolution" in the literature. Additionally, homomorphic deconvolution can be used to remove linear time-invariant channel effects. The cepstrum component due to such channel effects should be constant and can be subtracted out. Another benefit of using cepstra is that they can be reasonably modeled by multivariate Gaussian distributions, a nice property for reasons to be discussed below. Finally, and perhaps most importantly, cepstra work well experimentally.

Another feature often found in the literature is the *linear prediction coefficient* (LPC) cepstrum [4]. The LPC cepstrum is the cepstrum of the autocorrelation sequence of a speech frame. The LPC cepstrum is computationally less expensive, but less effective in our experiments.

Methods for Speaker Identification

Previous Approaches

In this article we emphasize current approaches to the text-independent problem with emphasis on the authors' techniques. All of the methods discussed have an underlying basis in probabilistic modeling of the features of the speakers. Prior to the development of probabilistic algorithms, methods based on template matching were employed and can still be useful under constrained circumstances. By template matching, we mean the comparison of an average computed on test data to a collection of stored averages developed for each of the speakers in training [9].

Also termed statistical feature averaging, the template matching approach employs the mean of some feature over a relatively long utterance to distinguish among speakers. For text-independent recognition, ideally one has utterances of several seconds or minutes in order to ensure that a voice is modeled by mean features of a broad range of sounds, rather than by a particular sound or phone. Test utterances are compared to training templates by the distance between feature means. All variations to the technique arise from the choices of features vectors and distance metrics.

Several metrics can be used for minimum distance classifiers of which the Euclidean is the best known and one of the easiest to compute. If one assumes that the training features have mean μ and covariance Σ, then an alternative is the Mahalonobis distance r with

$$r^2 = (\bar{x} - \mu)'\Sigma^{-1}(\bar{x} - \mu)$$

where $\bar{x}$ is the average of the feature vectors in the test. The prime indicates the vector transpose. Points of equal Mahalonobis distance from μ form a hyperellipsoid centered at μ. The principle axes and the lengths of the principle axes of the hyperellipsoids are determined by the eigenvectors and eigenvalues, respectively, of the covariance matrix Σ [10]. The Mahalonobis distances depend on the direction of x from μ. The sample or estimated covariance of a collection of samples $\{x_i\}$ is given by

$$S = \frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})'$$

In the future, the distinction between true speaker means, μ, and covariances, Σ, and sample means and covariances generated from training sessions is not made. The symbols $\bar{x}$ and S are used to denote the means and covariances of a test utterances.

Results using straightforward statistical feature averaging are usually suboptimal. The method is particularly sensitive to variations in channel and to background noise, both of which can alter the features, resulting in shifted means.

Probabilistic Modeling of Speakers

Probablistic modeling of speakers refers to modeling speakers by probability distributions rather than by average features and to basing classification decisions on probabilities or likelihoods rather than distances to average features. Assuming that the distributions for speakers are known and have continuous densities p_i then the likelihood that a feature x is generated by the i^{th} speaker is $p_i(x)$. Using Bayes' rule, the probability that the speaker is the i^{th} speaker is

$$P(\text{speaker} = i|x) = \frac{p_i(x)P_i}{p(x)} \qquad (1)$$

where P_i is the prior probability that the utterance came from the i^{th} speaker, and $p(x)$ is the probability of the feature x occurring from any speaker. Typically the prior probabilities for each of the speakers are assumed equal. The term $p(x)$ is the average of the speaker densities,

$$p(x) = \sum_{i=1}^{I} p_i(x)P_i$$

where I is the number of speakers. Note that $p(x)$ is the same for all speakers and if the prior probabilities are equal, the speaker to choose will simply depend on which speaker has the highest likelihood. This will be the most probable speaker given the observed feature, and is known to result in the minimum error strategy [11]. Schwartz *et al.* [12] were first to apply probabilistic modeling to the speaker identification task.

The likelihood of a sequence of independent samples $X = \{x_j\}$ is $\Pi_j\, p(x_j)$. It follows trivially that the log likelihood of X is $\ell(X) = \Sigma_j\, ln\, p(x_j)$. Since our only use for log likelihoods is to identify speakers by comparing values, any term of the likelihood which is constant over all training sessions can be neglected.

The Bayes equation, Eq. 1, can be viewed as the comparison of the likelihood of the feature for a particular speaker compared to the average likelihood for the observed feature. Thus $p(x)$ can be viewed as a means of normalizing the likelihood of $p_i(x)P_i$ and converting it into a probability that has an absolute meaning. While we have argued that such normalization is not ordinarily useful for the closed-set problem, it can be quite useful when employing robust identification procedures. The method of normalization we employ differs from that suggested by the Bayes equation but the idea is that normalizing likelihoods enables us to evaluate information and adjust the way we utilize it. We have more to say about this issue later.

Nonparametric versus Parametric Models

The distinction between parametric and nonparametric probability models is an important way of dichotomizing the space of such models. Models which assume a structure characterized by parameters are termed parametric. In nonparametric modeling, minimal assumptions regarding the probability density function are made.

The models that do assume some structure are referred to as parametric models since the structure is characterized by a collection of parameters. In bringing structure to a multivariate probability density function we have by definition limited the form which it can take. The more limited the form, the fewer data that are needed to specify the density. This characteristic has both its positive and negative aspects. The negative aspects stem from having a too restrictive structure, one which may not be adequate to the modeling task. The positive aspects derive from the succinctness of the representation. This manifests itself in a variety of ways. One way is that it permits efficient use of the data in estimating the models. This efficiency in use of data can also extends to the evaluation of test data through the use of statistical summaries of the data rather than the data itself. Another major feature of parametric models is that it becomes possible to model and understand changes in the data through changes in the parameters. This can be important and very useful when trying to understand distortions that occur to data and develop means for compensating for these distortions.

Nonparametric Methods

Nearest Neighbor and Vector Quantization Modeling

The nearest neighbor method for estimating the density from a sample $R = \{r_i\}$ at point x is to measure the distance between and the point in the sample closest to x, x's nearest neighbor:

$$d_{N\,N}(x,R) = \min_{r_j \in R} |x - r_j|$$

Intuitively, the idea is that the smaller the nearest neighbor distance, the higher the density. Specifically,

$$\hat{p}(x) = \frac{1}{V_n\,(d_{NN}\,(x,R))}$$

where $V_n(\rho)$ is the volume of sphere in n-dimensional space with radius ρ. Taking logarithms of both sides and recalling that the volume of an n-dimensional sphere with radius ρ is proportional to ρ^n, we get

$$\ln \hat{p}(x) \approx -\,n\,\ln(d_{NN}(x,\,R))$$

Let $U = \{u_i\}$ and $R = \{r_i\}$ denote the collections of feature vectors extracted from test and reference utterances, respectively. R is used to estimate the speaker's density. So In $\hat{p}(u_i) \approx -\,nd_{NN}\,(u_i,R)$ and

$$\hat{\ell}(U) \approx -\sum_{u_i \in U} \ln d_{NN}(u_i,R) \tag{2}$$

The speaker with the greatest log likelihood, equivalently, the speaker whose reference model R is closest to the test reference U, is identified.

Higgens, *et al.*, [13] use a modified normalized nearest

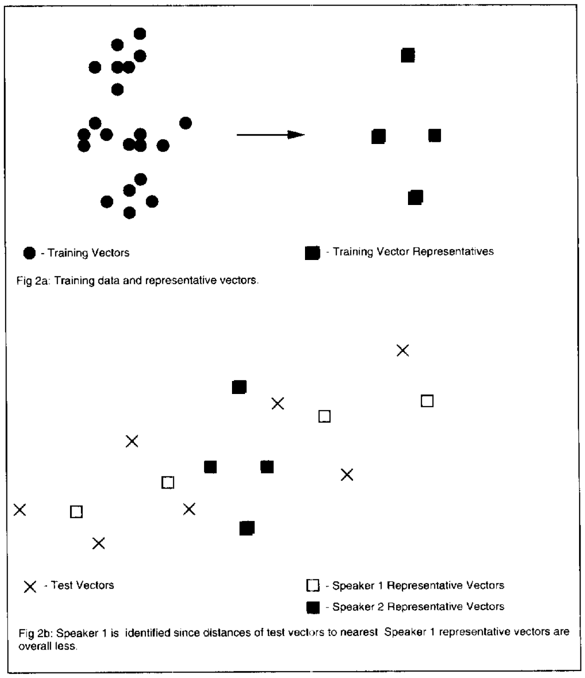

● - Training Vectors
■ - Training Vector Representatives
× - Test Vectors
□ - Speaker 1 Representative Vectors
■ - Speaker 2 Representative Vectors

Fig 2a: Training data and representative vectors.

Fig 2b: Speaker 1 is identified since distances of test vectors to nearest Speaker 1 representative vectors are overall less.

2. Vector quantization modeling.

neighbor distance measure. They define a symmetric difference between a test utterance U and reference utterance R as follows:

$$D(U,R) = \frac{1}{|U|}\sum_{u_i \in U} d_{NN}^2(u_i,R) + \frac{1}{|R|}\sum_{r_j \in R} d_{NN}^2(r_j,U) - \frac{1}{|U|}\sum_{u_i \in U} d_{NN}^2(u_i,U-\{u_i\}) - \frac{1}{|R|}\sum_{r_j \in R} d_{NN}^2(r_j,U-\{r_j\})$$

The first term in the sum looks suspiciously like the log likelihood except that instead of log distances, distances squared are accumulated which experimentally gives better results. The second term is to symmetrize the distance making $D(U,R) = D(R,U)$. The last two terms are measures of spread of test and reference utterances respectively and are present so that utterances with large variances are not unduly penalized. Finally, the terms are normalized to compensate for differences in cardinality between U and R by the $1/|U|$ and $1/|R|$ factors, giving average nearest neighbor differences.

Higgens [13] found this modification to the nearest neighbor approach was more effective than the use of the conventional nearest neighbor method, when applied to their problem. While it was not evident why this was the case, it may very well be related to the lack of robustness of Eq. 2 to outliers. For example, if a single feature vector of the unknown test falls very close to a reference feature vector, the total log likelihood will be very large regardless of the lack of match of the remainder of the test data. Current approaches to robust estimation replace the logarithm in log likelihoods with functions that that do not let the contribution of any single term to the log likelihood become excessive [14].

Model training consists only of collecting reference feature vectors. The probability distribution of a speaker's voice is described by the collection of samples over training utterances. No further modeling is attempted; the samples are themselves the model. However, in order to reduce significant memory and computational requirements in testing, training frames are sequentially examined and discarded by Higgens if they are not within a preset threshold of previously kept frames. By picking representative training frames, the approach is technically one of vector quantization modeling.

Vector quantization (VQ) modeling constructs representatives of the data. VQ modeling is identical to nearest neighbor modeling except that distances to nearest data *representatives* are measured. The need to reduce the computation and memory demands of the nearest neighbor approach is a chief motivation behind VQ modeling.

In Fig. 2a, the process of going from the training data to a representative data set is shown, and in Fig. 2b the classification of a test set is illustrated. We emphasize that the figures are strictly for illustrative purposes. In practice the features have dimension greater than two (nominally 14) and hundreds of vectors are employed for characterizing the speaker.

Selecting the data representatives can be approached as a problem of grouping the training feature vectors into clusters. All vectors falling inside a cluster are represented by a centroid, perhaps the cluster mean or a member of the cluster. The feature space is quantized by mapping every vector to one of the cluster centroids. Ideally clusters have the property that the average distance of vectors from their nearest centroid is minimized. Dozens of clustering algorithms exist. The k-means algorithm [15, 16] is often described in pattern recognition literature, but depends on obtaining reasonable starting estimates. Techniques for selecting initial estimates range from choosing random representatives to highly skilled art forms.

VQ uses centroids to estimate the modes of a probability distribution. Heuristically it is possible that each of the VQ clusters model particular speech components, perhaps nasals or vowels. The clusters, however, create unrealistically rigid boundaries in the sense there can be no overlap in the features generated by two different acoustic classes. Each vector belongs to one and only one cluster. The reader is referred to work by Soong *et al.* [17] for an example of VQ applied to speaker recognition.

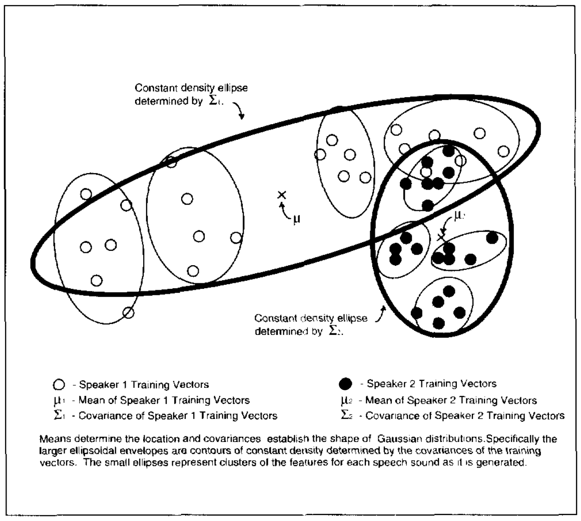
Parametric Models

Gaussian Model

The Gaussian model is a basic parametric model that has merit by itself and can, as we shall see below, be the basis of other, more sophisticated models, including robust models. Before discussing some of the properties of the Gaussian distribution let us first consider the likelihood of a collection of feature vectors with this distribution.

The likelihood of a test utterance consisting of n independent cepstral vectors, $X = \{x_1, ..., x_n\}$ for a Gaussian model with parameters μ and Σ is given by the density,

$$L(X;\mu,\Sigma) = |2\pi\Sigma|^{-n/2} \exp\left\{ -\frac{1}{2}\sum_{i=1}^{n} (x_i-\mu)' \Sigma^{-1}(x_i-\mu) \right\}$$


Constant density ellipse determined by Σ_1.
μ
μ_2
Constant density ellipse determined by Σ_2.
○ - Speaker 1 Training Vectors
μ_1 - Mean of Speaker 1 Training Vectors
Σ_1 - Covariance of Speaker 1 Training Vectors
● - Speaker 2 Training Vectors
μ_2 - Mean of Speaker 2 Training Vectors
Σ_2 - Covariance of Speaker 2 Training Vectors
Means determine the location and covariances establish the shape of Gaussian distributions.Specifically the larger ellipsoidal envelopes are contours of constant density determined by the covariances of the training vectors. The small ellipses represent clusters of the features for each speech sound as it is generated.

3. Gaussian modeling.

The vertical bars indicate a determinant, $|2\pi\Sigma|$ is the determinant of $2\pi\Sigma$. It is often convenient computationally to use log likelihoods,

$$\ell(X;\mu,\Sigma) = \log L(X;\mu,\Sigma) =$$
$$-\frac{n}{2}\log|2\pi\Sigma| - \frac{1}{2}\sum_{i=1} n(x_i - \mu)'\Sigma^{-1}(x_i - \mu)$$

Equivalently,

$$\ell(X;\mu,\Sigma) = \log L(X;\mu,\Sigma) =$$
$$-\frac{n}{2}\log|2\pi\Sigma| - \frac{n}{2}\,\mathrm{tr}\,(\Sigma^{-1}S) - \frac{n}{2}(\bar{x} - \mu)'\Sigma^{-1}(\bar{x} - \mu)$$

where S and $\bar{x}$ are the covariance and mean of the test utterance ([18] p. 97). If the covariances of the training utterances are equal, then classification based on the log likelihoods reduces to the minimum distance method using Mahalonobis distances.

A simple, but also effective model, the multivariate Gaussian distribution is completely characterized or parameterized by its mean μ and covariance matrix Σ. The likelihood or probability that a particular utterance was generated by a given model can be computed just on the basis of the sample mean $\bar{x}$, and sample covariance S, of the observations.

The large ellipses of Fig. 3 show contours of constant density around the means of the Gaussian models of two speakers and are determined by the eigenvectors and eigenvalues of the covariance matrices. (The figure is strictly for illustrative purposes.) The small ellipsoidal shapes represent clusters of the features for each speech sound as it is generated. As time progresses one cluster after another is generated. The sample means are simply the centroids of all these clusters and the sample covariances are represented by the ellipsoidal envelopes that encompasses the small clusters of features. The log likelihood function above is a probabilistic method for measuring the match between $\bar{x}$ and μ and between S and Σ

The Gaussian models are somewhat crude in that they model only the gross characteristics of the speaker's distribution. The Gaussian mixture model, discussed below, is an attempt to model the smaller clusters of speech. However, it is worthwhile to note that attempts to model detail which may be lost due to changes and distortion in the channel can lead to non-robust performance. The robust approach also attempts to capture detail, but by using multiple simple models rather than a single complicated fragile model.

Training Gaussian models is fast and relatively straightforward. The sufficient statistics μ and Σ are estimated from reference data using the corresponding sample statistics. For a 14-dimensional Gaussian distribution 105 covariance and 14 mean parameters must be estimated. At a rate of 100 frames per second, using 14-dimensional features, 1400 scalars per second are generated to estimate the 119 Gaussian parameters. In order to obtain a reasonable model, it is necessary to use at least 10 seconds of training speech. Computing the determinant and inverse of Σ is also part of training.

Mean removal refers to the process of shifting the cepstra to have zero means and is often used as a method of compensating for channel differences. Identifications are then based on the shape of the cepstra rather than on location. The cepstral means, however, contain both channel and speaker information. By removing the means, potentially useful speech information may be lost. In Figure 2b the means are removed whereas in Figure 3 means are retained.

Mixture Models

A mixture model is a weighted sum of densities

$$p(x) = \sum_{i=1}^{Q} P(w_i)p(x|w_i),$$

where the $P(w_i)$ are weights or prior probabilities of the terms $p(x|w_i)$. The $P(w_i)$add to unity ensuring that the mixture is a proper density. Generating an observation from a mixture model can be accomplished in two steps; first pick the density term according to the priors, then generate an observation from the chosen density. Typically Gaussian distributions are used as the mixture terms in which case the model is completely specified by the weight, mean and covariance of each term. The terms of mixture models for two speakers can be represented by the two sets of four smaller ellipses of Figure 3.

Training is accomplished via the Estimate Maximize (EM) algorithm [19]. The number of terms, k, in the mixture model must be determined in advance. Initially all frames are partitioned into k clusters, either randomly, via some clustering algorithm or perhaps by an automatic speech segmenter. An initial model can be obtained by estimating the parameters from the clustered frames. The proportion of feature vectors in each cluster gives the prior weights; means and covariances are estimated from the vectors in each cluster. This is the "estimate" step. The feature vectors can now be reclustered by choosing the term with the maximum likelihood from the estimated mixture model. This is the "maximize" step. The process is iterated until the model parameters converge. The EM algorithm guarantees that the likelihood of the feature vectors converges to a local maximum [19]. As above, likelihoods are used to identify speakers.

Mixture modeling is similar to VQ identification in that voices are modeled by components or clusters. One school of thought is to have as many model components as speech sounds (phonemes), typically about 50. The acoustic components are learned in training and do not need to be known *a priori*. Reynolds [20] applies mixture models to the speaker identification task.

Modified Gaussian Models

In the Gaussian model the likelihood of the data is represented in terms of statistics $\bar{x}$ and S. By the modified Gaussian model we mean, in general, a score that is a linear combination of the log likelihoods of statistics of the data. In particular we employ the log-likelihoods of the sample mean and covariance, as given above, as well as the sample covariance of the derivatives of the cepstra which we denote by $S_\triangle$.

First consider two statistic likelihoods, the log likelihoods of the mean and covariance: $\ell(\bar{x};\mu,\Sigma)$ and $\ell(S;\Sigma)$. Since the x_i are assumed $N_p(\mu,\Sigma)$ where $N_p(\mu,\Sigma)$ denotes a p-dimensional Gaussian distribution with mean μ and covariance matrix Σ, it follows that $\bar{x}$ has a $N_p(\mu,n^{-1}\Sigma)$ distribution. Hence

$$\ell(\bar{x};\mu,\Sigma) = -\frac{1}{2}\log|2\pi n^{-1}\Sigma| - \frac{n}{2}(\bar{x}-\mu)'\,\Sigma^{-1}(\bar{x}-\mu).$$

Suppose $M\,(m \times p)$is a matrix with m independent $N_p(0,\Sigma)$ distributed rows. The Wishart distribution is defined as the distribution of the square matrix $M'M_{(p\times p)}$ and is said to have scale matrix Σ and m degrees of freedom, $W_p(\Sigma, m)$ [21] . If $V\sim W_p(\Sigma,n)$ then the density of V is

$$p(V) = \frac{c_{n,p}|V|^{(n-p-1)/2}}{|\Sigma|^{n/2}}\exp\left\{-\frac{1}{2}\mathrm{tr}(\Sigma^{-1}V)\right\}$$

with $c_{n,p}$ depending only on n and p ([21] p. 245). It follows that $S\sim W_p(\Sigma/n,n-1)$. The log likelihood of S can be rewritten as

$$\ell(S;\Sigma) = -\frac{n-1}{2}\log|\Sigma/N| - \frac{n}{2}tr(\Sigma^{-1}S) + C_{S,n,p}$$

The last term in the sum can be treated as a constant since S and n depend only on the test utterance and so can be ignored when comparing the likelihoods of the training models.

Notice that the log likelihood of all the data, $\ell(X;\mu,\Sigma) = \ell(\bar{x};\mu,\Sigma) + \ell(S;\Sigma) + C_{S,n,p}$. The log likelihood of an utterance can be separated into the log likelihoods of $\bar{x}$ and S plus a constant. The sample mean and covariance are sufficient statistics for the log likelihood. Training is identical to that of standard Gaussian modeling.

If the data were truly independent and Gaussian, then $\bar{x}$ and S would be sufficient to evaluate the likelihood. Since, as we have noted, this is not the case, we employ cepstral derivatives to capture some of the dependencies that exist in the data. Our cepstral derivatives are actually the coefficients of a linear fit to 5 frames of cepstra. These derivative cepstra are characterized by their covariance $S_\triangle$. The mean of the difference cepstra $\bar{x}_\triangle$depends only on the starting and ending frames of an utterance and so is not useful.

We call the log likelihoods of $\bar{x}$, S, and $S_\triangle$the statistic scores and abbreviate these scores as *mean*, *cov* and *dcov*. These statistic scores can be weighted and combined to form new scores,

$$\alpha\,\ell\,(\bar{x};\mu,\Sigma) + \beta\ell(S;\Sigma) + \gamma\ell(S_\triangle;\Sigma_\triangle).$$

Note that $\ell(X,\mu,\Sigma)$, the log likelihood of a test utterance assuming a Gaussian model, is a special case of the combined score when $\alpha = \beta,\gamma = 0$. The *baseline* score is defined as the statistic scores combined with equal weights, $\alpha = \beta = \gamma$.

The covariances are invariant to linear time-invariant (LTI) channels. Such channels have the effect of adding a constant vector to the cepstra, thereby adding an offset to the mean, but leaving the covariance unaffected. The covariance scores are less susceptible to changes in handset or channel.

Robust Speaker Identification

In performing speaker identification we will often have a signal that has been corrupted in a variety of ways. These distortions to the speaker's signal can result in significantly worse identification performance. The types of corruption that we will focus on are those phenomena that can be localized in time. This will include spurious noise activity, crosstalk, uncharacteristic speech sounds from the target speaker, and also speech from other speakers in a conversation with multiple participants.

These robustness issues assume that the underlying models are satisfactory in the absence of time-localized anomalies. When the models themselves are a problem, other robustness approaches are available. See, for example, [22].

The key elements of our robust approach are based on the computation of the of speaker scores over relatively short time intervals. The use of the intervals enables isolation of the anomalous events. In addition to examining the scores over relatively short time segments, the scores themselves must be normalized in order to make comparisons between segment scores meaningful. It is these normalized-segment scores that enables our robust procedures to be effective; it is the selective combining of these scores that makes the procedure robust.

The above principles can be applied to any of the classes of recognizers that have already been discussed. We will base our robust system on the modified Gaussian approach. That is, we will create models for segment statistics, namely for the segment means and covariances. In addition we will create a multiplicity of models for each speaker. Below we give additional details.

Multiple Training Models

Naturally, recognition results will be best when training conditions match testing conditions. Often, however, testing and training takes place on different channels and in different environments. By training with multiple sessions under multiple conditions, the hope is that conditions from at least one of the sessions will be close to those of testing, thereby making the system more robust. Given multiple training sessions, it is advantageous to build separate models from each session rather than one model from all sessions combined. A single training session which is very different from testing may corrupt a combined model.

Conditions may even change within a training session. Experimentally, our results improve when we build two 30 second models from each 60 second training session. Using segments shorter than 20 seconds for training tends to be counterproductive. The shorter time span may not provide enough data to reliably estimate the numerous parameters for each model. (One-hundred five covariance parameters must be estimated when using 14 cepstral coefficients and neighboring cepstra are independent.)

A potential hazard of multiple models for each speaker is that there are not only more models for the true speaker, but for non-target speakers as well. Hence there are more chances for an improper match. Experience has shown, however, that it is extremely unlikely for non-target speaker models to achieve high log likelihood scores. Usually when a speaker is misidentified it is not due to a non-target speaker doing well, but rather to the true speaker's models doing poorly.

Segmentation and Normalization

Observe that if we decompose a segment X which is a collection of N frames, into K segments of size n_k where $\Sigma_{k=1}^{K} n_k = N$, and let log L_k denote the log likelihood of the k^{th} segment, then

$$\log L = \sum_{k=1}^{K} \log L_k$$

That is, the log likelihood of the full segment is, because of the frame independence assumption, the log likelihood of the smaller segments.

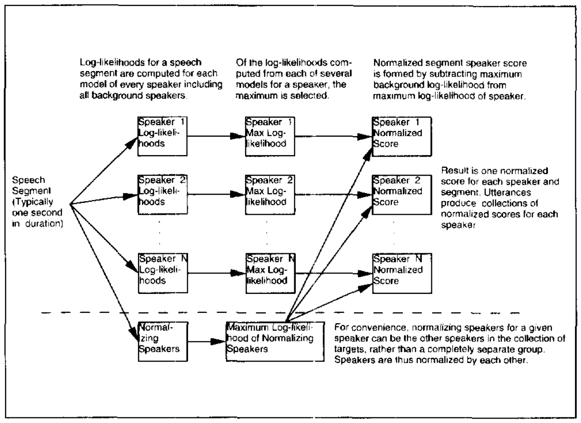
The first step is to extract segments from the test utterance. A direct approach is to uniformly chop the dialog into segments of some arbitrary size, with the idea that a subset of these segments will be pure enough to be recognized. The alternative is to non-uniformly segment speech based on the speech signal.

The importance of representing the log likelihood as the sum of smaller segments is that it enables us to generalize the scoring of a speaker's utterance in several important ways. The first aspect is our ability to select the best model from the collection of models for each speaker for each of the different segments. Denote the models belonging to speaker i by z_j. We define the likelihood of an utterance in terms of its subsegments as

$$\log L^{*} = \sum_{k=1}^{K} \max_{j} \log L_k(z_j)$$

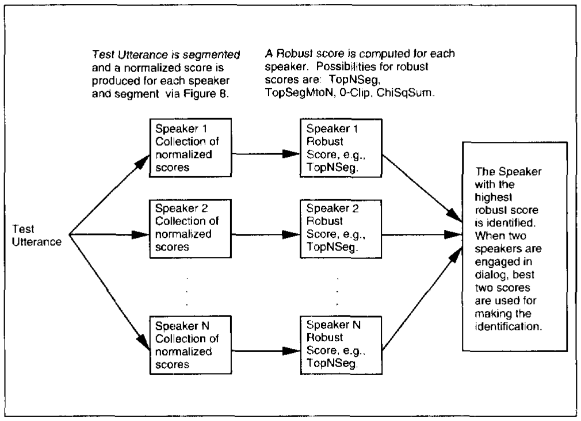
where $L_k(z_j)$ is the likelihood of the k^{th} segment employing the model z_j. The case in which all segments are constrained to use a single maximum likelihood model is equivalent to the case of all the data being in one large segment.

The second reason for evaluating multiple segments is that partitioning an utterance enables us to discard or deemphasize segments contaminated by other speakers and noise. Scoring procedures designed to take advantage of the segmentation are described in the next section. The segmentation and robust scoring methods also makes possible the identification of speakers engaged in dialog by choosing speakers with the highest scores.



Log-likelihoods for a speech segment are computed for each model of every speaker including all background speakers.
Of the log-likelihoods computed from each of several models for a speaker, the maximum is selected.
Normalized segment speaker score is formed by subtracting maximum background log-likelihood from maximum log-likelihood of speaker.
Speech Segment (Typically one second in duration)
Speaker 1 Log-likelihoods
Speaker 2 Log-likelihoods
Speaker N Log-likelihoods
Speaker 1 Max Log-likelihood
Speaker 2 Max Log-likelihood
Speaker N Max Log-likelihood
Speaker 1 Normalized Score
Speaker 2 Normalized Score
Speaker N Normalized Score
Result is one normalized score for each speaker and segment. Utterances produce collections of normalized scores for each speaker
Normalizing Speakers
Maximum Log-likelihood of Normalizing Speakers
For convenience, normalizing speakers for a given speaker can be the other speakers in the collection of targets, rather than a completely separate group. Speakers are thus normalized by each other.

4. Normalized speaker scores.



Test Utterance is segmented and a normalized score is produced for each speaker and segment via Figure 8.
A Robust score is computed for each speaker. Possibilities for robust scores are: TopNSeg, TopSegMtoN, 0-Clip, ChiSqSum.
Test Utterance
Speaker 1 Collection of normalized scores
Speaker 2 Collection of normalized scores
Speaker N Collection of normalized scores
Speaker 1 Robust Score, e.g., TopNSeg.
Speaker 2 Robust Score, e.g., TopNSeg.
Speaker N Robust Score, e.g., TopNSeg.
The Speaker with the highest robust score is identified. When two speakers are engaged in dialog, best two scores are used for making the identification.

5. Robust scoring.

The above log likelihoods provide us with the unnormalized scores for each of the three statistics that we are currently using, the mean, covariance and the covariance of the derivative cepstra. These scores require normalization. Recall that the log likelihood of the data for a segment is obtained by evaluating a probability model for the data in the segment; other segments have different data scored with different models and therefore there is no basis for comparison of likelihoods from different segments.

The normalization, obtained by comparing the likelihood of the data for different models, is designed to make comparisons of scores between segments meaningful. When the k^{th} segment, say, from a test utterance is observed, three different likelihood ratios are evaluated, one for each of the models for the three statistics. As an example, for the sample mean, $\bar{x}$, of the cepstra in the segment we compute,

$$\lambda(\bar{x})_{\mu,i,k} = \log \frac{\max_j p_\mu(\bar{x};\theta_{i,j})}{\max_{j,i'\neq i} p_\mu(\bar{x};\theta_{i',j})}$$

which is a generalized log likelihood ratio, for discriminating between speaker i and the speakers in the reference set on the basis of the sample mean of the cepstra in the k^{th} segment of the utterance. The numerator is the maximum likelihood of all models for the mean belonging to speaker i, and the denominator is the maximum over all models not belonging to speaker i.

The normalized scores for the j^{th} reference speaker are computed by subtracting the best log likelihood score from all speakers except the j^{th} speaker from the best scores of the j^{th} speaker. A speaker's scores are normalized by all other speakers' scores in the reference collection. A collection of speakers must be available to perform the normalization. The set of reference speakers can be comprised of the members of the closed set who are not the current target speaker. If the size of the closed set is small, it may be wise to use additional out-of-set speakers for normalization purposes (Figure 4). We note in passing that the reference set can include models of acoustic events other than speech. This can be very useful in applications in which specific interference signals are prevalent and can be modeled.

The actual value taken on by this log likelihood ratio represents the degree of evidence in favor of the i^{th} speaker. In like manner, the analogous likelihood ratios for the observed sample covariance of the cepstra, S, and the covariance of the differential cepstra, $S_{\triangle}$ are obtained from the observed segment. We refer to the log likelihood ratios as the normalized scores. The unnormalized score is simply the log of the numerator of the likelihood ratio. See [23].

Robust Scoring Methods

Several robust scoring procedures which take advantage of multiple models and/or segmentation are now presented (Fig. 5). In addition we will describe some nonrobust scoring methods for comparison. The robust scoring methods considered are of three types, (1) those which use a subset of the segmental scores, (2) those which modify scores from all the segments, and (3) a method which replaces the score for a segment with a measure of confidence that the segment was spoken by the target speaker. The first two approaches are based on robust methods for dealing with observations called R and M estimates respectively [14]. The third approach is new.

The **NoSeg** score is defined as the score of an entire test utterance of the best model from each speaker. Equivalently, NoSeg can be defined as the sum of the unnormalized scores of the best model over all segments as was noted above. The NoSeg scoring is equivalent to generalized Gaussian model scoring. The **SumSeg** score is the sum over all segments of each speaker's normalized score. The **TopNSeg** scoring method is to sum the top N ranked segment scores for each speaker, with N an integer less than the number of segments in the test utterance. Note that the segments selected in the sum vary by speaker. It seems reasonable that contaminating segments may have lower scores for the true speaker, in which case TopNSeg eliminates contaminating scores. **Top1Seg**, a special case of **TopNSeg** is the maximum segment score for each speaker. A variant on the TopNSeg score, **TopSegMtoN**, is defined as the sum of the ranked segment scores M to N. Unwanted high scoring segments can occur when an anomalous event scores significantly better on one speaker's models than other speakers' models even though the unnormalized scores of all models are relatively low.

The **0-Clip** score sums only the positive normalized segment scores. A positive score for a speaker results only if that speaker scored higher than everyone else in a given segment. The 0-clip method is ideal for identifying speakers engaged in dialog since it is not necessary to know the proportion of time each participant contributes to the conversation. In fact, the top two scores from the 0-Clip algorithm can be reliably used to identify both speakers of a two speaker dialog [24].

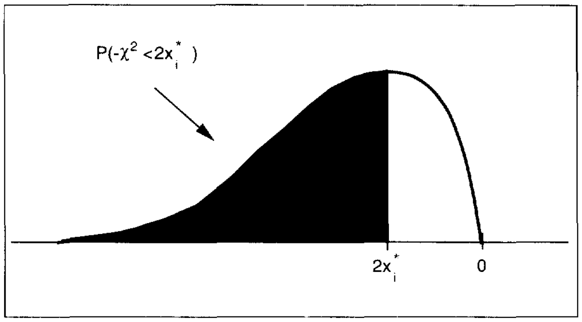
A final scoring algorithm is motivated as follows. Rather than using or discarding segment scores, scores are emphasized or deemphasized. Let Ω be a space of multi-dimensional values which can be taken by a parameter Θ. Suppose ω is a subset of Ω and define null and alternative hypotheses,

$$H_0 : \theta \in \omega \text{ and } H_1 : \theta \in \Omega, \theta \notin \omega$$

respectively. If L is a likelihood function depending on a parameter θ then the likelihood ratio statistic λ is defined as the ratio

$$\lambda = \frac{\max_{\theta \in \omega} L(\theta)}{\max_{\theta \in \Omega} L(\theta)}$$

Note that λ is always in the interval from zero to one. The likelihood ratio test favors H_0 when λ is high and H_1 when λ is low. Under suitable conditions $-2 \log \lambda$ approaches a chi-square distribution when H_0 is true ([18] p. 124). A likelihood-ratio test can be performed on each segment to measure whether the speaker of the test utterance is speaker i or some other speaker. Specifically, we can measure confidence in the statistic scores by applying likelihood ratio testing. Let H_0 be the hypothesis that speaker i is the speaker of a segment and let H_1denote the hypothesis that some other speaker is speaking. So ω consists of all means and covariances which could be generated by speaker i and Ω is the collection of parameters which could possibly be generated by all speakers. The numerator of Equation 3 is estimated using the speaker i available parameters; the means and covariances of the training



6. Chi-square segment score is the tail probability of a χ^2 distribution.

models. The denominator is estimated by maximizing over the parameters obtained from the available reference speakers. Let x_i and x_i^* denote the normalized and unnormalized segment scores of speaker i respectively. Recall that the unnormalized scores are log likelihoods and the normalized scores are likelihood ratios. Observe that log $\hat{\lambda} = x_i^*$, the normalized score, where $\hat{\lambda}$ is the estimate of the likelihood ratio statistic. It follows that $-2x_i^*$ should have an approximately chi-square distribution. New segment scores are defined on the segments using tail probabilities from the chi-square distribution as follows:

$$\text{Speaker } i \text{ chi-square segment score} = P(-\chi^2 < 2x_i^*)$$

These chi-square tail scores are between 0 and 1 for negative normalized segment scores and are equal to 1 for positive normalized scores (Fig. 6). If a speaker does better than all other reference speakers on a segment, then a score of 1 is achieved no matter how much better. The one parameter of chi-square distribution, the degrees of freedom, equivalently the mean, is estimated using all the positive normalized segmental scores. The **ChiSquareSum** score for a target is defined to be the sum of all his or her chi-square segment scores.

Combining Statistic Scores

As noted earlier, the three statistic scores can be weighted and combined to form new scores,

$$\alpha \text{ mean } + \beta \text{ cov} + \gamma \text{ dcov}$$

We can without loss of generality assume that $\alpha + \beta + \gamma = 1$. The weights should reflect the confidence in a given score. When no *a priori* information is available, the baseline score, generated by using equal weights, $\alpha = \beta = \gamma$, is an option.

Recall that the mean score is perhaps more sensitive to channel differences. If the scenario is such that channel is an indicator of speaker, for example if each speaker uses a single telephone, then it seems reasonable to emphasize the mean feature. Even with such information, it may not be clear which weights are optimal. Exactly how much should the mean score be emphasized or deemphasized?

For the answer to this question, we look to the training data, the only available data. The assumption or hope is that training conditions match test conditions closely, so that what holds for training should be true for testing. Cross-validation refers to the practice of retaining part of the training data for testing in order to assess performance. The advantage of using training data for testing is that truth is known. Cross-validation is a form of supervised training. By holding back training sessions one at a time for testing, as many tests as training sessions can be performed.

For each cross-validated test, it is possible to determine weights which generate correct identifications simply by trial. A systematic approach is to try all possible combinations of weights obtained by incrementing and decrementing α, β, and γ by a fixed amount, say 0.05. For each cross-validated test, weights which correctly identify the speaker are averaged. These means are then averaged over all tests and the resulting grand mean weights can be used on the real tests.

Measuring Confidence

After performing closed-set speaker identification, it may be useful to measure confidence in recognition results to know which results are most likely to be correct. By deciding on a confidence threshold, identification results can be classified as reliable or unreliable. By culling the reliable tests, collections of tests with high recognition accuracy can be obtained. Tests classified as unreliable can be subjected to further processing, either human or machine.

In performing speaker identification the speaker with the maximum score is identified. The underlying assumption for assessing our confidence that an identification is correct is that the maximum scores resulting in correct identifications are in general higher than the maximum scores resulting in incorrect identifications. Assigning a measure of confidence is a quantification of this underlying assumption. The confidence measure is a number from 0 to 1 with 0 reflecting no confidence, and 1 certainty.

Two methods of measuring confidence are presented, one based on significance testing, the other based on Baye's rule. Obviously the scores, and hence the distributions, depend on the scoring algorithm used. We assume scores have been obtained using the robust segmental scoring algorithms. Let C_F and C_T denote the class of incorrect and correct identifications respectively. Let x denote the score of the identified speaker and let $f_F(x) = f(x \mid C_F)$, $f_C(x) = f(x \mid c_T)$ denote the distributions of incorrectly and correctly identified speakers respectively. The scores obtained using the robust segmental scoring algorithms can be modeled by a two-term mixture model,

$$p(x) = P(C_F)\, f_F(x) + P(C_T)\, f_T\,(x),$$

where $P(C_T)$ is the probability of correct identification, $P(C_F) = 1 - P(C_T)$ and f_F, f_C are assumed normal. The four parameters associated with the two univariate normal distributions as well as $P(C_T)$ can be estimated using cross-validation.

Measuring the confidence in a score by the significance method simply employs $f_F(x)$. The significance confidence measure is a measure of how far on the tail of the distribution $f_F(x)$ the observed score occurs. Specifically the significance confidence measure, denoted by CM(x), is defined as follows,

$$\text{CM}(x) = 1 - \int_x^\infty f_F(x)dx$$

The higher the confidence measure, the more we believe that the score is too high to have been generated by a misclassification. The problem with the significance approach to measuring confidence is that it does not make use of $P(C_T)$ or $f_T\,(x)$, both of which can have a great bearing on our belief that a score is from a correct classification. The Bayes confidence measure, discussed below, overcomes this problem The Bayes confidence measure, defined by Bayes' rule, is given by

$$P(C_T \mid x) = \frac{P(C_T) f_T(x)}{P(C_F) f_F(x) + P(C_T) f_T(x)}$$

which the probability that a correct classification is made given the observed score x of a speaker identification test.

It is possible to predict the identification accuracy of a collection of tests by averaging the Bayes confidences of the tests in the collection. In particular, a collection of tests having Bayes confidence measure greater than K percent should achieve a recognition accuracy of at least K percent.

The Bayes measure assumes that $P(C_T)$ is known or has been estimated from a cross-validation experiment. If this is not the case, $P(C_T)$ can be estimated directly from the unclassified tests if a collection of tests is available. Given a collection of test scores $\{x_1, x_2, \ldots, x_n\}$ $P(C_T)$ can be estimated by maximizing the likelihood

$$\prod_{i=1}^{n} \{ [1 - P(C_T)] f_F(x_i) + P(C_T) f_T(x_i) \} \tag{4}$$

Ideally, f_F and f_T are well separated when measuring confidence. As an extreme case, if f_F and f_C are identical, then it is impossible to distinguish between scores from correct and incorrect identifications. It is still possible for the recognition accuracy to be very high. On the other hand, if the distributions have no overlap, then it is possible to determine with 100 percent confidence which identifications are correct. In some cases, it may be more useful to achieve 80 recognition accuracy, say, and know which identifications are correct than to obtain 95 correct, but not know which 95. Scoring algorithms which generate fewest identification errors may not be best for measuring confidence.

Switchboard Experiments

All experiments presented were performed one of two subsets of the Switchboard corpus. The corpus, collected by Texas Instruments, consists of spontaneous conversational speech recorded over long distance telephone lines from speakers representing all regions of the United States. The interested reader is referred to [25] for further information on Switchboard. The Switchboard speaker identification task, defined by the National Institute of Standards, consists of 24 targets, 12 male and 12 female. We refer to this task as SWBDTEST. Up to six approximately 60 second training sessions are available for each speaker as well as 97 test utterances containing approximately 60 seconds of target speech plus 30 seconds of non-target noises, silence and cross-talk. These tests are referred to as 60 second tests. Shorter 10, 20 and 30 second tests are defined by dividing the 60 second tests into three parts. Each target may have from one to six tests. Roughly 90 of the test handsets were used in at least one of the training conversations.

The second subset of Switchboard is SPIDRE, the SPeaker IDentification REsearch Corpus. SPIDRE consists of 45 targets, 23 male and 22 female. Each target has four sessions, three of which can be used for training, and one for testing. Three different phone numbers, and so handsets, were used for the four conversations (two are the same) for each target making it possible to investigate the effect of channel on results. All SPIDRE results reported are for 30 second tests (roughly 30 seconds of target and 15 seconds of noise and cross-talk) using channels not seen in training. Table 1 summarizes both SWBDTEST and SPIDRE.

Table 1: SWBDTEST and SPIDRE summary

SWBDTEST	SPIDRE
24 targets, 12 male, 12 female	45 targets, 23 male, 22 female
6 60s training sessions per speaker	3 60s training sessions per speaker
Target speech only in training sessions	Noise, cross-talk, etc. included in training
97 tests, 1 to 6 per speaker	45 tests, 1 per speaker
10,20,30,60 second tests	30 second test
Test handsets usually seen in training	Test handsets never seen in training

In Table 2 we see that a significant performance improvement is obtained by using probabilistic modeling over template matching. SPIDRE tests are especially difficult for template matching since the change in channels between training and testing alters the means. Using the mean statistic only from the Gaussian model results in the same 60% error rate as for template matching. The error rate using only the covariance feature is 38 percent.

Table 2: Template matching with Mahalonobis metric vs. Gaussian modeling

SPIDRE	Template Matching	Gaussian Model
Recognition Error Rate	60%	31%

Table 3 allows comparison among a few of the various modeling techniques. Notice all methods perform reasonably

Table 3: Recognition error rates for the various identification algorithms

SWBDTEST	60 Second Test	30 Second Test	10 Second Test
Nearest Neighbor/ VQ, [13]	4%	-	-
Gaussian Model	6%	6%	11%
Mixture Model	-	-	-
Modified Gaussian Model, Baseline	5%	5%	10%
Robust Segmental Method	Top40Seg 0%	Top20Seg 1%	Top7Seg 3%

well, but that no identification mistakes are made on the SWBDTEST 60 second test using the robust Top40Seg score.

Recognition Accuracy on SWBDTEST is usually so high that differences in the performance of scoring algorithms may be obscured. Nevertheless results for the 60, 30 and 10 second tests are displayed in Tables 4, 5 and 6. On SWBDTEST the cross-validated weights for combining the mean, cov and dcov scores are nearly equal, thereby justifying the baseline scoring method. Going from cov+dcov to baseline, misidentifications are reduced, showing that the mean statistic scores contain useful information.

Table 4: SWBDTEST 60 second test identification error rate

SWBDTEST 60 second	mean	cov	dcov	cov+dcov	Baseline
NoSeg	26%	8%	8%	5%	6%
SumSeg	20%	8%	6%	6%	3%
Top40Seg	14%	8%	4%	3%	0%
TopSeg6to35	15%	8%	4%	4%	0%
0-Clip	18%	10%	9%	5%	1%

Table 5: SWBDTEST 30 second test identification error rate.

SWBDTEST 30 second	mean	cov	dcov	cov+dcov	Baseline
NoSeg	25%	9%	9%	7%	5%
SumSeg	22%	9%	9%	7%	2%
Top20Seg	19%	11%	7%	6%	1%
TopSeg6to25	20%	9%	6%	7%	1%
0-Clip	19%	12%	9%	9%	3%
ChiSqSum	18%	6%	14%	5%	4%

Table 6: SWBDTEST 10 second test identification error rate.

SWBDTEST 10 second	mean	cov	dcov	cov+dcov	Baseline
NoSeg	30%	16%	21%	18%	10%
SumSeg	25%	13%	15%	15%	3%
Top7Seg	28%	14%	14%	13%	3%
TopSeg2to7	28%	14%	14%	13%	1%
0-Clip	31%	16%	19%	15%	5%
ChiSqSum	26%	13%	16%	12%	3%

Since the SPIDRE tests always occur on unknown channels, it is reasonable to expect that the mean statistic score may not be useful, and that the cov+dcov scores should be employed. Additionally cross-validation on training, building models from the two training sessions from each speaker with the same phone number, and testing on the third training session, shows that the mean feature is not helpful when combined with the other two features. SPIDRE results are presented in Table 7. Significant performance increases are obtained by the robust segmental scoring algorithms.

Table 7: SPIDRE 30 second test identification error rate.

SPIDRE 30 second	mean	cov	dcov	cov+dcov	Baseline
NoSeg	60%	38%	31%	29%	27%
SumSeg	64%	18%	24%	16%	22%
Top20Seg	62%	24%	18%	13%	22%
TopSeg6to25	62%	22%	11%	6%	22%
0-Clip	67%	27%	33%	18%	22%
ChiSqSum	64%	18%	20%	8%	20%

Comparing Tables 5 and 7, we see that the error rates for the SPIDRE tests are greater than for SWBDTEST. This is not surprising since SPIDRE consists of 45 speakers compared to only 24 for SWBDTEST and the benefit of channel information and the mean statistic score is not available. We also found that channel information affects the cov and dcov results, though not nearly as much as the mean results, by comparing results of the cov and dcov scores on the SWBDTEST tests with channel training to those without.

Normalizing the speakers' segment scores by other speakers' segment scores as described earlier improves the performance of the robust segmental scores significantly as demonstrated in Table 8.

Table 8: Comparison of normalized and unnormalized segment scores.

SWBDTEST 60 Second Test	Normalized Segment Scores	Unnormalized Scores
Top40Seg	0%	14%

Finally, an example of the performance of the Bayes confidence measure is presented. The error rate for the SPIDRE tests using the mean statistic feature and the SumSeg scoring algorithm is a high 64 percent. As noted earlier, the mean statistic is not the best feature to use for speaker identification; however it illustrates a situation where we may want to consider recognizing speakers only on a subset of the tests. Using Bayes confidences, we should be able to obtain a subset of tests that has any desired accuracy.

Before Bayes confidences can be computed, the error rate of the tests, $P(C_F)$, must be known or estimated. The error rate for the SPIDRE cross-validated tests using the mean feature is 43 percent. Since two of the three training conversations for each speaker use the same handset, the cross-validated test handsets will often be known in training. Thus this cross-validated performance estimate of $P(C_F)$, will not be appropriate for the real tests because of the difference in conditions between the training tests and the real tests. The maximum likelihood estimated error rate (Eq. 4) using the actual test scores, but not knowing truth, is 59%, much closer to the actual 64% error rate. These results are summarized in Table 9.

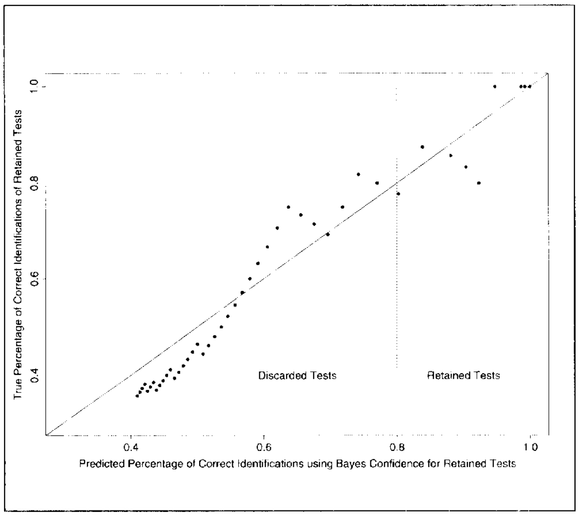
Table 9 SPIDRE using mean statistic and SumSeg algorithm.

Estimated Error Rate via Training CV Tests	43%
Estimated Error Rate using Test Scores	59%
Actual Error Rate of Tests	64%

Figure 7 compares the predicted percentage of correct identifications using Bayes confidences for the retained tests with the true percentage of correct identifications. To understand the figure, choose a desired recognition accuracy, say 80 percent. This recognition accuracy determines a threshold (viz., in this case 0.8), above which only those tests with a Bayes confidence greater than the threshold should be retained. The retained tests are represented by the points to the right of the dashed vertical line. Note that 9 tests are retained. The true recognition accuracy of the 9 retained points is given by the leftmost point and is 78 percent. The figure indicates that the correlation between the Bayes predicted accuracy and the true accuracy is high. (The solid diagonal line shows perfect correlation.)

Conclusion

We have described current approaches to text-independent speaker identification based on probabilistic modeling techniques. The probabilistic approaches have largely supplanted methods based on comparisons of long-term feature averages. The probabilistic approaches have an important and basic dichotomy into nonparametric and parametric probability models. Nonparametric models have the advantage of being potentially more accurate models (though possibly more fragile) while parametric models that offer computational efficiencies and the ability to characterize the effects of the environment by the effects on the parameters.



7. SPIDRE tests, SumSeg method.

A robust speaker-identification system was presented that was able to deal with various forms of anomalies that are localized in time, such as spurious noise events and cross-talk. It was based on a segmental approach in which normalized segment scores formed the basic input for a variety of robust procedures. Experimental results were presented, illustrating the advantages and disadvantages of the different procedures.

We showed the role that cross-validation can play in determining how to weight the different sources of information when combining them into a single score. Finally we explored a Bayesian approach to measuring confidence in the decisions made, which enabled us to reject the consideration of certain tests in order to achieve an improved, predicted performance level on the tests that were retained.

Herbert Gish and Michael Schmidt are with BBN Systems and Technologies, Cambridge, MA.

References

1. D. O'Shaughnessy, "Speaker Recognition," *IEEE ASSP Magazine*, October 1986, pp. 4-17.

2. G. R. Doddington, "Speaker Recognition - Identifying People by their Voices," *Proc IEEE*, 73 , 1985, pp. 1651-1664.

3. H. Gish, M. Krasner, W. Russell, J. Wolf, "Methods and Experiments for Text-Independent Speaker Recognition over Telephone Channels," *Proc. ICASSP '86*, April 1986, Tokyo, pp. 865 - 868.

4. L. R. Rabiner, B. H. Juang, *Fundamentals of Speech Recognition*, Englewood Cliffs, New Jersey: Prentice-Hall, 1993.

5. J. R. Deller, J. G. Proakis, J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*, New York: Macmillan, 1993.

6. S. S. Stevens, J. Volkmann, "The Relation of Pitch of Frequency: A Revised Scale," *Am. J. Psychol.*, Vol. 53, pp. 329-353, 1940.

7. C. Rowden "Analysis," *Speech Processing*, London: McGraw Hill, 1992.

8. A. M. Noll "Cepstrum Pitch Determination," *Journal of the Acoustical Society of America*, Vol. 41, pp. 293-309, 1967.

9. J. D. Markel, S. B. Davis, "Text-Independent Speaker Recognition from a Large Linguistically Unconstrained Time-spaced Data Base," *IEEE Trans. ASSP*, Vol. 27, No. 1, Feb. 1979, pp. 74-82.

10. K. Fukunaga, *Statistical Pattern Recognition*, 2nd Edition, San Diego: Academic Press, 1990.

11. R. O. Duda, P. E. Hart, *Pattern Classification and Scene Analysis*, New York: John Wiley Sons, 1973.

12. R. Schwartz, S. Roucos, M. Berouti, "The Application of Probability Density Estimation to Text-Independent Speaker Identification," *Proc. ICASSP '82*, May 1982, Paris, pp. 1649-1652.

13. A. Higgins, L. Bahler, J.Porter, "Voice Identification Using Nearest-Neighbor Distance Measure," *Proc. ICASSP* '93, April 1993, Minneapolis, Vol. II, pp. 375-378.

14. P. J. Huber, *Robust Statistics*, New York: John Wiley Sons, 1981.

15. E. W. Forgy, "Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classifications," *Biometrics*, Vol. 21, p. 768, abstract, 1965.

16. W. R. Dillon, M. Goldstein, *Multivariate Analysis, Methods and Applications*, New York: J. Wiley Sons, 1984.

17. F. Soong, A. Rosenberg, L. Rabiner, B. Juang "A Vector Quantization Approach to Speaker Recognition," Proc. ICASSP '85, March 1985, Tampa, pp. 387-390.

18. K. V. Mardia, J. T. Kent, J. M. Bibby, *Multivariate Analysis*, San Diego: Academic Press, 1979.

19. A. P. Dempster, N. M. Laird, D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society B*, Vol. 39, No. 1, 1977, pp. 1-22.

20. D. A. Reynolds, "A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification," Technical Report 967, Lincoln Laboratory, 1993.

21. T. W. Anderson, An Introduction to Multivariate Statistical Analysis, 2nd Edition, New York: J. Wiley Sons, 1971.

22. H. Gish, "Robust Discrimination in Automatic Speaker Identification," *Proc. ICASSP '90*, April 1990, Albuquerque, pp. 289-292.

23. H. Gish, M. Schmidt, A. Mielke, "A Robust Segmental Method for Text-Independent Speaker Identification," *Proc. ICASSP '94*, April 1994, Adelaide, South Australia, pp. 145-148.

24. G. Yu and H. Gish, "Identification of Speakers Engaged in Dialog," *Proc. ICASSP '93*, April 1993, Minneapolis, Vol. II, pp. 383-386.

25. J. Godfrey, E. Holliman, J. McDaniel, "Switchboard: Telephone Speech Corpus for Research and Development," *Proc. ICASSP '92*, Mar. 1992, San Francisco, Vol. I, pp. 517-520.