PERGAMON

Pattern Recognition 35 (2002) 2801–2821

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Speaker recognition—general classifier approaches and data fusion methods

Ravi P. Ramachandran[a,*], Kevin R. Farrell[b], Roopashri Ramachandran[a], Richard J. Mammone[c]

[a] *Department of Electrical Engineering, Rowan University, 201 Mullica Hill Road, Glassboro, NJ 08028, USA*
[b] *T-Netix Inc., Englewood, CO 80012, USA*
[c] *CAIP Center, Rutgers University, Piscataway, NJ 08855, USA*

Received 28 February 2000; received in revised form 28 August 2001; accepted 16 October 2001

Abstract

Speaker recognition refers to the concept of recognizing a speaker by his/her voice or speech samples. Some of the important applications of speaker recognition include customer verification for bank transactions, access to bank accounts through telephones, control on the use of credit cards, and for security purposes in the army, navy and airforce. This paper is purely a tutorial that presents a review of the classifier based methods used for speaker recognition. Both unsupervised and supervised classifiers are described. In addition, practical approaches that utilize diversity, redundancy and fusion strategies are discussed with the aim of improving performance. © 2002 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Speaker recognition; Feature; Classifier; Robust; Speaker model; Unsupervised; Supervised; Diversity; Redundancy; Fusion

1. Introduction

Speaker recognition refers to the concept of recognizing a speaker by his/her voice or speech samples. In automatic speech recognition, an algorithm takes the listener's role in deciphering speech waves into the underlying textual message. In automatic speaker recognition, an algorithm generates a hypothesis concerning the speaker's identity or authenticity. When the task is to identify the person talking rather than what the person is saying, the speech signal must be processed to extract measures of speaker variability instead of segmental features. Some of the important applications of speaker recognition include customer verification for bank transactions, access to bank accounts through telephones, control on the use of credit cards, and for security purposes in the army, navy and airforce. Speaker recognition is described in detail in [1–5].

The two main tasks within speaker recognition are *speaker identification* (ID) and *speaker verification.* Speaker identification deals with a situation where the person has to be identified as being one among a set of persons by using his/her voice samples. The objective of speaker verification is to verify the claimed identity of that speaker based on the voice samples of that speaker alone. For speaker recognition, the acoustic aspects of what characterizes the differences between voices are obscure and difficult to separate from signal aspects that reflect segment recognition. There are three sources of variation among speakers. They are

- Differences in vocal cords and vocal tract shape,
- Differences in speaking style (including accent),
- Differences in how speakers express themselves (words or phrases used) to convey a particular message [6,7].

Automatic speaker recognizers exploit only the first two variation sources, examining low-level acoustic features of speech, since a speaker's tendency to use certain words and

* Corresponding author. Tel.: +856-256-5334; fax: +856-256-5241.
E-mail address: ravi@rowan.edu (R.P. Ramachandran).

PII: S0031-3203(01)00235-7

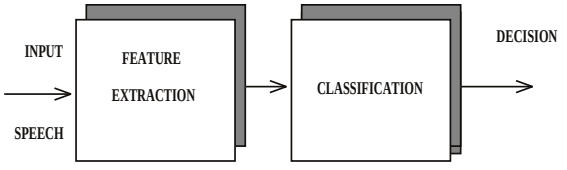

Fig. 1. A general diagram of a recognition system.

syntactic structures (the third source) is difficult to quantify or control in an experiment.

Speaker recognition consists of two stages, namely, *Feature extraction* and *Classification* as shown in Fig. 1. Feature extraction is associated with obtaining the characteristic patterns of the signal that are representative of the speaker in question. The parameters or features used in speaker recognition are a transformation of the speech signal into a compact acoustic representation that contains information useful for the identification of the speaker. This is often done using short-time linear predictive (LP) [8] analysis. The classifier uses these features to render a decision as to the speaker identity or verifies the claimed identity of the speaker. The speaker ID problem may further be subdivided into *closed set* and *open set*. The closed set speaker ID problem refers to a case where the speaker is known a priori to belong to a set of M speakers. In the open set case, the speaker may be out of the set and hence, a "none of the above" category is necessary. Another distinguishing aspect of speaker ID systems is that they can either be text-independent or text-dependent depending on the application. In the text-independent case, there is no restriction on the sentence or phrase to be spoken, whereas in the text-dependent case, the input sentence or phrase is fixed for each speaker. A text-dependent scenario is commonly encountered in speaker verification systems in which a person's password is critical for verifying his/her identity.

A classifier consists of M speaker models (one for each speaker) and the decision logic necessary to render a decision. In the training phase, the feature vectors are used to create a model for each speaker. During the testing phase, when the test feature vector comes in, a number will be associated with each speaker model indicating the degree of match with that speaker's model. This is done for a set of feature vectors and the derived numbers can be used to find a likelihood score for each speaker's model. For the speaker ID problem, the feature vectors of the test utterance will be passed through all the speakers' models and the scores are calculated. The model having the best score gives the speaker's identity (which is the decision). The output or score of the models may be a distortion measure or probability depending on the type of model. The identification success rate of the system is calculated as the ratio of the number of test cases for which the speaker is identified correctly to the total number of test cases for all the speakers. In open set problems, a scheme is used wherein a threshold value is needed in order to find out if the speaker is out of the set of M speakers. A similar thresholding scheme is also used for speaker verification. In closed set speaker identification, there is one source of error, namely, when a speaker is not identified correctly. In the open set case, there are two additional sources of error. First, a speaker not in the set of M speakers is deemed to be within the set. The second is the opposite scenario when a speaker in the set is deemed to be outside the set. In speaker verification, there are two sources of error. The first is known as a *false accept* (FA) and occurs when the user is accepted as the claimed speaker but in fact, is not the claimed speaker. This is when an imposter breaks in to the system. The second is known as a *false reject* (FR) and occurs when the user is rejected as the claimed speaker but is in fact the claimed speaker.

The recognition task is highly successful if the environmental conditions for training and testing are the same (known as matched conditions). Studies have shown that recognition performance degrades when the training and testing conditions are not the same (known as mismatched conditions) [4]. This occurs if the speaker is trained on one type of telephone (handset, cordless or speakerphone) and during the testing phase, a different type of telephone is used. In this particular case, channel mismatch is encountered and this contributes to the degradation in the recognition or success rate. Channels have a filtering effect on the speech and tend to alter the overall spectral envelope of the speech signal. Other examples of mismatched conditions are when training is done on clean speech (recorded in a sound-proof room with no telephone channel effects and no noise background) and testing is done on speech emanating from different noise backgrounds. There is great need to make the recognition system robust by achieving a high success rate even for mismatched conditions. Some of the feature-based methods for robust speaker recognition were discussed in a previous tutorial paper [9].

The robustness issue in speaker recognition can be examined from different angles all devoted to mitigating the effects of mismatched conditions. When training is done on clean speech and testing is done on channel corrupted or noisy speech, robustness can be accomplished at the signal level by speech enhancement techniques. Speech enhancement algorithms mitigate or even remove channel or noise artifacts in order to transform test speech into clean speech thereby providing a closer match to the training condition. Two well known enhancement techniques for noise effects include spectral subtraction [10] and Weiner filtering [11]. A method for transforming channel corrupted test speech into clean speech by inverse filtering the channel effect is described in Refs. [9,12]. Robustness at the feature level refers to either configuring robust features that show little variation for different conditions [9] or explicitly mapping the test feature vectors into the space of training feature vectors. The affine transform has been used to map feature vectors as a means to robustize speaker recognition systems [9,13]. The concept of mapping features for speech recognition has been considered in [14–16] and for word spotting has been

considered in [17]. At the classifier level, robustness can be achieved by integrating a model for the speech signal distortion into the overall classifier model [47] or mapping the classifier model obtained during training to better fit the testing condition [15]. Also, robust distortion measures that indicate a small distortion between a feature vector for a training condition and the corresponding vector obtained for a different testing condition have been formulated and applied to speech recognition [18]. Fusion enhances robustness by taking multiple opinions into account (discussed later). This tutorial discusses general classifier approaches and data fusion methods.

2. Feature extraction

The emphasis of this tutorial is on classifier based techniques. In this section, we present a brief synopsis of feature extraction for the purpose of completeness. The composite transfer function $V(z)$ of the glottal pulse model $G(z)$, the vocal tract model $H(z)$ and the radiation model $R(z)$ (at the lips) is usually modeled as an all-pole autoregressive process as [8]

$$V(z) = G(z)H(z)R(z) = \frac{1}{A(z)} = \frac{1}{\sum_{k=1}^{p} a(k)z^{-k}} = \frac{1}{\prod_{k=1}^{p}(1 - v_k z^{-1})}, \quad (1)$$

where v_k are the poles of $V(z)$ and p is the order of the model. The poles v_k that are close to the unit circle specify the formant frequencies (vocal tract resonances). The autoregressive model motivates a difference equation for synthesizing the speech signal $s(n)$ as a weighted linear combination of the p previous speech samples and input $e(n)$ as given by

$$s(n) = \sum_{k=1}^{p} a(k)s(n-k) + e(n), \quad (2)$$

where $a(k)$ are the weights. The model also motivates a linear prediction (LP) point of view in that $s(n)$ can be predicted as a weighted linear combination of the p previous samples. In this context, the weights $a(k)$ are known as the predictor coefficients and $e(n)$ is the prediction or estimation error signal.

The computation of $a(k)$ from a given speech signal is accomplished by minimizing the mean-square value of $e(n)$ over a frame of N samples. By assuming that the speech is zero outside the frame of interest, the procedure of minimizing the mean-square error is known as the autocorrelation method of linear prediction [8]. This yields a linear system of equations for obtaining $a(k)$. The autocorrelation method guarantees that all the poles of $V(z)$ are within the unit circle. The magnitude spectrum of $V(z)$ represents the spectral envelope of the speech. The predictor coefficients $a(k)$ are computed adaptively on a frame by frame basis due to the time-varying nature of the vocal tract. The size of the frame is usually between 20 and 30 ms.

The LP coefficients are converted into feature vectors. Ideally, it is desired that feature vectors show much similarity for a particular speaker and simultaneously show much variability for different speakers. This allows for easy discrimination among different speakers. In addition, the feature vectors should be robust to a wide variety of environmental conditions (such as different noise backgrounds and transmission media) by showing little variability as the environmental condition is changed. Examples of different features include the predictor coefficients themselves, reflection coefficients, LP cepstrum, line spectral frequencies, log area ratios, vocal tract area functions and the impulse response of $V(z)$ [8]. A comparative study of some of the above features revealed the LP cepstrum is the best for speaker recognition [19]. However, the LP cepstrum is not robust to channel and noise effects. Recently, features such as the Adaptive Component Weighted (ACW) cepstrum [20,21] and the Postfilter (PFL) cepstrum [21] have been found to be more robust to channel and noise.

3. General classifier structure

A general block diagram of a speaker recognition system is depicted in Fig. 2. As mentioned earlier, LP analysis of the speech and subsequent feature extraction leads to a set of feature vectors. The classifier consists of the various speaker models and the decision logic. Its operation constitutes two important steps. In the training phase, feature vectors are used to obtain the M speaker models. For each speaker, a different model is obtained from his/her speech. In the testing phase, feature vectors from an unknown speaker are first computed. For speaker identification, the feature vectors are compared with each of the M speaker models to get the scores Score(1) to Score(M). These scores are used to render a decision. In a closed set scenario, the best score, Score(i), identifies the unknown speaker as speaker i. This means that speaker model i most likely generated the feature vectors. In an open set scenario, Score(i) is further compared against a threshold to decide if there is an adequate match between the unknown speaker and the best model i. If the match is deemed to be adequate, the speaker is identified. Otherwise, it is decided that no speaker model represents the unknown speaker. For speaker verification, the unknown speaker claims a certain identity j. Only Score (j) is calculated and compared against a threshold to verify or reject the claimed identity.

There are two main categories of models, namely, those that are trained with unsupervised training algorithms and those that are trained with supervised training algorithms. The term "supervision" means that the data provided to the model training algorithm contains class information via a label. Unsupervised algorithms do not require any information regarding class membership for the training data. In terms of

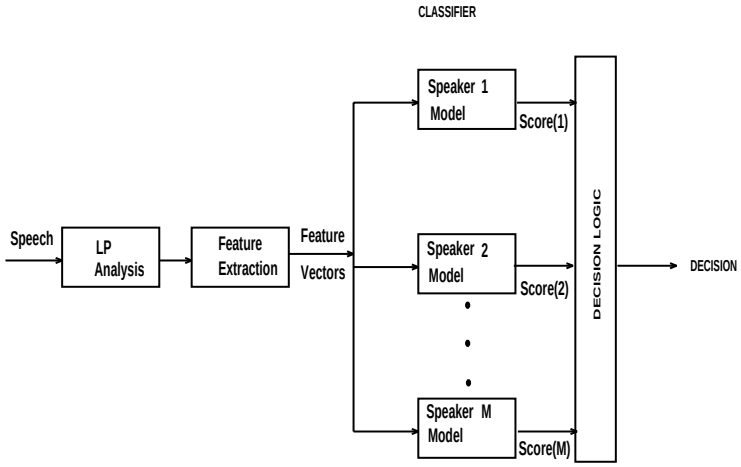

Fig. 2. A general block diagram of a speaker recognition system.

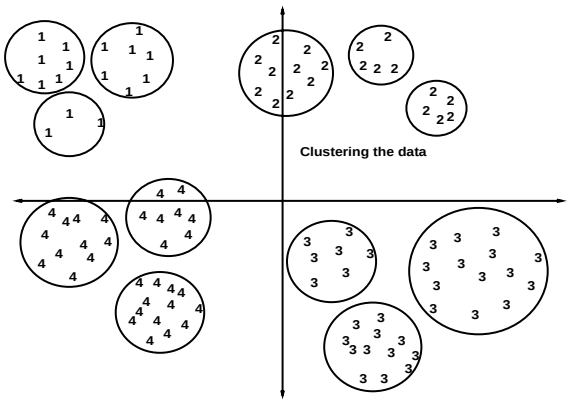

Fig. 3. Clustering approach to classification (4 speaker example).

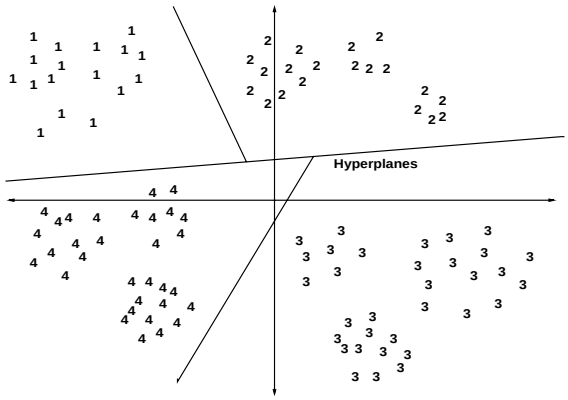

Fig. 4. Discriminator based approach to classification (4 speaker example).

speaker recognition, unsupervised training algorithms only use data from a specific speaker to create a model. Examples of unsupervised modeling approaches include nearest neighbor (NN), vector quantization (VQ), Gaussian mixture models (GMM) and hidden Markov models (HMM). Supervised training algorithms assign different labels to different speakers in a population. Examples of supervised training algorithms include multilayer perceptrons (MLP), radial basis functions (RBF), decision trees and neural tree networks (NTN). Supervised training algorithms have an advantage over unsupervised training algorithms in that they can better capture the differences between a particular speaker and other speakers in the population. However, the amount of training data and hence, the computational effort in deriving a speaker model can be more than that for unsupervised training algorithms.

Generally, classifiers use a clustering or discriminator based approach for rendering a decision. Fig. 3 illustrates the phenomenon of clustering in which the training data for each speaker are grouped into clusters that form the speaker model. The distance from the test data to the mean of the clusters indicates the similarity of the test data to each speaker model. The speaker model closest to the test data identifies the speaker. Clustering is used for the NN and VQ approaches. Fig. 4 illustrates the discriminator based method. Here, the feature space is divided into distinct regions by a series of hyperplanes. Test feature vectors falling into a specific region are deemed to have been generated by the corresponding speaker. The NTN and neural networks in general use a discriminator based method. In the next two sections, we describe the various classifiers in more detail.

4. Unsupervised classifiers

In unsupervised training, the speaker model is configured from feature data for that speaker only. This has an

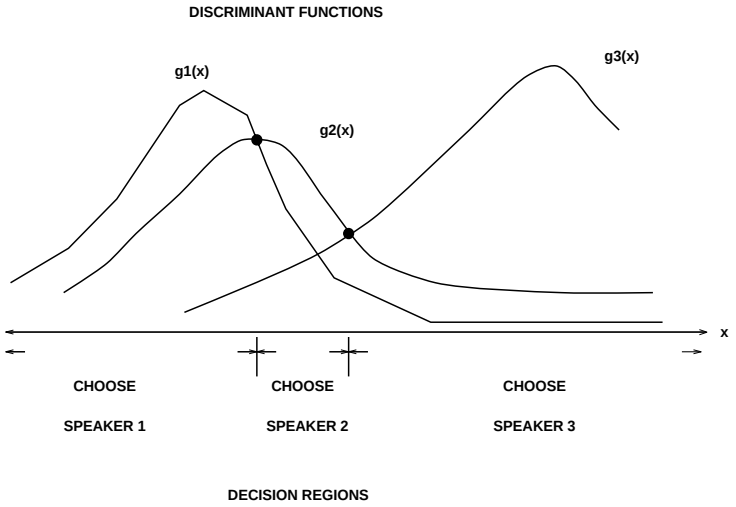

Fig. 5. Examples of decision regions given discriminant functions.

advantage both in terms of computation and in avoiding the reconfiguration of all speaker models when a new speaker is to be enrolled in the system. The nearest neighbor, vector quantizer and dynamic time warping classifiers use a distortion or distance measure to determine the score in comparing the test feature vectors and a speaker model. The Bayesian discriminant, Gaussian mixture model and hidden Markov model employ a probabilistic model for the speaker [22]. Classification decisions are based on probabilities and likelihoods.

4.1. Bayesian discriminant

Bayesian discriminant methods use probabilistic methods to model the speakers and then use Bayes theorem when making decisions. The speaker models correspond to a set of discriminant functions [23] $g_i(\mathbf{x})$ (for i=1 to M) where $\mathbf{x}$ is a feature vector. A feature vector $\mathbf{x}$ is most likely to have come from speaker model i if $g_i(\mathbf{x}) > g_j(\mathbf{x})\ \forall j \neq i$. The choice of discriminant functions is not unique. Given $\mathbf{x}$, the probability of making an incorrect decision is minimized if speaker i is chosen to maximize the a posteriori probability $p(i|\mathbf{x})$ [23]. Then, $g_i(\mathbf{x}) = p(i|\mathbf{x})$. From Bayes theorem which states that

$$p(i|\mathbf{x}) = \frac{f(\mathbf{x}|i)\,p(i)}{\sum_{k=1}^{M} f(\mathbf{x}|k)\,p(k)}, \tag{3}$$

we can define two equivalent discriminant functions as

$$g_i(\mathbf{x}) = f(\mathbf{x}|i)\,p(i), \tag{4}$$

$$g_i(\mathbf{x}) = \log f(\mathbf{x}|i) + \log p(i), \tag{5}$$

where $f(\,)$ denotes the probability density function. The feature space is divided into M distinct decision regions R_i each corresponding to a particular speaker or speaker model. Therefore, for each speaker model, the decision region contains all the feature vectors most likely emanating from that model. Fig. 5 shows the decision regions for the case of a scalar feature and three speaker models. The overall probability of correct classification is given by

$$p(\text{correct}) = \sum_{i=1}^{M} \int_{R_i} f(\mathbf{x}|i)\,p(i)\,\mathrm{d}\mathbf{x} \tag{6}$$

and is maximized since the regions R_i are chosen such that the integrands are maximum. In effect, the probability of error or incorrect decision $p(\text{error}) = 1 - p(\text{correct})$ is minimized.

In practice, an ensemble of test feature vectors $\mathbf{x}_1, \ldots, \mathbf{x}_q$ are used to recognize the speaker. Given such an ensemble,

$$\text{Score}(i) = \sum_{k=1}^{q} g_i(\mathbf{x}_k). \tag{7}$$

The speaker is identified as corresponding to the model that maximizes Score(i) for $1 \leqslant i \leqslant M$. Threshold comparisons as described earlier are required for the open set case and speaker verification. Speaker verification using discriminants has been accomplished in Ref. [24] in which $f(\mathbf{x}|i)$ are specified as multivariate Gaussian probability density functions.

The usage of discriminant functions relies on an estimate of $f(\mathbf{x}|i)$ given N samples of the feature vectors for speaker i. These samples are the training data used to obtain the speaker model for speaker i. The Parzen window approach [23,25] is effective for estimating both unimodal and multimodal densities $f(\mathbf{x})$ as described below. It is well known that the probability p of $\mathbf{x}$ falling in a region R is given by

$$p = \int_R f(\mathbf{x})\,\mathrm{d}\mathbf{x}. \tag{8}$$

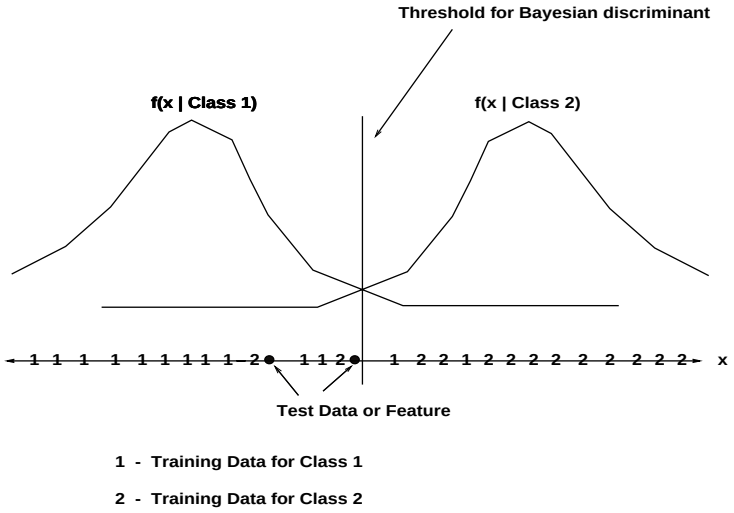

Fig. 6. Illustrative example comparing Bayesian discriminant and nearest neighbor rules.

If $f(\mathbf{x})$ is continuous and R is small enough such that $f(\mathbf{x})$ does not vary much within R, an estimate is given by

$$f(\mathbf{x}) = \frac{k/N}{V}, \tag{9}$$

where V is the volume of region R, N is the number of sample feature vectors and k is the number of sample feature vectors in R. Suppose a sequence of regions $R_1, R_2, \ldots, R_n$ is formed such that V_j is the volume of R_j and k_j is the number of sample feature vectors in R_j. The nth estimate of $f(\mathbf{x})$ is

$$f_n(\mathbf{x}) = \frac{k_n/N}{V_n}. \tag{10}$$

Convergence of $f_n(\mathbf{x})$ to $f(\mathbf{x})$ is assured if the volumes of the successive regions shrink uniformly. The Parzen window approach assumes that each region R_n is an m dimensional hypercube with volume $V_n = h_n^m$ where h_n is the length of an edge of the hypercube. The uniform shrinkage of the regions is assured by taking $h_n = h_1/\sqrt{n}$. Consider a vector $\mathbf{u}$ with components $u_1, u_2, \ldots, u_m$ and a window function $\psi(\mathbf{u})$ given by

$$\psi(\mathbf{u}) = \begin{cases} 1 & \text{for } |u_j| \leqslant 1/2,\ j = 1, 2, \ldots, m, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Note that $\psi(\mathbf{u})$ defines a unit hypercube centered at the origin. Given feature vector samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$, it follows that $\psi((\mathbf{x} - \mathbf{x}_i)/h_n)$ equals 1 if the sample vector $\mathbf{x}_i$ falls within the hypercube of volume V_n centered at $\mathbf{x}$. Therefore, the number of samples in the hypercube is

$$k_n = \sum_{i=1}^{n} \psi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) \tag{12}$$

and hence, the density estimate is

$$f_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \psi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right). \tag{13}$$

Finally, note that other window functions $\psi(\mathbf{u})$ can be used as long as $f_n(\mathbf{x})$ is a legitimate density function. For this to be so, it is required that

$$\psi(\mathbf{u}) \geqslant 0, \tag{14}$$

$$\int \psi(\mathbf{u})\, \mathrm{d}\mathbf{u} = 1. \tag{15}$$

4.2. Nearest neighbor

The nearest neighbor (NN) approach bypasses the step of explicitly estimating $f(\mathbf{x}|i)$. The speaker model i is the set of training feature vectors for speaker i which would otherwise be used to estimate the underlying density function $f(\mathbf{x}|i)$. This would in turn be used to get the discriminant function for the Bayesian classification approach as described above. Given a test feature vector $\mathbf{x}$, its closeness or similarity to model i is the distance to the closest training feature vector comprising model i (denoted as d_i). From the M distances d_1 to d_M, $\mathbf{x}$ is associated with model i if $d_i < d_j\ \forall j \neq i$. This is equivalent to finding the closest training vector among the training data for all speakers and associating the test vector with the model containing this closest training vector. The NN rule is suboptimal in that the probability of error for the NN rule is lower bounded by the probability of error for the Bayesian discriminant classifier. The upper bound is twice the probability of error of the Bayesian discriminant classifier [23,25]. The suboptimality of the NN rule is illustrated by a figurative example in Fig. 6 (a similar example is

given in Ref. [26]). In this example, consider an equiprobable two class scalar problem for which the conditional probability density functions and the decision threshold for the Bayesian discriminant method are shown in Fig. 6. Moreover, the points marked 1 and 2 are the training data for classes 1 and 2, respectively. Two test features from class 1 are correctly classified by the Bayesian discriminant method but incorrectly classified by the NN rule because they are closer to a training data sample from class 2. Note that the class 2 training data samples that force the errors for the NN rule occur with low probability. It is this phenomenon that marks the major source of difference between the Bayesian discriminant and the NN approaches. Generally, if the two classes are well separated (thereby rendering a low probability of error for the Bayesian discriminant approach), then the probability of error for the NN rule is also low if the amount of training data is large [23].

The NN method has been used for classifying individual pixels of an image of the earth's surface as being soil or vegetation [26]. For speaker recognition, we use an ensemble of test feature vectors $\mathbf{x}_1, \ldots, \mathbf{x}_q$. A feature vector is associated with a particular speaker model according to the NN rule. A score of one is recorded for the matching model with a score of zero for the other models. This process is repeated for each feature vector in the test ensemble. The overall score for model i, namely Score(i), is the sum of the scores obtained for each individual test vector. The speaker is identified as corresponding to the model that maximizes Score(i) for $1 \leqslant i \leqslant M$. Again, appropriate thresholding is required for the open set case and for speaker verification. A generalization of the NN rule is the k nearest neighbor (kNN) rule. Consider a test feature vector. Among the training data for all the speakers, the k closest or nearest neighbors are found. A majority vote among these closest neighbors associates the test vector with a speaker model. The scoring is as described earlier for the NN case. Experiments suggest that k should increase as the speaker population M becomes larger [27].

Another generalization of the NN rule involves the explicit incorporation of distances [28,29]. Consider speaker model i containing the training set $T_i = (\mathbf{t}_1, \mathbf{t}_2, \ldots)$ and a set of test feature vectors $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots)$. The accumulated distance between X and T_i is Score(i) and is given by

$$\text{Score}(i) = \frac{1}{N(X)} \sum_{\mathbf{x}_j \in X} \min_{\mathbf{t}_k \in T_i} d(\mathbf{x}_j, \mathbf{t}_k) \tag{16}$$

$$+ \frac{1}{N(T_i)} \sum_{\mathbf{t}_k \in T_i} \min_{\mathbf{x}_j \in X} d(\mathbf{x}_j, \mathbf{t}_k)$$

$$- \frac{1}{N(X)} \sum_{\mathbf{x}_j \in X} \min_{\mathbf{x}_k \in X, k \neq j} d(\mathbf{x}_j, \mathbf{x}_k)$$

$$- \frac{1}{N(T_i)} \sum_{\mathbf{t}_j \in T_i} \min_{\mathbf{t}_k \in T_i, k \neq j} d(\mathbf{t}_j, \mathbf{t}_k), \tag{17}$$

where $N(.)$ is the number of vectors in the set and $d(.,.)$ is the distance between two vectors. The quantity Score(i) is computed for each speaker model. The speaker is identified as corresponding to the model with the minimum score. The expected value of Score(i) can be shown to be directly related to the divergence between the underlying probability density function of the training set and the test vectors [28]. The divergence is a measure of the similarity of two probability density functions. Hence, a low score reveals a strong match between the training and test data. Finally, note that the use of NN classifiers requires the storage of a huge amount of training data and extensive computations to find the closest neighbor. This motivates the use of vector quantizers as discussed next.

4.3. Vector quantizer

The vector quantizer (VQ) approach, like the NN method, does not involve the step of explicitly estimating $f(\mathbf{x}|i)$. Moreover, the memory and computational complexity of the NN method is considerably alleviated by establishing the speaker models to be a compressed version of the set of training feature vectors. The training vectors are from an underlying density function $f(\mathbf{x}|i)$ and span the feature space. The feature space is partitioned into a set of mutually exclusive convex regions, the number of which is much less than the number of training vectors [30,31]. The regions represent a different cluster of the training data and have a center of gravity or centroid [30,31]. This collection of centroids is called the codebook which is a compressed version of the training data and which also emanates from the same density function $f(\mathbf{x}|i)$. The speaker model is the codebook derived from the training data and reflects the underlying density $f(\mathbf{x}|i)$. The transformation or compression of the training data to the codebook is known as VQ design. Algorithms for VQ design that compress the training data include the Linde–Buzo–Gray (LBG) method [32], the learning vector quantization (LVQ) method [33] and group vector quantization (GVQ) [34]. The LBG method is based on iteratively updating the convex regions and the centroids or codebook to locally minimize the mean-square quantization distortion over the set of training data. Minimizing the quantization distortion does not imply that the identification success rate is maximized. The LVQ method is an attempt to reduce the number of misclassified feature vectors by considering one training vector at a time to iteratively refine the borders between classes. However, the identification success rate for the LVQ method is not guaranteed to be better than the LBG approach since the score is calculated over a speech utterance comprised of a group or set of feature vectors (elaborated on in the next paragraph). The GVQ method iteratively updates the codebooks by considering a group of training vectors. The speaker identification performance for the GVQ is shown to be better than the LBG and LVQ methods [34].

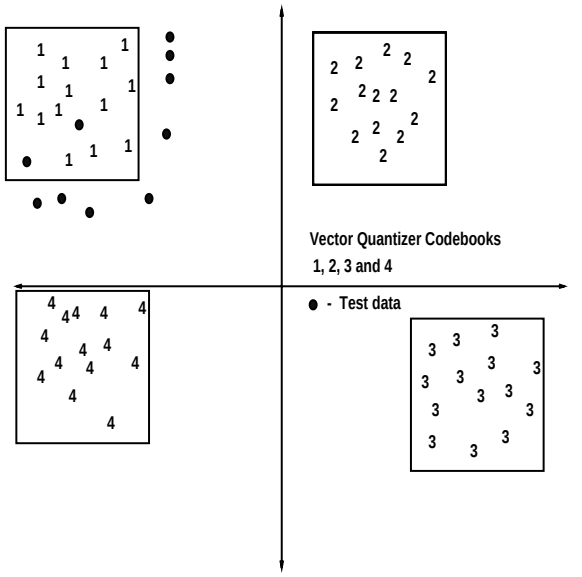

Fig. 7. VQ codebooks and test data for 4 speakers.

Given a test feature vector $\mathbf{x}$, its closeness or similarity to model i is the distance to the centroid vector or codevector comprising model i (denoted as d_i). A total of M distances d_1 to d_M are found to get a similarity measure between $\mathbf{x}$ and each speaker model. This process is repeated for each feature vector in the test ensemble. The overall score for model i, namely Score(i), is the sum of the distances obtained for each individual test vector [27,35,36]. The speaker is identified as corresponding to the model that minimizes Score(i) for $1 \leqslant i \leqslant M$. This, like the NN method, is equivalent to finding the best match between the underlying probability density of the test data and the underlying density functions of the training data for each speaker. Fig. 7 shows the positioning of the codebook for a closed set system involving 4 speakers. The test data is closest to the codebook of speaker 1 and hence, this speaker is identified as most probably generating the test data.

For the open set case and speaker verification, the use of adaptive thresholding based on the relative accumulated distances to similar speakers has been investigated. This is known as cohort normalization and can be applied to VQ or other unsupervised methods [37,38]. Here, feature vectors from speakers other than the target speaker (known as cohorts) are used to get VQ codebooks. Given a set of test feature vectors, a score based on the VQ codebook for the target speaker is obtained. Then, a score based on the VQ codebooks for the other speakers is obtained. The ratio of these scores (known as a likelihood ratio) is compared to a threshold to either accept or reject the speaker. In practice, VQ classifiers are very popular [27,35,36]. Also, a small codebook size of 64 or 128 codevectors is sufficient [27]. Even if there are 50 speakers enrolled, the memory requirement is feasible for real-time applications.

4.4. Gaussian mixture models

When creating a speaker model using a Gaussian mixture model (GMM) approach, we start with a discriminant function of the form $g_i(\mathbf{x}) = \log f(\mathbf{x}|i)$. The density $f(\mathbf{x}|i)$ is a linear combination or mixture of L multivariate Gaussian densities as given by [39–41].

$$f(\mathbf{x}|i) = \sum_{j=1}^{L} w_j b_j(\mathbf{x}), \tag{18}$$

where w_j are the scalar weights applied to each Gaussian multivariate probability density function $b_j(\mathbf{x})$ which is in turn completely specified by its mean vector and covariance matrix. The sum of the weights w_j equals 1. The GMM classifier is essentially a Bayesian discriminant with speaker models $g_i(\mathbf{x})$ being a Gaussian mixture density that is multimodal. A unimodal Gaussian density can be thought of as modeling an acoustic class representing a phonetic event (like vowels, nasals or fricatives) and hence, reflecting a speaker dependent vocal tract configuration [40]. A mixture density models a set of acoustic classes that are speaker dependent. A multimodal density consisting of a linear combination of Gaussian basis functions can also approximate arbitrary continuous density functions [40]. A GMM can be viewed as a hybrid between a simple Bayesian discriminant using only one Gaussian density and a VQ codebook that can model arbitrary probability densities by a discrete collection of centroids [40,41] in that a smooth, continuous overall fit is provided for arbitrary densities.

In configuring a speaker model, training feature vectors are used to establish the parameters of the mixture density which are the weights w_j and the mean vectors and covariance matrices of the individual Gaussian densities $b_j(\mathbf{x})$. The speaker model represents a match to the underlying density of the training vectors. The number of mixtures L is predetermined. Determination of the GMM parameters is accomplished by a maximum likelihood formulation in turn achieved by the iterative expectation-maximization (EM) algorithm [42,43]. For testing, an ensemble of test feature vectors $\mathbf{x}_1, \ldots, \mathbf{x}_q$ are used to recognize the speaker. Given such an ensemble,

$$\begin{aligned} \text{Score}(i) &= \sum_{k=1}^{q} g_i(\mathbf{x}_k) \\ &= \sum_{k=1}^{q} \log f(\mathbf{x}_k|i). \end{aligned} \tag{19}$$

The speaker is identified as corresponding to the model that maximizes Score(i) for $1 \leqslant i \leqslant M$. Threshold comparisons are required for the open set case and speaker verification. A hypothesis testing framework based on cohort normalization is used to calculate a likelihood ratio for speaker verification [41]. Cohorts generally include speakers similar to the target speaker. This is impractical for portable systems since a different database must exist for each customer. A

more practical universal speaker independent background GMM model is proposed in Ref. [44] which can be adapted to the target speaker by changing the mean and variance parameters.

In practice, $L = 30$ to 50 Gaussian mixtures are used for each speaker model. Each mixture has a different diagonal covariance matrix [40,41,45–47]. It has been shown that the performance using GMM classifiers degrades due to mismatched conditions involving telephone channels, noise and nonlinear distortions due to handset or transducer variability [45,46]. One method of enhancing the robustness of the GMM classifier to noise is to establish a model for noise that is also a linear combination of Gaussian densities and integrate this model with the existing speaker models [47]. Substantial performance gains when training is done on clean speech and testing is done on noisy speech has been demonstrated with this integrated GMM speaker model [47].

4.5. Dynamic time warping

A pattern matching technique using Dynamic Time Warping (DTW) is a simple and effective nonstatistical tool that has been successfully used in several text-dependent speaker recognition tasks [7,48]. The speaker models consist of a reference "template" (generated during the training phase) that is a time-ordered series of feature vectors. During the testing or recognition phase, there is an ensemble of time-ordered test feature vectors $\mathbf{x}_1, \ldots, \mathbf{x}_q$ that are compared to the various reference templates. Since the time duration (and therefore the number of feature vectors) of the test utterance and the reference template may not be the same, it is necessary to normalize these timing differences. This is achieved by "warping" the time axis of the ensemble of test feature vectors to maximize the coincidence with the reference template using a dynamic programming based technique called Dynamic Time Warping (DTW) [49,50]. An example [7] of this nonlinear time alignment is illustrated in Fig. 8. After performing the DTW operation to register the training and testing patterns in time, a time-normalized distance is calculated as the minimized residual distance between the patterns. For speaker model i, this distance is Score(i). In speaker identification, the speaker model that yields the smallest time-normalized distance corresponds to the identified speaker. In the case of speaker verification, the time-normalized distance is compared against the threshold of the claimed identity's model to determine the accept/reject decision.

The reference template can be generated by using one training utterance or by averaging several training utterances. The time variations in the feature vectors of the multiple training utterances are normalized either by using linear warping functions or a nonlinear warping function using DTW. Sometimes, for enhancing robustness, a speaker model consists of multiple reference templates [51]. In this case, the kNN rule can be used to render a decision as follows. For a test pattern, the k closest reference templates

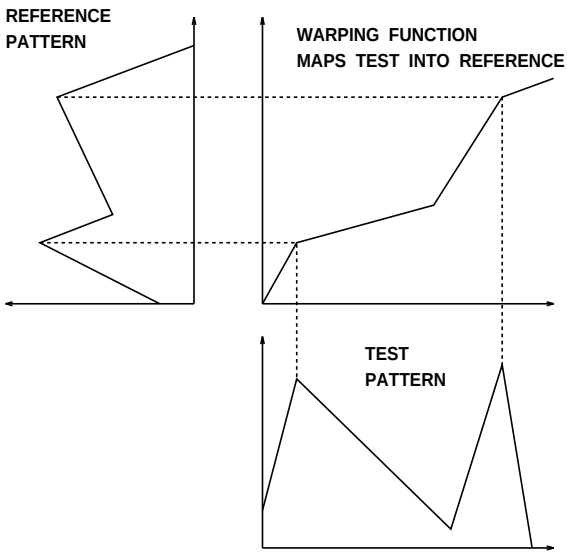

Fig. 8. Nonlinear time alignment of two patterns.

among the templates for all speakers are found. A majority vote among these selected templates identifies the speaker. Recently, discriminative training based algorithms have also been proposed that can be used to generate reference templates which will minimize the recognition errors [52].

The strengths of the DTW-based template matching system are its easy implementation and low complexity. It also captures very well the temporal variations typical of a speech signal. The distance from a DTW reference template can be viewed as a distance from a set of mean vectors. A reference model that incorporates both the mean and covariance statistics of a speaker may be more robust to variability in the testing conditions. The success of DTW is for text-dependent tasks due to the phonetic dependence between the training and testing speech which in turn allows for capture of temporal variation. For text-independent tasks, VQ is preferred.

4.6. Hidden Markov models

In a first order Markov process, the state at a particular time t depends only on the state at time $t-1$. The output of the process (observations) is the set of states at each instant of time which in turn are physically observable events. In a hidden Markov model (HMM), the states are not directly observable and the observations are probabilistic functions of the state [53–55]. Again, transition to a particular state only depends on the state at the previous time instant. An HMM is a doubly embedded stochastic process with an underlying stochastic process that is not observable (hidden) but can only be observed through another set of processes that produce the sequence of observations [54]. The elements of an HMM are the number of states, number of distinct observation symbols, the state transition probability distribution (the

probabilities of going from one state to another), the observation symbol probability distribution (probabilities of an observation symbol given a particular state) and the initial state distribution (probability of being in a particular state initially). At each time instant, the system goes from one state to another and produces an observation symbol based on the state transition and observation symbol probability distributions. Given an HMM, one of the basic problems is to determine the most likely state sequence given the observations.

The HMMs used for speaker recognition are either discrete or continuous in that the observation symbols are either VQ codebook labels (discrete case) or continuous probability measures from a GMM (continuous case). For each speaker, the training feature vectors are converted into codebook labels or probability measures which are in turn used to train or configure the HMM speaker model. Configuring an HMM speaker model is equivalent to finding the state transition, observation symbol and initial state probability distributions to maximize the probability of the observation symbols obtained from the training feature vectors. An iterative algorithm known as the Baum-Welch method is used for this purpose [53,54]. For closed set speaker identification [56], the test feature vectors are converted into codebook labels or probability measures. For each speaker model, the score is calculated as the probability of the observation sequence (see Refs. [53,54] for a full description of how the computation is done). The HMM speaker model that renders the highest probability or highest score identifies the speaker. For speaker verification [57,58], threshold comparisons are required. A universal cohort background model based on simple modifications of the HMM model of the target speaker is proposed in [59]. For certain applications, HMMs have been found to yield performance improvements over DTW based systems [60].

5. Supervised classifiers

Supervised training algorithms have recently come under investigation for speaker recognition applications. Supervised training methods differ from unsupervised methods in that the data used to generate a model is comprised of that from numerous speakers. Unsupervised training methods only use training data from the target speaker. The supervision during training is attributed to a label that is associated to each feature vector. This label determines the class membership of that vector, i.e., the speaker to which it belongs. One application of supervised training could be to use M labels for M classes. A classifier can then be trained to determine which of these M classes a feature vector belongs. Many supervised training algorithms are capable of generating a model that can distinguish one of M classes. However, in practice, typically a model is generated so that it can distinguish between one of two classes, namely whether or not the feature vector belongs to the target speaker. Such a model is then developed for each speaker in the population. This partitioning of training data is illustrated in Fig. 9. We have found that typically higher performance can be achieved by using a system with M binary classification systems as opposed to one M-ary classification system.

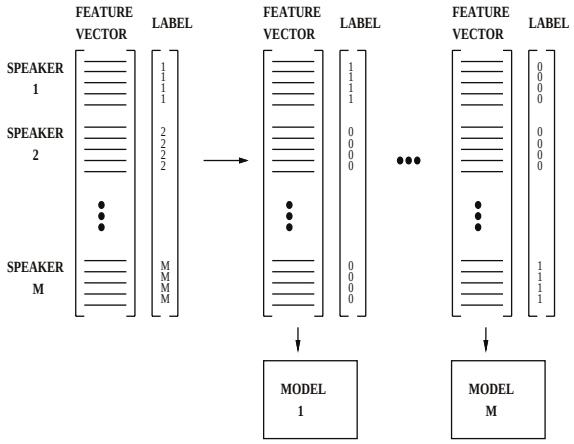

Fig. 9. Supervised training data partitioning.

Several supervised training algorithms have been investigated for speaker recognition. These include multilayer perceptrons [61,62], radial basis function networks [63], time-delay neural networks [64], and decision trees/neural tree networks [27]. Many of the initial speaker recognition applications of supervised training algorithms were done for closed set speaker identification [61,62,64]. Typically, the performance was compared to a benchmark obtained using vector quantization (VQ) techniques. The performance obtained with the supervised training algorithms was typically comparable to VQ techniques. However, the extensive training times necessary for most supervised algorithms was an undesirable feature. More recently, supervised training algorithms were evaluated extensively for speaker verification [27,63,65,66]. For tasks that require rejection capabilities, such as speaker verification and open set speaker identification, it was found that using the data from additional speakers during training greatly enhanced performance. The methods based on supervised training algorithms were found to consistently outperform the more traditional unsupervised algorithms, such as VQ [27], dynamic time warping (DTW) [65], and hidden Markov models (HMMs) [66].

This section describes some of the supervised training algorithms that have been applied to speaker recognition. These include multilayer perceptrons, radial basis functions, time-delay neural networks, decision trees, and neural tree networks. A summary of their implementation is provided in addition to a synopsis of the research findings regarding the specific approach.

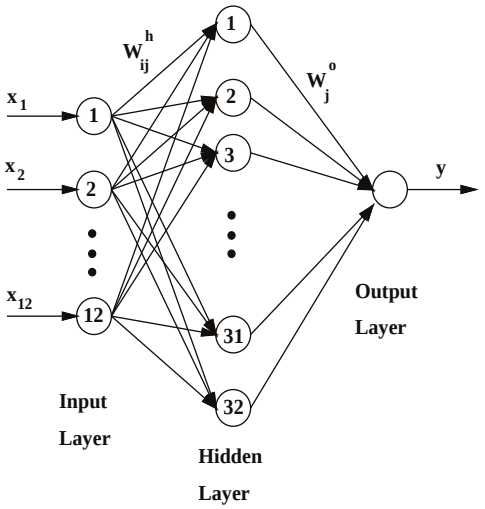

Fig. 10. Multilayer perceptron.

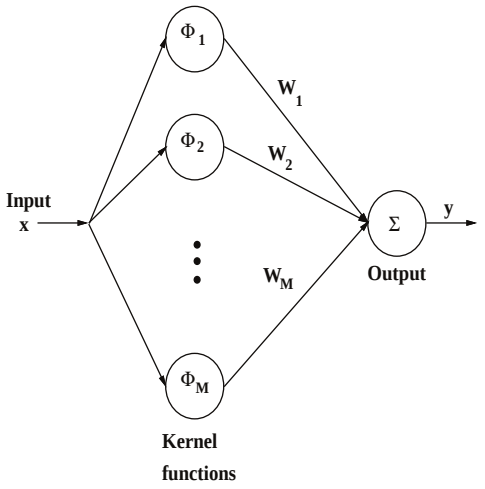

Fig. 11. Radial basis function network.

5.1. Multilayer perceptron

The multilayer perceptron (MLP) is a popular form of neural network that has been considered for various speech processing tasks [67,68]. The structure of a MLP is illustrated in Fig. 10. The weights for MLPs are trained with the backpropagation algorithm [69] such that they can associate a high output response with particular input patterns.

For speaker recognition, the MLP has been evaluated using a collection of binary-labeled training data sets [61] as described in the overview for supervised training algorithms. Ideally, test vectors for a specific speaker should have a "one" response for that speaker's MLP, whereas test vectors from different speakers should have a "zero" response. For speaker identification, all test vectors are applied to each MLP and the outputs of each are accumulated. The speaker is selected as corresponding to the MLP with the maximum accumulated output. For speaker verification, the test vectors are applied to the model of the speaker to be verified. The output is accumulated and normalized by the number of test vectors. If the normalized output exceeds a threshold, the speaker is verified, else rejected.

The MLP-based classifier has been reported to perform comparable to VQ for speaker identification [61,64]. A MLP-based classifier for speaker identification has been presented in [61], where each speaker is represented by a MLP. The training data consisted of 10 LP-derived cepstral coefficients. A MLP with one hidden layer and 128 hidden nodes achieved a 92% identification rate for this experiment, which was just slightly worse than the performance obtained with a VQ classifier with 64 codebook entries per speaker. The performance improved as the number of hidden nodes increased. However, it was observed that increasing the number of hidden layers did not improve generalization. It was also noted that the performance of MLPs degrades rapidly as the speaker population increases.

5.2. Radial basis functions

Radial basis function (RBF) networks [70] can be viewed as clustering followed by a perceptron. Consider the RBF in Fig. 11. The training phase consists of first clustering the training data into M clusters. The centroids of these M clusters are used within the kernel functions, which are typically Gaussian kernels or sigmoids. The outputs of the kernel functions are used to train a single layer perceptron. For the proper choice of kernel function and perceptron weights, the RBF network becomes equivalent to the GMM with the exception that supervision is available here.

Radial basis function networks have been considered for text-independent speaker verification [63]. The experiment was performed for a population of 40 speakers, for which 10 speakers were used as the enrolled speakers, and the remaining 30 were used as imposters. The experiments showed the RBFs to outperform both MLPs and VQ for speaker verification.

5.3. Time delay neural networks

Time delay neural networks (TDNNs) [71] are layered feed-forward neural networks that incorporate delayed versions of the inputs, so as to learn the temporal correlations of the input data. TDNNs are the supervised counterpart to the HMM in that they attempt to capitalize on the temporal information. TDNNs have been considered for text-independent speaker identification [64]. Here, TDNNs were evaluated for a population of 20 speakers (10 males and 10 females) and demonstrated a 98% identification rate.

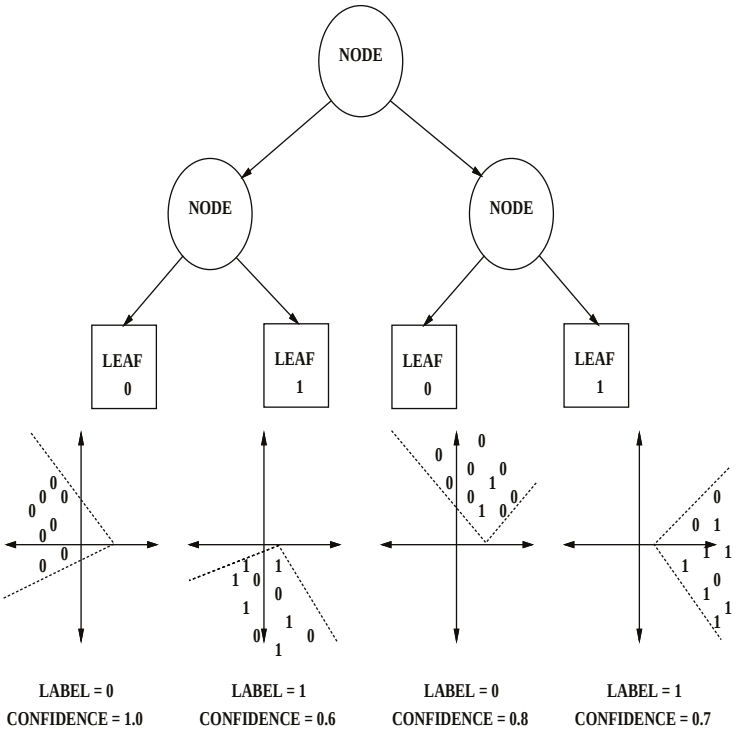

Fig. 12. Neural tree network—pruning.

5.4. Decision trees

A decision tree is a collection of rules, organized in a hierarchical fashion, that implement a decision structure [72]. The traditional algorithms for decision trees, such as ID3 [73] and C4 [74], will analyze the training feature vectors to determine the element of that vector most related to the class entropy. A decision rule is then generated which only considers that feature element. After the data is divided with this decision rule, the same concept will be applied to each subset of data resulting from the previous split. This process will continue until some stopping criteria is satisfied, else until the set of data only contains feature vectors from one class. Decision trees are useful in that they provide knowledge as to which elements of the feature vector are most pertinent for the recognition task. However, in terms of classification performance, this can be a drawback in pattern recognition tasks since the discriminant boundaries are constrained to being perpendicular to the feature axes. This was found to be the case in a vowel recognition task where MLPs were found to outperform decision trees [75].

A decision tree can be used for speaker recognition as follows. First, the feature vectors are obtained from the training data for all speakers. The data is then labeled as described earlier in this section. A binary decision tree is then trained for each speaker. The leaves of the binary decision tree contain the class label, where a one corresponds to the speaker and a zero corresponds to "not the speaker", or "antispeaker". For speaker identification, all feature vectors for the test utterance are applied to each decision tree. The labels are scored and the speaker is selected as having the decision tree with the maximum accumulated score. Decision trees were evaluated for closed set speaker identification [27]. They were found to perform slightly worse than MLPs.

5.5. Neural tree network

The neural tree network (NTN) [76] is a hierarchical classifier that combines the properties of feed-forward neural networks and decision trees. The neural tree network implements the sequential decision strategy used by decision trees. However, instead of determining the discriminant boundary from a feature element, a single layer perceptron is used to partition the data. The discriminant boundaries obtainable with the single layer perceptron are far more flexible than that of traditional decision trees and attribute to higher performance for classification tasks.

The NTN can be used for speaker recognition [27] in a similar fashion to the method described above for standard decision trees. Each speaker is associated with a binary NTN that is trained with feature vectors extracted from their enrollment session. Performance improvements can also be obtained by *pruning* the NTN during training. This prevents the NTN from growing too deep in order to partition a few

outliers. A simple method for pruning the NTN is to truncate the growth of the tree beyond a certain level and use relative frequency methods to compute the class probabilities at the leaves [27]. This concept is illustrated in Fig. 12. Here, it is seen that a probability can be computed as the ratio of the number of vectors for the target speaker to the total number of vectors.

The assignment of probabilities at the leaves allows one to deemphasize the contribution of vectors that fall within confusable regions of feature space. This is one of the fundamental differences between the NTN and other supervised modeling approaches. For example, multilayer perceptrons output a number that can be interpreted as a probability. However, this probability is obtained as an output of a sigmoidal activation function and is based on the position of the input vector with respect to the hyperplanes constructed by the MLP.

6. Diversity and redundancy

Diversity can be used as a means for improving system performance through the incorporation of different information. Similarly, redundancy can achieve the same goals through the re-use of data. One example where diversity has been investigated is for communications and, in particular, communications over fading channels. Several diversity based schemes have been used to decrease the bit error rates in such systems. One such scheme can be viewed as transmitting a signal twice in sequence. The hope here is that a channel distortion occurring at one instant in time is unlikely to occur during the second transmission and vice versa. This is a form of time diversity. Another scheme is to simultaneously transmit the information over two separate channels. Here, it is assumed that a distortion at one frequency may not occur at that frequency for the other channel. This is known as frequency diversity. Yet another method is based on space diversity. Here, the signal is received simultaneously by two antennas. The concept of redundancy has also been investigated in communications systems. One such application is error correction codes where redundant information is added to the binary signal to allow for error correction at the receiving end.

By applying diversity and redundancy techniques, a system will obtain several pieces of information that must then be combined. The combination of such information is typically referred to as data fusion, which will be discussed in the following section. In order for diversity techniques to provide some benefit to a system, the errors must have some level of uncorrelation. If two systems make the same errors on a task, then no combination of these systems will result in one that remedies these errors. On the other hand, if the two systems make different errors on a task, then the benefits of both can be capitalized by the proper combination. Uncorrelation of errors is a necessary condition for the success of diversity-based systems. A sufficient condition is that the errors be separable in a generalization map. As an example, consider the combination of GMM and NTN scores for speaker verification. A scatter plot of the scores for true speakers and imposters is shown in Fig. 13. A proper combination of the scores would eliminate errors that would otherwise occur if only one of the scores was used. It is seen that if a decision is based only on a GMM score and a threshold between 0.7 and 0.8 is used, there will be confusion between a true speaker and an imposter. The same applies to the NTN where a threshold between 0.35 and 0.5 will always have either a false reject or a false accept. Even though these individual models both have errors, the data in Fig. 13 is clearly separable. A simple method of combining the scores is to project them onto the diagonal. This leads to a set of one dimensional scores whose histogram is depicted in Fig. 14. These one dimensional scores are separable.

Several diversity schemes are applicable for speaker recognition. These are data diversity, feature diversity, and model diversity. These different forms of diversity may be combined to improve overall performance as long as they meet the criteria that the errors are uncorrelated. However, one must accept a tradeoff in memory and speed when using diversity. Each form of diversity that is used will result in an additional model, or models, which will require additional memory to store and processing time to evaluate. The merits of the performance gains must be weighed against these considerations. The following sections provide more detail on these forms of diversity for speaker recognition.

6.1. Data diversity

Data diversity refers to different training exemplars that can be applied to a speaker recognition algorithm. For example, if a person were to enroll in a system with eight repetitions of a password, then one example of data diversity would be to create two separate systems where one system is trained with the first four repetitions and the second system is trained with the last four. This would be in contrast to just training one model with all eight repetitions. Redundancy can also be applied to this situation in the form of a leave-one-out approach. The leave-one-out method [78] was originally proposed for estimating a statistic when only a small number of observations were available. The speaker recognition application is analogous in that the goal is to design a speaker model with a limited number of repetitions. In the above example with eight repetitions, eight models can be generated as opposed to one by using a different set of seven repetitions for each training session. Note that the leave-one-out method can be extended to a leave-M-out approach which would result in fewer models for $M > 1$. In either case, since there is a substantial amount of overlap in the training data for each model, this would be viewed as more of a redundancy application as opposed to a diversity application.

One motivation for training separate models in speaker recognition is that the unused enrollment data can be used

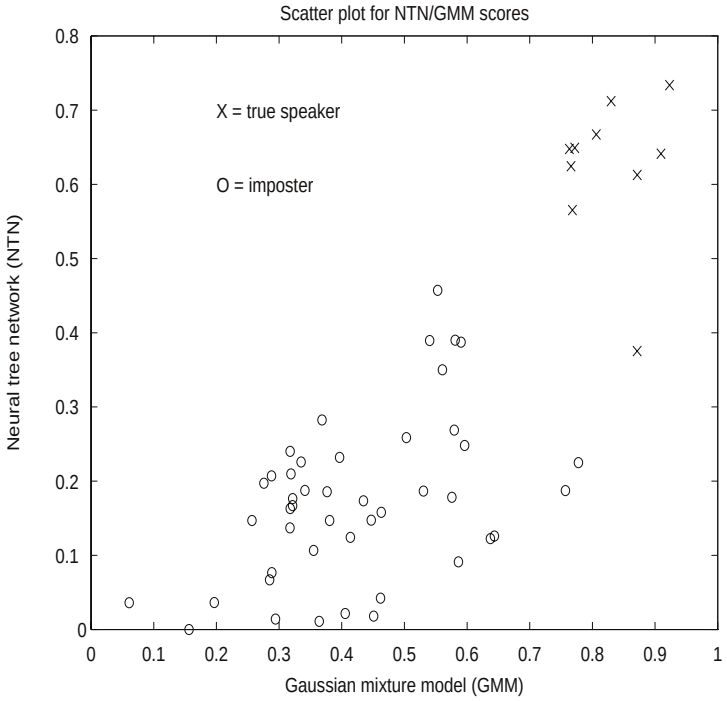

Fig. 13. Scatter plot of model scores.

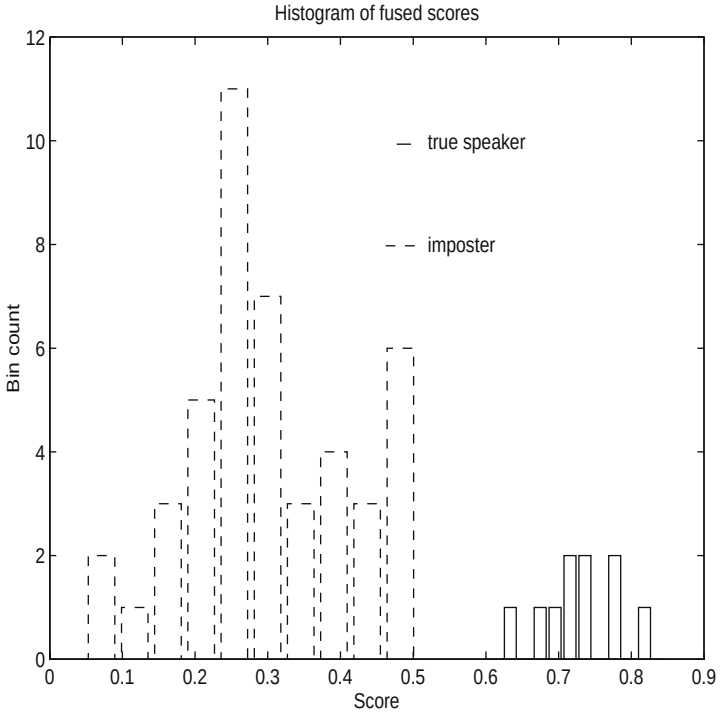

Fig. 14. Histogram of scores projected onto diagonal.

for cross-validation tasks or threshold estimation. For example, in the speaker verification system discussed in Ref. [65], only four repetitions of a password were available for model generation and threshold estimation. Here, the leave-one-out method was evaluated for accommodating this limitation in enrollment data. Three of the four utterances were used to train a model and the fourth utterance was used to obtain an independent speaker score for the model. By leaving out a different utterance each time, four models were generated and correspondingly four scores were available for estimating a mean speaker score to help in determining the threshold.

6.2. Feature diversity

Feature diversity refers to extracting different sets of features from the same speech signal(s). These different feature sets would then be used to build and test separate models. Again, the hope is that each feature set will capture some aspect of the speech signal that may be missed by the other feature set. In particular, if this attribute results in uncorrelated sets of errors for each feature set, then a performance improvement can be achieved. One example of feature diversity was investigated in which the cepstrum and delta cepstrum were used to train separate models whose test scores were subsequently combined [79]. Here, benefits were achieved from the instantaneous spectral information provided by the cepstrum and the transitional information provided by the delta cepstrum.

6.3. Model diversity

Model diversity refers to the use of different modeling techniques from the same feature sets. Different modeling approaches are known to capture different aspects of the features. For example, Gaussian mixture models and hidden Markov models will capture the statistics of the probability distributions for the feature vectors. Distortion based methods, such as vector quantization and dynamic time warping, will provide similarity measurements for the feature vectors observed during training and testing. Discriminant trained classifiers, such as neural networks, will capture differences between the feature vectors for a target speaker and those of non-target speakers. These different classes of modeling approaches each capture different characteristics of the feature vectors for a given speaker. When these approaches are evaluated on identical tasks, the errors tend to be uncorrelated [80]. Hence, model diversity can be a powerful method for enhancing speaker recognition systems.

Several model diversity approaches have been applied to speaker recognition applications in recent years. The combination of neural tree networks and vector quantization was evaluated for text-independent speaker identification [81]. Here, the distortion scores from the vector quantization models were converted to probabilities and then combined with the neural tree network scores to improve identification rates. Later, the combination of neural tree networks and dynamic time warping was evaluated for text-dependent speaker verification [65]. Here, the overall distortion score of the DTW approach was converted to a probability and combined with the output probability of the NTN. Error rates were cut in half as a result of this combination. Advantages were also found by combining a DTW score with a cohort normalized VQ score for speaker verification [82]. Another text-dependent speaker verification application used model diversity on a subword basis [83]. Here, a neural tree network and Gaussian mixture model was trained for each subword within a password. A further reduction in the error rate was obtained by combining three models, namely, the NTN, DTW and HMM [86] to a subword based text-dependent speaker verification system.

7. Data fusion

The combination of different sources of information has been explored within fields known as data fusion, consensus building, team decision theory, combination of multiple experts, along with numerous other titles. Here, we will refer to the combination of data from various sources as data fusion. Within the context of speaker recognition, data fusion comprises the combination of scores from different models trained for a speaker. These models may be trained with different speech data, different feature data, or different modeling techniques. Ultimately, it is desired that the errors of one model are corrected by the others and vice versa. If all models are in agreement upon an error, i.e., they all make the same mistake, then no combination will rectify the error. However, as long as there is some degree of uncorrelation among the errors, performance can be improved with the proper combination.

The selection of data fusion techniques can be subdivided based on the type of information that will be combined. For example, if the model outputs are probabilities, then methods such as linear or log opinion pools can be used [85]. If the model outputs are actually class labels, then methods such as voting [86] or ranking [87] can be used. For fuzzy decisions, Dempster–Shafer theory can be used for score combination [86]. The fuzzy integral method is also described.

7.1. Linear opinion pool

The linear opinion pool is a commonly used data fusion technique that is convenient due to its simplicity. The linear opinion pool is evaluated as a weighted sum of the outputs for each model:

$$P_{linear}(x) = \sum_{i=1}^{n} \alpha_i p_i(x), \tag{20}$$

where $P_{linear}(x)$ is the probability of the combined system, α_i are weights, $p_i(x)$ is the probability output by the ith model, and n is the number of models. The parameters α_i

are generally chosen such that α_i is between zero and one and the sum of the α_i's is equal to one.

The linear opinion pool is appealing in that the output is a probability distribution and the weights α_i provide a rough measure for the contribution of the ith model. However, it is noted that the probability distribution of the combiner output, namely $P_{linear}(x)$, may be multimodal. This may impose a more complicated decision strategy. The linear opinion pool has been evaluated with several scenarios within speaker recognition. These include the combination of VQ codebooks trained with cepstrum and delta cepstrum features [79], DTW templates with NTN [65], NTN and GMM [77,80] and VQ codebooks with dynamic time warping templates [82].

7.2. Log opinion pool

An alternative to the linear opinion pool is the log opinion pool. If the α_i weights are constrained to lie between zero and one and sum up to one, then the log opinion pool also outputs a probability distribution. However, as opposed to the linear opinion pool, the output distribution of the log opinion pool is unimodal [85].

The log opinion pool consists of a weighted product of the model outputs:

$$P_{log}(x) = \prod_{i=1}^{n} p_i^{\alpha_i}(x). \tag{21}$$

Note that with this formulation, if any model assigns a probability of zero, then the combined probability will also be zero. Hence, an individual model has the capability of a "veto", whereas in the linear opinion pool the zero probability would be averaged in with the other probabilities. The log opinion pool has also been evaluated for speaker verification, particularly for the combination of NTN/DTW [65] and NTN/GMM [80]. Here, the log opinion pool method was found to provide performance similar to the linear opinion pool method.

7.3. Voting/ranking methods

Another simple method for combining the results of multiple models is to use a voting procedure. In this case, each model must output a decision instead of a score. Typically, an odd number of models is used, to avoid ties, and the final decision is based on a majority rule. The voting method has been applied to handwriting recognition [86] and speaker verification [65,80]. In Refs. [65,80], the voting method was compared to the linear and log opinion pool methods. Here, the voting method performed slightly worse than the linear and log opinion pool methods, but still showed improvements over the individual model performances.

Ranking methods are appropriate for problems that involve numerous classes. Rankings do not use class labels or numerical scores, but instead utilize the order of the classes as estimated by the model(s). Ranking methods use class set reduction to reduce the number of class candidates without losing the true class. By reducing the number of classes and reordering the remaining classes, the true class should surface to the top of the ranking. The ranking method has been evaluated for printed character recognition [87].

7.4. Dempster–Shafer approach

The Dempster–Shafer approach has been used to combine the outputs of multiple classifiers for handwriting recognition [86] and is also analyzed in comparison with Bayesian methods and fuzzy set theory in [88]. The following constitutes a brief description. Consider a set of outcomes of an experiment to be denoted by Θ. For example, in a coin toss, if H represents heads and T represents tails, $\Theta = (H, T)$. The set Θ has $2^{n(\Theta)}$ subsets (including the null set and Θ itself) where $n(\Theta)$ is the number of elements in Θ. These subsets are known as propositions and the set of propositions is denoted as P. A proposition consisting of only one element is called a singleton. For example, if the experiment involves rolling a die, a singleton is the element 2 while the proposition that the number is even is $(2, 4, 6)$. A basic probability assignment (BPA) is assigned to each proposition or subset of Θ as opposed to each individual element of Θ as in conventional probability theory. If $A \in P$ is a subset of Θ, then BPA(A) represents the impact of the evidence (output of classifier) on A [86]. From the BPA, a numeric value in the range [0, 1] that indicates the belief in proposition A (denoted by bel(A)) is computed. The belief in A, bel(A), indicates the degree to which the evidence or classifier output supports A and is given by

$$\text{bel}(A) = \sum_{B \subseteq A} \text{BPA}(B). \tag{22}$$

Given Θ and P, the evidence (output) provided by the classifier induces a set of BPAs from which the beliefs are calculated. With multiple classifiers, each piece of evidence induces a different set of BPAs which must be combined or fused. If $A \in P$ is a subset of Θ which is not the null set, $\text{BPA}_1(A)$ is the BPA for one classifier and $\text{BPA}_2(A)$ is the BPA for the other classifier, the combining rule is given by Refs. [86,88]

$$\text{BPA}(A) = \frac{\sum_{C \cap D = A} \text{BPA}_1(C)\text{BPA}_2(D)}{1 - k}, \tag{23}$$

where BPA(A) is the overall BPA after fusion, $C \in P$, $D \in P$ and k is given by

$$k = \sum_{C \cap D = \text{null}} \text{BPA}_1(C)\text{BPA}_2(D). \tag{24}$$

Note that the classifier outputs are assumed to be independent. The BPAs for all $A \in P$ are found and the beliefs bel(A) computed before proceeding to invoke the decision

rule based on the beliefs. Note that if $k=1$, the two evidences are in complete conflict and BPA(A) does not exist. Also, the above formulation can easily be extended to include more than two classifiers [86,88].

It has been pointed out that there is no theoretical justification for the combining rule and hence, the potentially exponential complexity is hard to justify [88]. Similarly, the linear and log opinion pools have no theoretical justification but offer good results. The Dempster–Shafer method is intuitively pleasing in that it considers subsets as opposed to just singletons. However, the result can sometimes be counterintuitive in that high disbeliefs can be combined to render a high belief [88]. This counterintuitive phenomenon can be beneficial [88].

7.5. Fuzzy integral method

Fuzzy integrals use objective evidence supplied by various sources and the expected worth of subsets of these sources in the fusion process [89]. It can also be interpreted to be a maximum degree of belief for a class obtained from the fusion of several objective evidences [91]. The fuzzy integral approach has been used in handwritten word recognition [89–94] and computer vision [95]. It is included in this paper to provide completeness to this section on data fusion.

Let $X=(x_1,x_2,\ldots,x_n)$ be an arbitrary set. A function g defined on any Borel field of X is a fuzzy measure if (1) $g(\emptyset)=0$, (2) $g(X)=1$, (3) $g(A)\leqslant g(B)$ if $A\subset B$ and both A and B belong to the Borel field and (4) $\lim g(A_i)=g(\lim A_i)$ where A_i is an increasing sequence of measurable sets in the Borel field. A g_λ fuzzy measure satisfies an additional condition

$$g(A\cup B)=g(A)+g(B)+\lambda g(A)g(B), \tag{25}$$

where $A,B\subset X$, $A\cap B=\emptyset$ and $\lambda>-1$. The fuzzy density of the g_λ fuzzy measure is denoted by $g_\lambda(x_i)$ whose range is the closed interval $[0,1]$. The quantity λ can be obtained by solving the equation

$$\lambda+1=\prod_{i=1}^{n}(1+\lambda g_\lambda(x_i)). \tag{26}$$

If f is a measurable function whose range is the closed interval $[0,1]$ and such that $0\leqslant f(x_1)\leqslant f(x_2)\cdots\leqslant f(x_n)\leqslant 1$, then the Choquet (fuzzy) integral of f with respect to the fuzzy measure g is defined as

$$\int_A f(x)dg(.)=\sum_{i=1}^{n}[(f(x_i)-f(x_{i-1}))g_\lambda(A_i)], \tag{27}$$

where $f(x_0)=0$.

In implementing the fuzzy integral approach, the first step is to determine the fuzzy densities for the classifiers either subjectively or from the training data [91]. The fuzzy densities give a measure of the importance of the classifier for recognition of a certain class. Given the fuzzy densities, the second step is to calculate the g_λ fuzzy measure for each class from the different classifiers by the equation

$$g_\lambda(A_i)=\frac{1}{\lambda}\left[\prod_{x_i\in A_i}(1+\lambda g_\lambda(x_i))-1\right]. \tag{28}$$

From $g_\lambda(A_i)$, the fuzzy integral is computed where the function f is a membership function mapped from the actual output of the test data. The final output class (final decision) is that which maximizes the value of the fuzzy integral.

8. System description: techniques and methodologies

In this section, we will concentrate on describing the techniques, methodologies and issues in developing a speaker verification system. Most of the techniques described here have been implemented in a real system called the T-NETIX *SpeakEZ Voice Print SM* system [77]. This system is text-dependent, allows for user selectable passwords, and uses linear fusion to combine the scores of classifier models. Moreover, the system is based on segmenting each word in the password into sub-words.

The first issue to resolve is the choice between a text-dependent and text-independent system. Text-dependent speaker verification systems use the same password for training and testing. This method of verification has two important advantages, namely, (1) the password is fixed thereby making it difficult for an imposter to get in as the correct password must be known and (2) the system yields a better performance when compared to text-independent speaker verification systems. The second issue is to decide whether scoring and a decision is made on the entire password or by dividing the password into sub-words. The use of sub-words is better since (1) a method known as blind segmentation of a whole word into sub-words exists and automatically determines the number of phonemes and phonemic boundaries in which no apriori knowledge of what phonemes are present exists [96] and (2) allows for a user-selectable password that makes the system more consumer friendly. Also, the use of multiple sub-word level classifier models allows for scoring based on what would be a (1) phonemic match for a true speaker's utterance and (2) a glaring phonemic mismatch for an imposter's utterance. This allows for a larger disparity in the scores for true speakers and imposters which is in turn desired for getting better performance. The third issue is on selection of a robust feature(s). The Adaptive Component Weighted (ACW) cepstrum [20,21], the Postfilter (PFL) cepstrum [21] and pole filtered cepstral mean subtraction (PFCMS) [12] are individually robust but do not achieve any performance improvement when fused since the errors are not necessarily uncorrelated. Moreover, experiments reveal that transforming channel corrupted speech into clean speech by

inverse filtering the channel effect (as described in [9,12]) and using the LP cepstrum as the feature is preferred over directly using the ACW, PFL or PFCMS features. The use of inverse filtering and the LP cepstrum is the method of choice.

There are numerous training methods that can be applied to a speaker verification system. One simple case is when many repetitions of the password are recorded and one model is trained on all the repetitions. This overall model is based on all the password repetitions and will be comprised of various sub-word models (the sub-words are based on the blind segmentation). Another way is to train two overall models, each on half of the repetitions. A leave-one-out method is used (discussed earlier) to accomplish data diversity with only four repetitions of a password being available. Each overall model is trained with 3 repetitions with a different repetition "left-out" for each model. The left-out repetition is applied as a test utterance to get an unbiased score that is used to set the threshold for accepting or rejecting a claimant speaker. The approach generates 4 multiple overall models with each model consisting of the same number of sub-word models. The training method as above can be applied to any type of classifier. The mean and diagonal covariance matrix of the feature are used to obtain the GMM parameters. The GMM, VQ, HMM and DTW are trained only using the speech of the speaker being trained. For the NTN, both speaker and anti-speaker speech data are used to obtain the hyperplanes that partition the feature space into feature and anti-speaker feature vectors. The anti-speaker data corresponds to the subwords of other speakers enrolled in the database from which the extracted feature vectors are close to the feature vectors of the subword of the speaker being trained.

In the testing phase, the test utterance is first segmented by blind segmentation [96] as is done for training. Consider one of the overall models obtained during training that have various sub-word models. The feature vectors from each sub-word are scored by the corresponding sub-word model. This score is accumulated across all sub-word models to get a total score for the overall model. Similarly, a total score is obtained for each overall model. These scores are averaged to yield a final score for the model. The final score is compared to a threshold to render an acceptance or rejection. There are two kinds of errors, namely, false accept (FA) and false reject (FR). A false accept arises when an imposter is accepted as the true speaker. A false reject arises when a true speaker is rejected or deemed an imposter. Varying the threshold that the final score is compared to accomplishes a tradeoff between the FA and FR. If we need to decrease either of these two types of errors, we end up increasing the other. The equal error rate (EER) of the system is when the FA and FR are equal and is commonly used to measure speaker verification system performance. A receiver operating curve (ROC) curve is a plot of FA versus FR for various thresholds. A user can adjust the threshold to get a desired FA and FR.

Table 1
Equal error rates (EER) for best single, dual and trio models [84]

Database	Best single	Best dual	Best trio
Landline	1.46% (NTN)	1.15% (HMM-NTN)	0.84%
Cellular	5.99% (DTW)	3.71% (DTW-NTN)	3.71%
Multimedia	2.27% (DTW)	0.12% (DTW-NTN)	0.03%

Table 2
Equal error rates (EER) for different weight selection methods given a 3 model system [84]

Database	Equal weights	Fisher	Exhaustive
Landline	0.99%	0.95%	0.84%
Cellular	4.27%	3.75%	3.71%
Multimedia	0.10%	0.85%	0.03%

Given the training and testing methods, the next issue is to choose what types of models to use, see if fusion helps and if so, what fusion rule to use. Experiments with various databases have shown that the NTN, HMM and DTW models generally show the best performance [80,84]. Classifier fusion is clearly achievable due to model diversity. The linear and log opinion pools are usually equal in performance and consistently outperform the voting approach [80]. Of the three considered models, the issue of how many and which to combine is a significant issue. Also, the issue of how to choose the fusion weights must be resolved. To first understand which models to combine, an exhaustive search of fusion weights for a linear opinion pool was done on three databases [84]. The "landline" database comprised of speech collected over a standard landline telephone [84]. The "cellular" database comprised of speech collected in a cellular environment [84]. The "multimedia" database comprised of speech collected over a PC microphone [84]. Table 1 gives the results in terms of the EER for the best single, dual and trio model systems.

The results in Table 1 encourages the use of all three models. In selecting the linear fusion weights, two strategies were considered [84]. The first strategy is to assign equal weights to each model. The second is to use the Fisher discriminant approach [23,84]. Table 2 shows the corresponding results.

9. Summary

This tutorial has presented a review of the classifier based methods used for speaker recognition. Both unsupervised and supervised classifiers are described. In addition, practical approaches that utilize diversity, redundancy and fusion strategies are discussed with the aim of improving performance.

References

[1] B.S. Atal, Automatic recognition of speakers from their voices, Proc. IEEE 64 (1976) 460–475.

[2] G.R. Doddington, Speaker recognition—identifying people by their voices, Proc. IEEE 73 (1985) 1651–1664.

[3] A.E. Rosenberg, Automatic speaker verification: A review, Proc. IEEE 64 (1976) 475–487.

[4] A.E. Rosenberg, F.K. Soong, Recent research in automatic speaker recognition, in: S. Furui, M.M. Sondhi (Eds.), Advances in Speech Signal Processing, Marcel Dekker, New York, 1991, pp. 701–738.

[5] J.P. Campbell, Speaker recognition: A tutorial, Proc. IEEE 85 (1997) 1437–1462.

[6] D. O'Shaughnessy, Speech Communication: Human and machine, Addison-Wesley, Reading, MA, 1987.

[7] D. O'Shaughnessy, Speaker recognition, IEEE Acoust. Speech and Signal Proc. Mag., October 1986, pp. 4–16.

[8] L.R. Rabiner, R.W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, Englewood Cliffs, NJ, 1978.

[9] R.J. Mammone, X. Zhang, R.P. Ramachandran, Robust speaker recognition—A feature based approach, IEEE Signal Proc. Mag. 13 (1996) 58–71.

[10] S.F. Boll, Suppression of acoustic noise in speech using spectral subtraction, IEEE Trans. Acoust. Speech Signal Proc. ASSP-27 (1979) 113–120.

[11] J.R. Deller, J.G. Proakis, J.H. Hansen, Discrete-time Processing of Speech Signals, Macmillan, New York, 1993.

[12] D. Naik, K.T. Assaleh, R.J. Mammone, Robust speaker identification using pole filtering, ESCA Workshop on Automatic Speaker Recognition, 1994.

[13] X. Zhang, R.J. Mammone, Channel and noise normalization using affine transformed cepstrum, International Conference on Spoken Language Proc., Philadelphia, Pennsylvania, October 1996.

[14] M.G. Rahim, B.-H. Juang, Signal bias removal by maximum likelihood estimation for robust telephone speech recognition, IEEE Trans. Speech Audio Process. 4 (1996) 19–30.

[15] A. Sankar, C.-H. Lee, A maximum-likelihood approach to stochastic matching for robust speech recognition, IEEE Trans. Speech Audio Process. 4 (1996) 190–202.

[16] L. Neumeyer, M. Weintraub, Probabilistic optimum filtering for robust speech recognition. IEEE International Conference on Acoustic, Speech and Signal Processing, Adelaide, Australia, April 1994, pp. I-417–I-420.

[17] K. Ng, H. Gish, J.R. Rohlicek, Robust mapping of noisy speech parameters for HMM word spotting, IEEE International Conference on Acoustic, Speech and Signal Processing, San Francisco, California, March 1992, pp. II-109–II-112.

[18] D. Mansour, B.-H. Juang, A family of distortion measures based upon projection operation for robust speech recognition, IEEE Trans. Acoust. Speech Signal Process. 37 (1989) 1659–1671.

[19] B.S. Atal, Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification, J. Acoust. Soc. Amer. 55 (1974) 1304–1312.

[20] K.T. Assaleh, R.J. Mammone, New LP-derived features for speaker identification, IEEE Trans. Speech Audio Process. 2 (1994) 630–638.

[21] M.S. Zilovic, R.P. Ramachandran, R.J. Mammone, Speaker identification based on the use of robust cepstral features obtained from pole-zero transfer functions, IEEE Trans. Speech Audio Process. 6 (1998) 260–267.

[22] H. Gish, M. Schmidt, Text-independent speaker identification, IEEE Signal Proc. Mag. 11 (1994) 18–32.

[23] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.

[24] J.B. Attili, M. Savic, J.P. Campbell Jr., A TMS32020 based real time, text-independent, automatic speaker verification system, IEEE International Conference on Acoustic, Speech and Signal Processing, New York, NY, April 1988, pp. 599–602.

[25] K. Fukunaga, Statistical Pattern Recognition, Academic Press, New York, 1990.

[26] T. Hastie, R. Tibshirani, Discriminant adaptive nearest neighbor classification, IEEE Trans. Pattern Anal. Machine, Intelligence 18 (1996) 607–616.

[27] K.R. Farrell, R.J. Mammone, K.T. Assaleh, Speaker recognition using neural networks and conventional classifiers, IEEE Trans. Speech Audio Process. 2 (1994) 194–205.

[28] A.L. Higgins, L.G. Bahler, J.E. Porter, Voice identification using nearest neighbor distance measure, IEEE International Conference on Acoustic, Speech and Signal Processing, Minneapolis, Minnesota, April 1993, pp. II-375–II-378.

[29] L.G. Bahler, J.E. Porter, A.L. Higgins, Improved voice identification using a nearest neighbor distance measure, IEEE International Conference on Acoustic, Speech and Signal Processing, Adelaide, Australia, April 1994, pp. I-321–I-323.

[30] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, Dordrecht, 1991.

[31] J. Makhoul, S. Roucos, H. Gish, Vector quantization in speech coding, Proc. IEEE 73 (1985) 1551–1588.

[32] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizer design, IEEE Trans. Comm. COM-28 (1980) 84–95.

[33] T. Kohonen, The self-organizing map, Proc. IEEE 78 (1990) 1464–1480.

[34] J. He, L. Liu, G. Palm, A discriminative training algorithm for VQ-based speaker identification, IEEE Trans. Speech Audio Process. 7 (1999) 353–356.

[35] F.K. Soong, A.E. Rosenberg, L.R. Rabiner, B.-H. Juang, A vector quantization approach to speaker recognition, IEEE International Conference on Acoustic, Speech and Signal Processing, Tampa, Florida, March 1985, pp. 387–390.

[36] A.E. Rosenberg, F.K. Soong, Evaluation of a vector quantization talker recognition system in text independent and text dependent modes, Comp. Speech Lang. 22 (1987) 143–157.

[37] A.L. Higgins, L. Bahler, J. Porter, Speaker verification using randomized phrase prompting, Digital Signal Process. 1 (1991) 89–106.

[38] A.E. Rosenberg, J. Delong, C.H. Lee, B.-H. Juang, F.K. Soong, The use of cohort normalized scores for speaker recognition, International Conference on Spoken Language Processing, October 1992, pp. 599–602.

[39] G.J. McLachlan, K.E. Basford, Mixture Models, Inference and Applications to Clustering, Marcel Dekker, New York, 1988.

[40] D.A. Reynolds, R.C. Rose, Robust text-independent speaker identification using Gaussian mixture speaker models, IEEE Trans. Speech Audio Process. 3 (1995) 72–83.

[41] D.A. Reynolds, Speaker identification and verification using Gaussian mixture speaker models, Speech Commun. 17 (1995) 91–108.

[42] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Royal Statist. Soc. 39 (1977) 1–38.
[43] T.K. Moon, The expectation-maximization algorithm, IEEE Signal Proc. Mag. 13 (1996) 47–60.
[44] D.A. Reynolds, Comparison of background normalization methods for text-independent speaker verification, European Conference on Speech Processing, 1997, pp. 963–966.
[45] D.A. Reynolds, M.A. Zissman, T.F. Quatieri, G.C. O'Leary, B.A. Carlson, The effects of telephone transmission degradations on speaker recognition performance, IEEE International Conference on Acoustic, Speech and Signal Processing, Detroit, Michigan, May 1995, pp. 329–332.
[46] D.A. Reynolds, The effects of handset variability on speaker recognition performance: Experiments on the switchboard corpus, IEEE International Conference on Acoustic, Speech and Signal Processing, Atlanta, Georgia, May 1996, pp. 113–116.
[47] R.C. Rose, E.M. Hofstetter, D.A. Reynolds, Integrated models of signal and background with application to speaker identification in noise, IEEE Trans. Speech Audio Process. 2 (1994) 245–257.
[48] S. Furui, Cepstral analysis technique for automatic speaker verification, IEEE Trans. Acoust. Speech and Signal Process. ASSP-29 (1981) 254–272.
[49] F. Itakura, Minimum prediction residual principle applied to speech recognition, IEEE Trans. Acoust. Speech and Signal Process. ASSP-23 (1975) 67–72.
[50] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Trans. Acoust. Speech Signal Process. ASSP-26 (1978) 43–49.
[51] J.G. Wilpon, L.R. Rabiner, A modified K-means clustering algorithm for use in isolated word recognition, IEEE Trans. Acoust. Speech Signal Process. ASSP-33 (1985) 587–594.
[52] P.-C. Chang, B.-H. Juang, Discriminative template training for dynamic programming speech recognition, IEEE International Conference on Acoustic, Speech and Signal Processing, San Francisco, California, March 1992, pp. I-493–I-496.
[53] L.R. Rabiner, B.-H. Juang, Fundamentals of Speech Recognition, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[54] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (1989) 257–286.
[55] J. Picone, Continuous speech recognition using hidden Markov models, IEEE Acoust. Speech Signal Process. Mag. 7 (1990) 26–41.
[56] T. Matsui, S. Furui, Comparison of text-independent speaker recognition methods using VQ distortion and discrete/continuous HMMs, IEEE International Conference on Acoustic, Speech and Signal Processing, San Francisco, California, March 1992, pp. II-157–II-160.
[57] T. Matsui, S. Furui, Concatenated phoneme models for text-variable speaker recognition, IEEE International Conference on Acoustic, Speech and Signal Processing, Minneapolis, Minnesota, April 1993, pp. II-391–II-394.
[58] T. Matsui, T. Nishitani, S. Furui, Robust methods of updating model and a priori threshold in speaker verification, IEEE International Conference on Acoustic, Speech and Signal Processing, Atlanta, Georgia, May 1996, pp. 97–100.
[59] O. Siohan, C.-H. Lee, A.C. Surendran, Q. Li, Background model design for flexible and portable speaker verification systems, IEEE International Conference on Acoustic, Speech and Signal Processing, Phoenix, Arizona, March 1999, pp. 825–828.
[60] J.M. Naik, L.P. Netsch, G.R. Doddington, Speaker verification over long distance telephone lines, IEEE International Conference on Acoustic, Speech and Signal Processing, Glasgow, Scotland, May 1989, pp. 524–527.
[61] J. Oglesby, J.S. Mason, Optimization of neural models for speaker identification, IEEE International Conference on Acoustic, Speech and Signal Processing, Albuquerque, New Mexico, April 1990, pp. 261–264.
[62] Y. Bennani, P. Gallinari, A connectionist approach for speaker identification, IEEE International Conference on Acoustic, Speech and Signal Processing, Albuquerque, New Mexico, April 1990, pp. 265–268.
[63] J. Oglesby, J.S. Mason, Radial basis function networks for speaker recognition, IEEE International Conference on Acoustic, Speech and Signal Processing, Toronto, Canada, May 1991, pp. 393–396.
[64] Y. Bennani, P. Gallinari, On the use of TDNN-extracted features information in talker identification, IEEE International Conference on Acoustic, Speech and Signal Processing, Toronto, Canada, May 1991, pp. 385–388.
[65] K.R. Farrell, Text-dependent speaker verification using data fusion, IEEE International Conference on Acoustic, Speech and Signal Processing, Detroit, Michigan, May 1995, pp. 349–352.
[66] H.-S. Liou, R.J. Mammone, A subword neural tree network approach to text-dependent speaker verification, IEEE International Conference on Acoustic, Speech and Signal Processing, Detroit, Michigan, May 1995, pp. 357–360.
[67] D.P. Morgan, C.L. Scofield, Neural Networks and Speech Processing, Kluwer Academic Publishers, Dordrecht, 1991.
[68] R.P. Lippmann, Review of neural networks for speech recognition, Neural Computation 1 (1989) 1–38.
[69] D.E. Rumelhart, J.L. McClelland, Parallel Distributed Processing, MIT Cambridge Press, Cambridge, MA, 1986.
[70] D. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive networks, Complex Systems 3 (1988) 269–303.
[71] A. Waibel, T. Hanazawa, G. Hinton, Phoneme recognition with time delay neural networks, IEEE Trans. Acoust. Speech Signal Process. 37 (1989) 328–339.
[72] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth International Group, Belmont, CA, 1984.
[73] J. Quinlan, Induction of decision trees, Machine Learn. 1 (1986) 81–106.
[74] J. Quinlan, in: G. Gaines, J. Boose (Eds.), Simplifying decision trees in knowledge acquisition for knowledge-based systems, Academic Press, London, 1988.
[75] L. Atlas, R. Cole, Y. Muthusamy, A. Lippman, J. Connor, D. Park, M. El-Sharkawi, R. Marks, A performance comparison of trained multilayer perceptrons and trained classification trees, Proc. IEEE 78 (1990) 1614–1619.
[76] A. Sankar, R.J. Mammone, Growing and pruning neural tree networks, IEEE Trans. Comput. C-42 (1993) 221–229.
[77] K.R. Farrell, R.P. Ramachandran, M. Sharma, R.J. Mammone, Sub-word speaker verification using data fusion methods, IEEE Workshop on Neural Networks for Signal Processing, Amelia Island Plantation, Florida, September 1997, pp. 531–540.

[78] P.A. Lachenbruch, M.R. Mickey, Estimation of error rates in discriminant analysis, Technometrics 10 (1968) 1–11.
[79] F.K. Soong, A.E. Rosenberg, On the use of instantaneous and transitional spectral information in speaker recognition, IEEE Trans. Acoust. Speech, Signal Process. ASSP-36 (1988) 871–879.
[80] K.R. Farrell, R.P. Ramachandran, R.J. Mammone, An analysis of data fusion methods for speaker verification, IEEE International Conference on Acoustic, Speech and Signal Processing, Seattle, Washington, May 1998, pp. 1129–1132.
[81] K.R. Farrell, S.V. Kosonocky, R.J. Mammone, Neural tree network/vector quantization probability estimators for speaker recognition, Proceedings Neural Networks for Signal Processing, 1994.
[82] J. Schalkwyk, N. Jain, E. Barnard, Speaker verification with low storage requirements, IEEE International Conference on Acoustic, Speech and Signal Processing, Atlanta, Georgia, May 1996, pp. 693–696.
[83] M. Sharma, R.J. Mammone, Subword-based text-dependent speaker verification system with user-selectable passwords, IEEE International Conference on Acoustic, Speech and Signal Processing, Atlanta, Georgia, May 1996, pp. 93–96.
[84] K.R. Farrell, Model combination and weight selection for speaker verification, IEEE Workshop on Neural Networks for Signal Processing, Madison, Wisconsin, August 1999.
[85] J.A. Benediktsson, P.H. Swain, Consensus theoretic classification methods, IEEE Trans. Systems Man Cybernet. 22 (1992) 688–704.
[86] L. Xu, A. Krzyzak, C.Y. Suen, Methods of combining multiple classifiers and their applications to hand-written character recognition, IEEE Trans. Systems Man Cybernet. 23 (1992) 418–435.
[87] T.K. Ho, J.J. Hull, S.N. Srihari, Decision combination in multiple classifier systems, IEEE Trans. Pattern Anal. Machine Intelligence 16 (1994) 66–75.
[88] S.J. Henkind, M.C. Harrison, An analysis of four uncertainty calculi, IEEE Trans. Systems Man Cybernet. 18 (1988) 700–714.
[89] P.D. Gader, M.A. Mohamed, Multiple classifier fusion for handwritten word recognition, IEEE International Conference on Systems, Man and Cybernetics, 1995, pp. 2329–2334.
[90] J. Cao, M. Shridhar, M. Ahmadi, Fusion of classifiers with fuzzy integrals, Third International Conference on Document Analysis and Recognition, 1995, pp. 108–111.
[91] T.D. Pham, H. Yan, Information fusion by fuzzy integral, Australia New Zealand Conference on Intelligent Information Systems, 1996, pp. 191–194.
[92] T.D. Pham, H. Yan, Fusion of handwritten numeral classifiers based on fuzzy and genetic algorithms, Annual Meeting of the North American Fuzzy Information Processing Society, 1997, pp. 257–262.
[93] Y. Lu, F. Yarnaoka, Fuzzy integration of classification results, Pattern Recognition 30 (1997) 1877–1891.
[94] S.-B. Cho, J.H. Kim, Combining multiple neural networks by fuzzy integral for robust classification, IEEE Trans. Systems Man Cybernet. 25 (1995) 380–384.
[95] H. Tahani, J.M. Keller, Information fusion in computer vision using the fuzzy integral, IEEE Trans. Systems Man Cybernet. 20 (1990) 733–741.
[96] M. Sharma, R.J. Mammone, Blind speech segmentation: Automatic segmentation of speech without linguistic knowledge, International Conference on Spoken Language Proc., Philadelphia, Pennsylvania, October 1996.

About the Author—RAVI P. RAMACHANDRAN received the B.Eng. degree (with great distinction) from Concordia University, Montreal, P.Q., Canada, in 1984, and the M.Eng. and Ph.D. degrees from McGill University, Montreal, in 1986 and 1990, respectively. From January to June 1988, he was Visiting Postgraduate Researcher, University of California, Santa Barbara. From October 1990 to December 1992, he worked in the Speech Research Department, AT&T Bell Laboratories, Murray Hill, NJ. From January 1993 to August 1997, he was a Research Assistant Professor at the Center for Advanced Information Processing, Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ. Also, from July 1996 to August 1997, he was a Senior Research Scientist at T-NETIX Inc., Piscataway. Since September 1997, he has been an Associate Professor in the Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ. Dr. Ramachandran has co-edited the book, Modern Methods of Speech Processing (Kluwer Academic Publishers, 1995). He is on the editorial board of the IEEE Circuits and Systems Magazine since 2000. His main research interests are in speech processing, data communications, and digital signal processing.

About the Author—KEVIN R. FARRELL received his Ph.D. in Electrical Engineering from Rutgers University in 1993. His graduate studies focused on pattern recognition and signal processing as applied to speaker recognition. He has since worked with AT&T/Bell Laboratories, Dictaphone, and T-NETIX on topics including language acquisition, speech recognition, and speaker recognition. He has numerous publications and issued patents in these areas. Dr. Farrell is currently with SpeakEZ Incorporated, a wholly-owned subsidiary of T-NETIX. His responsibilities here focus on the commercializaiton of speaker verification technology.

About the Author—ROOPASHRI RAMACHANDRAN received her Bachelors degree in Electrical Engineering (with distinction) from Bangalore University, Bangalore, India in 1990 and her Masters degree in Electrical Engineering from Rutgers University in 1995.

About the Author—RICHARD J. MAMMONE is a Professor of Electrical and Computer Engineering at Rutgers University, Piscataway, NJ. He was also a founder of SpeakEZ, Inc., Piscataway, NJ, and was chief Technical Advisor for T-NETIX, Inc., Englewood, CO. His research areas include speech processing and neural networks. He is a frequent consultant to industry and government agencies. He has published numerous articles and edited several books and special issues of international journals. Dr. Mammone was the Senior Editor for Chapman & Hall, London, U.K., for neural networks. He is a founding member of the Technical Committee on Neural Networks for the IEEE Signal Processing Society. He has been a Guest Editor of the IEEE Transactions on Speech and Audio Processing. He has also been Associate Editor of Pattern Recognition, IEEE Transactions on Neural Networks, and the IEEE Communications magazine. He is listed in Marquis' Who's Who in the World and Who's Who in Science and Engineering. He holds more than a dozen patents.