Digital Signal Processing **10**, 19–41 (2000)
doi:10.1006/dspr.1999.0361, available online at http://www.idealibrary.com on IDEAL®

Speaker Verification Using Adapted Gaussian Mixture Models [1]

Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn

M.I.T. Lincoln Laboratory, 244 Wood St., Lexington, Massachusetts 02420

E-mail: dar@sst.ll.mit.edu, tfq@sst.ll.mit.edu, rbd@sst.ll.mit.edu

Reynolds, Douglas A., Quatieri, Thomas F., and Dunn, Robert B., Speaker Verification Using Adapted Gaussian Mixture Models, *Digital Signal Processing* **10** (2000), 19–41.

In this paper we describe the major elements of MIT Lincoln Laboratory's Gaussian mixture model (GMM)-based speaker verification system used successfully in several NIST Speaker Recognition Evaluations (SREs). The system is built around the likelihood ratio test for verification, using simple but effective GMMs for likelihood functions, a universal background model (UBM) for alternative speaker representation, and a form of Bayesian adaptation to derive speaker models from the UBM. The development and use of a handset detector and score normalization to greatly improve verification performance is also described and discussed. Finally, representative performance benchmarks and system behavior experiments on NIST SRE corpora are presented. © 2000 Academic Press

Key Words: speaker recognition; Gaussian mixture models; likelihood ratio detector; universal background model; handset normalization; NIST evaluation.

1. INTRODUCTION

Over the past several years, Gaussian mixture models (GMMs) have become the dominant approach for modeling in text-independent speaker recognition applications. This is evidenced by the numerous papers from various research sites published in major speech conferences such as the International Conference on Acoustics Speech and Signal Processing (ICASSP), the European Conference on Speech Communication and Technology (Eurospeech), and the International Conference on Spoken Language Process-

The U.S. Government's right to retain a nonexclusive royalty-free license in and to the copyright covering this paper, for governmental purposes, is acknowledged.

[1] This work was sponsored by the Department of Defense under Air Force Contract F19628-95-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Air Force.

1051-2004/00 $35.00
Copyright © 2000 by Academic Press
All rights of reproduction in any form reserved.

ing (ICSLP), as well as articles in ESCA Transactions on Speech Communications and IEEE Transactions on Speech and Audio Processing. A GMM is used in speaker recognition applications as a generic probabilistic model for multivariate densities capable of representing arbitrary densities, which makes it well suited for unconstrained text-independent applications. The use of GMMs for text-independent speaker identification was first described in [1–3]. An Extension of GMM-based systems to speaker verification was described and evaluated on several publicly available speech corpora in [4, 5].

In more recent years, GMM-based systems have been applied to the annual NIST Speaker Recognition Evaluations (SRE). These systems, fielded by different sites, have consistently produced state-of-the-art performance [6, 7]. In particular, a GMM-based system developed by MIT Lincoln Laboratory [8], employing Bayesian adaptation of speaker models from a universal background model and handset-based score normalization, has been the basis of the top performing systems in the NIST SREs since 1996. The system is referred to as the Gaussian Mixture Model-Universal Background Model (GMM-UBM) speaker verification/detection [2] system. In this paper we describe the development and evaluation of the GMM-UBM system as applied to the NIST SRE corpora for single-speaker detection.

The remainder of this paper is organized as follows. Section 2 describes the basic speaker verification/detection task and the likelihood ratio detector approach used to address it. In Section 3 the main components of the GMM-UBM system are described. This section also presents the use of a handset detector and score normalization technique known as HNORM which greatly improves performance when training and testing with different microphones. Section 4 presents experiments and results of the GMM-UBM system using the NIST SRE corpora. Finally, conclusions and future directions are given in Section 5.

2. LIKELIHOOD RATIO DETECTOR

Given a segment of speech, Y, and a hypothesized speaker, S, the task of speaker detection, also referred to as verification, is to determine if Y was spoken by S. An implicit assumption often used is that Y contains speech from only one speaker. Thus, the task is better termed single-speaker detection. If there is no prior information that Y contains speech from a single speaker, the task becomes multispeaker detection. In this paper we will focus on the core single-speaker detection task. Discussion of systems that handle the multispeaker detection task can be found in [9].

[2] We will use the terms verification and detection interchangeably in this paper.

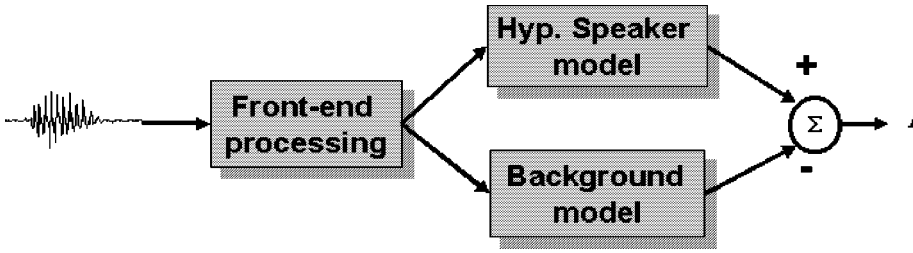


FIG. 1. Likelihood ratio-based speaker detection system.

The single-speaker detection task can be restated as a basic hypothesis test between

H_0: Y is from the hypothesized speaker S

and

H_1: Y is not from the hypothesized speaker S.

The optimum test [3] to decide between these two hypotheses is a likelihood ratio test given by

$$\frac{p(Y \mid H_0)}{p(Y \mid H_1)} \begin{cases} \geq \theta & \text{accept } H_0 \\ < \theta & \text{reject } H_0, \end{cases} \tag{1}$$

where $p(Y \mid H_i)$, $i = 0, 1$, is the probability density function for the hypothesis H_i evaluated for the observed speech segment Y, also referred to as the *likelihood* of the hypothesis H_i given the speech segment. [4] The decision threshold for accepting or rejecting H_0 is θ. The basic goal of a speaker detection system is to determine techniques to compute values for the two likelihoods, $p(Y \mid H_0)$ and $p(Y \mid H_1)$.

Figure 1 shows the basic components found in speaker detection systems based on likelihood ratios. The role of the front-end processing is to extract from the speech signal features that convey speaker-dependent information. In addition, techniques to minimize confounding effects from these features, such as linear filtering or noise, may be employed in the front-end processing. The output of this stage is typically a sequence of feature vectors representing the test segment, $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$, where $\boldsymbol{x}_t$ is a feature vector indexed at discrete time $t \in [1, 2, \ldots, T]$. There is no inherent constraint that features extracted at synchronous time instants be used; as an example, the overall speaking rate of an utterance could be invoked as a feature. These feature vectors are then used to compute the likelihoods of H_0 and H_1. Mathematically, H_0 is represented by a model denoted λ_{hyp} that characterizes the hypothesized speaker S in the feature space of $\boldsymbol{x}$. For example, one could assume that a Gaussian distribution best represents the distribution of feature vectors for H_0 so that λ_{hyp} would be

[3] Strictly speaking, the likelihood ratio test is only optimal when the likelihood functions are known exactly. In practice this is rarely the case.

[4] $p(A \mid B)$ is referred to as a likelihood when B is considered the independent variable in the function.

denoting the mean vector and covariance matrix parameters of the Gaussian distribution. The alternative hypothesis, H_1, is represented by the model $\lambda_{\overline{\text{hyp}}}$. The likelihood ratio statistic is then $p(X \mid \lambda_{\text{hyp}})/p(X \mid \lambda_{\overline{\text{hyp}}})$. Often, the logarithm of this statistic is used giving the log-likelihood ratio

$$\Lambda(X) = \log p(X \mid \lambda_{\text{hyp}}) - \log p(X \mid \lambda_{\overline{\text{hyp}}}). \tag{2}$$

While the model for H_0 is well defined and can be estimated using training speech from S, the model for $\lambda_{\overline{\text{hyp}}}$ is less well defined since it potentially must represent the entire space of possible alternatives to the hypothesized speaker. Two main approaches have been taken for this alternative hypothesis modeling. The first approach is to use a set of other speaker models to cover the space of the alternative hypothesis. In various contexts, this set of other speakers has been called likelihood ratio sets [10], cohorts [11], and background speakers [4]. Given a set of N background speaker models $\{\lambda_1, \ldots, \lambda_N\}$, the alternative hypothesis model is represented by

$$p(X \mid \lambda_{\overline{\text{hyp}}}) = \mathcal{F}(p(X \mid \lambda_1), \ldots, p(X \mid \lambda_N)), \tag{3}$$

where $\mathcal{F}()$ is some function, such as average or maximum, of the likelihood values from the background speaker set. The selection, size, and combination of the background speakers has been the subject of much research (for example, [4, 11, 12]). In general, it has been found that to obtain the best performance with this approach requires the use of speaker-specific background speaker sets. This can be a drawback in applications using a large number of hypothesized speakers, each requiring their own background speaker set.

The second major approach to alternative hypothesis modeling is to pool speech from several speakers and train a single model. Various terms for this single model are a general model [13], a world model, and a universal background model [8]. Given a collection of speech samples from a large number of speakers representative of the population of speakers expected during recognition, a single model, λ_{bkg}, is trained to represent the alternative hypothesis. Research on this approach has focused on selection and composition of the speakers and speech used to train the single model [14, 15]. The main advantage of this approach is that a single speaker-independent model can be trained once for a particular task and then used for all hypothesized speakers in that task. It is also possible to use multiple background models tailored to specific sets of speakers [15, 16]. In this paper we will use a single background model for all hypothesized speakers and we refer to this as the universal background model (UBM).

3. GMM-UBM VERIFICATION SYSTEM

Given the canonical framework for the likelihood ratio speaker detection system, we next describe the specific components of the GMM-UBM system.

3.1. Gaussian Mixture Models

An important step in the implementation of the above likelihood ratio detector is selection of the actual likelihood function, $p(X \mid \lambda)$. The choice of this function is largely dependent on the features being used as well as specifics of the application. For text-independent speaker recognition, where there is no prior knowledge of what the speaker will say, the most successful likelihood function has been Gaussian mixture models. In text-dependent applications, where there is strong prior knowledge of the spoken text, additional temporal knowledge can be incorporated by using hidden Markov models (HMMs) as the basis for the likelihood function. To date, however, use of more complicated likelihood functions, such as those based on HMMs, has shown no advantage over GMMs for text-independent speaker detection tasks as in the NIST SREs.

For a D-dimensional feature vector, $\boldsymbol{x}$, the mixture density used for the likelihood function is defined as

$$p(\boldsymbol{x} \mid \lambda) = \sum_{i=1}^{M} w_i \, p_i(\boldsymbol{x}). \tag{4}$$

The density is a weighted linear combination of M unimodal Gaussian densities, $p_i(\boldsymbol{x})$, each parameterized by a mean $D \times 1$ vector, $\boldsymbol{\mu}_i$, and a $D \times D$ covariance matrix, Σ_i;

$$p_i(\boldsymbol{x}) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\left\{ -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)'(\Sigma_i)^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i) \right\}. \tag{5}$$

The mixture weights, w_i, furthermore satisfy the constraint $\sum_{i=1}^{M} w_i = 1$. Collectively, the parameters of the density model are denoted as $\lambda = \{w_i, \boldsymbol{\mu}_i, \Sigma_i\}$, where $i = 1, \ldots, M$.

While the general model form supports full covariance matrices, i.e., a covariance matrix with all its elements, we use only diagonal covariance matrices in this paper. This is done for three reasons. First, the density modeling of an Mth order full covariance GMM can equally well be achieved using a larger order diagonal covariance GMM.[5] Second, diagonal-matrix GMMs are more computationally efficient than full covariance GMMs for training since repeated inversions of a $D \times D$ matrix are not required. Third, empirically we have observed that diagonal matrix GMMs outperform full matrix GMMs.

Given a collection of training vectors, maximum likelihood model parameters are estimated using the iterative expectation–maximization (EM) algorithm [17]. The EM algorithm iteratively refines the GMM parameters to monotonically increase the likelihood of the estimated model for the observed feature vectors, i.e., for iterations k and $k+1$, $p(X \mid \lambda^{(k+1)}) > p(X \mid \lambda^{(k)})$. Generally, five iterations are sufficient for parameter convergence. The EM equations for training a GMM can be found in [3, 18]. As discussed later, parameters for

[5] GMMs with $M > 1$ using diagonal covariance matrices can model distributions of feature vectors with correlated elements. Only in the degenerate case of $M = 1$ is the use of a diagonal covariance matrix incorrect for feature vectors with correlated elements.

the UBM are trained using the EM algorithm, but a form of Bayesian adaptation is used for training speaker models.

Usually, the feature vectors of X are assumed independent, so the log-likelihood of a model λ for a sequence of feature vectors, $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$, is computed as

$$\log p(X \mid \lambda) = \sum_{t=1}^{T} \log p(\boldsymbol{x}_t \mid \lambda), \tag{6}$$

where $p(\boldsymbol{x}_t \mid \lambda)$ is computed as in Eq. (4). Often, the average log-likelihood value is used by dividing $\log p(X \mid \lambda)$ by T. This is done to normalize out duration effects from the log-likelihood value. Since the incorrect assumption of independence is underestimating the actual likelihood value with dependencies, this scaling factor can also be considered a rough compensation factor to the likelihood value in Eq. (6).

The GMM can be viewed as a hybrid between a parametric and nonparametric density model. Like a parametric model it has structure and parameters that control the behavior of the density in known ways, but without constraints that the data must be of a specific distribution type, such as Gaussian or Laplacian. Like a nonparametric model, the GMM has many degrees of freedom to allow arbitrary density modeling, without undue computation and storage demands. It can also be thought of as a single-state HMM with a Gaussian mixture observation density, or an ergodic Gaussian observation HMM with fixed, equal transition probabilities. Here, the Gaussian components can be considered to be modeling the underlying broad phonetic sounds that characterize a person's voice. A more detailed discussion of how GMMs apply to speaker modeling can be found in [2, 3].

The advantages of using a GMM as the likelihood function are that it is computationally inexpensive, is based on a well-understood statistical model, and, for text-independent tasks, is insensitive to the temporal aspects of the speech, modeling only the underlying distribution of acoustic observations from a speaker. The latter is also a disadvantage in that higher levels of information about the speaker conveyed in the temporal speech signal are not used. The modeling and exploitation of these higher-levels of information may be where approaches based on speech recognition [19] produce benefits in the future. To date, however, these approaches (e.g., large vocabulary or phoneme recognizers) have basically been used only as means to compute likelihood values, without explicit use of any higher-level information such as speaker-dependent word usage or speaking style.

3.2. Front-End Processing

Several processing steps occur in the front-end analysis. First, the speech is segmented into frames by a 20-ms window progressing at a 10-ms frame rate. A speech activity detector is then used to discard silence–noise frames. The speech activity detector is a self-normalizing, energy based detector that tracks the noise floor of the signal and can adapt to changing noise conditions [2, 20]. The speech detector discards 20–25% of the signal from

conversational telephone recordings such as that in the Switchboard databases from which the NIST SRE corpora are derived.

Next, mel-scale cepstral feature vectors are extracted from the speech frames. The mel-scale cepstrum is the discrete cosine transform of the log-spectral energies of the speech segment Y. The spectral energies are calculated over logarithmically spaced filters with increasing bandwidths (mel-filters). A detailed description of the feature extraction steps can be found in [2, 3]. For bandlimited telephone speech, cepstral analysis is performed only over the mel-filters in the telephone passband (300–3400 Hz). All cepstral coefficients except its zeroth value (the DC level of the log-spectral energies) are retained in the processing. Finally, delta cepstra are computed using a first order orthogonal polynomial temporal fit over ± 2 feature vectors (two to the left and two to the right over time) from the current vector [21]. The choice of features is based on previous good performance and results in [22] comparing several standard speech features for speaker identification.

Finally, the feature vectors are channel normalized to remove linear channel convolutional effects. Since we are using cepstral features, linear convolutional effects appear as additive biases. Both cepstral mean subtraction (CMS) and RASTA filtering [23] have been used successfully and, in general, both methods have comparable performance for single speaker detection tasks. When training and recognition speech are collected from different microphones or channels (e.g., different telephone handsets and/or lines), this is a crucial step for achieving good recognition accuracy. However, as seen in several NIST SRE results and reported in [24], this linear compensation does not completely eliminate the performance loss under mismatched microphone conditions. In this paper, we describe one approach to address this remaining mismatch using a normalization of log-likelihood ratio scores [8]. An alternative approach to account specifically for differences in microphone nonlinearities across train and test data is to operate on the waveform with nonlinear transformations, rather than adjusting the log-likelihood ratio scores [25].

3.3. Universal Background Model

In the GMM-UBM system we use a single, speaker-independent background model to represent $p(X \mid \lambda_{\overline{\text{hyp}}})$. The UBM is a large GMM trained to represent the speaker-independent distribution of features. Specifically, we want to select speech that is reflective of the expected alternative speech to be encountered during recognition. This applies to both the type and the quality of speech, as well as the composition of speakers. For example, in the NIST SRE single-speaker detection tests, it is known *a priori* that the speech comes from local and long-distance telephone calls and that male hypothesized speakers will only be tested against male speech. In this case, we would train the UBM used for male tests using only male telephone speech. In the case where there is no prior knowledge of the gender composition of the alternative speakers, we would train using gender-independent speech.

Other than these general guidelines and experimentation, there is no objective measure to determine the right number of speakers or amount of

speech to use in training a UBM. Empirically, from the NIST SRE we have observed no performance loss using a UBM trained with one hour of speech compared to one trained using six hours of speech. In both cases, the training speech was extracted from the same speaker population. Careful experiments controlling the number of speakers present in the UBM training data have not been conducted.

Given the data to train a UBM, there are many approaches that can be used to obtain the final model. The simplest is to merely pool all the data to train the UBM via the EM algorithm (Fig. 2a). One should be careful that the pooled data are balanced over the subpopulations within the data. For example, in using gender-independent data, one should be sure there is a balance of male and female speech. Otherwise, the final model will be biased toward the dominant subpopulation. The same argument can be made for other subpopulations such as speech from different microphones. Another approach is to train individual UBMs over the subpopulations in the data, such as one for male and one for female speech, and then pool the subpopulation models together (Fig. 2b). This approach has the advantages that one can effectively use unbalanced data and can carefully control the composition of the final UBM. Still other approaches can be found in the literature (see, for example, [15, 26]).

Over the past several SREs, our approach has been to train UBMs over subpopulations in the data and then pool the models to create the final UBM (Fig. 2b). For the 1999 NIST SRE we created a gender-independent UBM by training two 1024 mixture GMMs, one for male speech and one for female speech, and then pooling the two models to create our 2048 mixture UBM. We trained these using one hour of speech per gender which was extracted from the 1997 SRE 30-s test files. The speech was equally distributed over carbon-

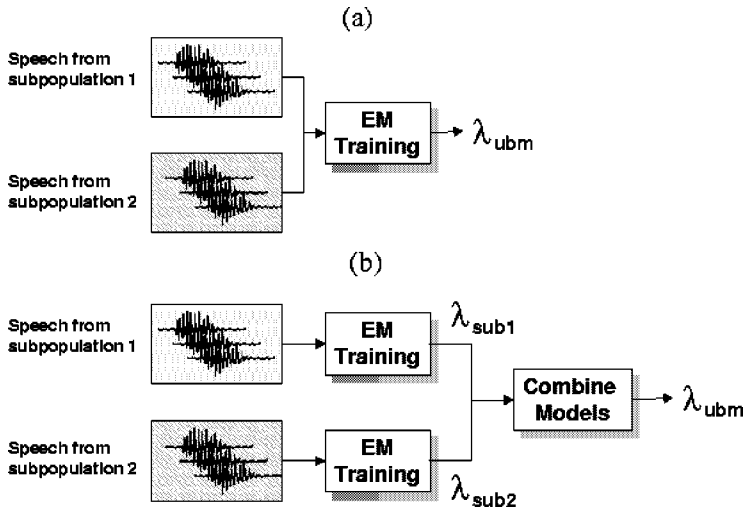


FIG. 2. Data and model pooling approaches for creating a UBM. (a) Data from subpopulations are pooled prior to training the UBM via the EM algorithm. (b) Individual subpopulation models are trained then combined (pooled) to create final UBM.

button and electret handset types (using handset labels provided by NIST). The models were pooled simply by agglomerating the Gaussians and renormalizing the mixture weights.

3.4. Adaptation of Speaker Model

In the GMM-UBM system, we derive the hypothesized speaker model by adapting the parameters of the UBM using the speaker's training speech and a form of Bayesian adaptation [6] [18, 27]. Unlike the standard approach of maximum likelihood training of a model for the speaker independently of the UBM, the basic idea in the adaptation approach is to derive the speaker's model by updating the well-trained parameters in the UBM via adaptation. This provides a tighter coupling between the speaker's model and UBM which not only produces better performance than decoupled models, but, as discussed later in this section, also allows for a fast-scoring technique. Like the EM algorithm, the adaption is a two step estimation process. The first step is identical to the expectation step of the EM algorithm, where estimates of the sufficient statistics [7] of the speaker's training data are computed for each mixture in the UBM. Unlike the second step of the EM algorithm, for adaptation these new sufficient statistic estimates are then combined with the old sufficient statistics from the UBM mixture parameters using a data-dependent mixing coefficient. The data-dependent mixing coefficient is designed so that mixtures with high counts of data from the speaker rely more on the new sufficient statistics for final parameter estimation and mixtures with low counts of data from the speaker rely more on the old sufficient statistics for final parameter estimation.

The specifics of the adaptation are as follows. Given a UBM and training vectors from the hypothesized speaker, $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$, we first determine the probabilistic alignment of the training vectors into the UBM mixture components (Fig. 3a). That is, for mixture i in the UBM, we compute

$$\Pr(i \mid \boldsymbol{x}_t) = \frac{w_i p_i(\boldsymbol{x}_t)}{\sum_{j=1}^{M} w_j p_j(\boldsymbol{x}_t)}. \tag{7}$$

We then use $\Pr(i \mid \boldsymbol{x}_t)$ and $\boldsymbol{x}_t$ to compute the sufficient statistics for the weight, mean, and variance parameters: [8]

$$n_i = \sum_{t=1}^{T} \Pr(i \mid \boldsymbol{x}_t) \tag{8}$$

$$E_i(\boldsymbol{x}) = \frac{1}{n_i} \sum_{t=1}^{T} \Pr(i \mid \boldsymbol{x}_t) \boldsymbol{x}_t \tag{9}$$

[6] This is also known as Bayesian learning or *maximum a posteriori* (MAP) estimation. We use the term Bayesian adaptation since, as applied to the speaker-independent UBM to estimate the speaker-dependent model, the operation closely resembles speaker adaptation used in speech recognition applications.

[7] These are the basic statistics needed to be estimated to compute the desired parameters. For a GMM mixture, these are the count and the first and second moments required to compute the mixture weight, mean, and variance.

[8] $\boldsymbol{x}^2$ is shorthand for $\mathrm{diag}(\boldsymbol{x}\boldsymbol{x}')$.

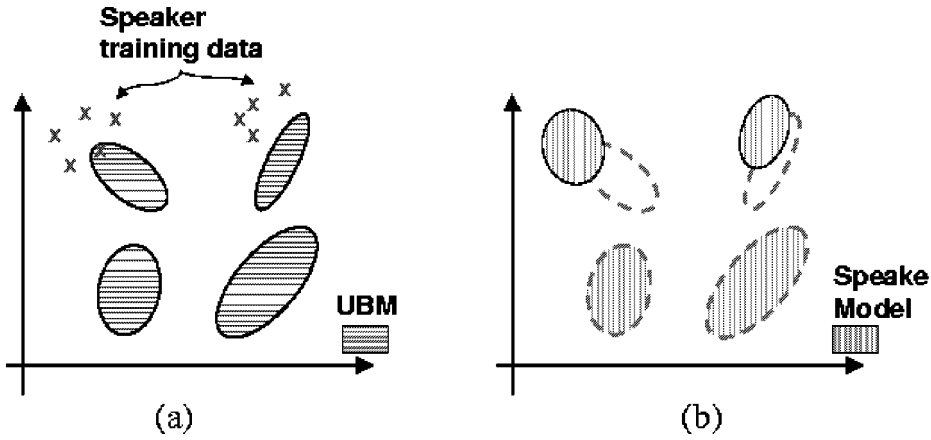


FIG. 3. Pictorial example of two steps in adapting a hypothesized speaker model. (a) The training vectors (x's) are probabilistically mapped into the UBM mixtures. (b) The adapted mixture parameters are derived using the statistics of the new data and the UBM mixture parameters. The adaptation is data dependent, so UBM mixture parameters are adapted by different amounts.

$$E_i(\boldsymbol{x}^2) = \frac{1}{n_i}\sum_{t=1}^{T} \Pr(i \mid \boldsymbol{x}_t)\boldsymbol{x}_t^2. \tag{10}$$

This is the same as the expectation step in the EM algorithm.

Finally, these new sufficient statistics from the training data are used to update the old UBM sufficient statistics for mixture i to create the adapted parameters for mixture i (Fig. 3b) with the equations:

$$\hat{w}_i = \left[\alpha_i^w n_i / T + (1-\alpha_i^w) w_i\right]\gamma \tag{11}$$

$$\hat{\boldsymbol{\mu}}_i = \alpha_i^m E_i(\boldsymbol{x}) + (1-\alpha_i^m)\boldsymbol{\mu}_i \tag{12}$$

$$\hat{\boldsymbol{\sigma}}_i^2 = \alpha_i^v E_i(\boldsymbol{x}^2) + (1-\alpha_i^v)(\boldsymbol{\sigma}_i^2 + \boldsymbol{\mu}_i^2) - \hat{\boldsymbol{\mu}}_i^2. \tag{13}$$

The adaptation coefficients controlling the balance between old and new estimates are $\{\alpha_i^w, \alpha_i^m, \alpha_i^v\}$ for the weights, means and variances, respectively. The scale factor, γ, is computed over all adapted mixture weights to ensure they sum to unity. Note that the sufficient statistics, not the derived parameters, such as the variance, are being adapted.

For each mixture and each parameter, a data-dependent adaptation coefficient α_i^ρ, $\rho \in \{w, m, v\}$, is used in the above equations. This is defined as

$$\alpha_i^\rho = \frac{n_i}{n_i + r^\rho}, \tag{14}$$

where r^ρ is a fixed relevance factor for parameter ρ.[9] The parameter updating as described in Eqs. (11)–(14) can be derived from the general MAP estimation equations for a GMM using constraints on the prior distribution described in [27] (Section V, Eqs. (47) and (48)). The parameter updating equation for the

[9] Thanks go to Michael Newman of Dragon Systems for early discussions about the use of a relevance factor in the adaptation coefficient.

weight parameter (Eq. (11)), however, does not follow from the general MAP estimation equations. The MAP-based equation is $\hat{w}_i = (r^w + n_i)/(T + Mr^w)$. It was found experimentally that using this estimate reduced performance compared to using the current weighted average in Eq. (11).

Using a data-dependent adaptation coefficient allows a mixture-dependent adaptation of parameters. If a mixture component has a low probabilistic count, n_i, of new data, then $\alpha_i^\rho \to 0$ causing the deemphasis of the new (potentially undertrained) parameters and the emphasis of the old (better trained) parameters. For mixture components with high probabilistic counts, $\alpha_i^\rho \to 1$, causing the use of the new speaker-dependent parameters. The relevance factor is a way of controlling how much new data should be observed in a mixture before the new parameters begin replacing the old parameters. This approach should thus be robust to limited training data.

The use of parameter-dependent relevance factors (and hence parameter-dependent adaptation coefficients α_i^ρ) further allows tuning of different adaptation rates for the weights, means, and variances. However, experiments reported in [28] found there was only a minor gain in using parameter-dependent adaptation coefficients. In the GMM-UBM system we use a single adaptation coefficient for all parameters ($\alpha_i^w = \alpha_i^m = \alpha_i^v = n_i/(n_i + r)$) with a relevance factor of $r = 16$. Development experiments found performance was rather insensitive to relevance factors in the range (8–20). The only exception to the single adaptation coefficient is for experiments where we only adapt selective parameters. In this case, for the unadapted parameter, ρ, we set $\alpha_i^\rho = 0, \forall i$.

Since the adaptation is data dependent, not all Gaussians in the UBM are adapted during speaker model training. Knowing the amount of unadapted Gaussians can be an important factor in reduced model storage requirements, since it is possible to efficiently store models using only the difference with the UBM. To examine the amount of Gaussians adapted, we looked at 19 male and 19 female models adapted from the gender-independent UBM described in Section 3.3. Each model was adapted with two minutes of data and only means were adapted ($\alpha_i^w = \alpha_i^v = 0$). We computed the Bhattacharyya distance [29] between corresponding Gaussians in the speaker model and UBM and counted the number of zero distances (those Gaussians which were unchanged in adaptation). For the male speakers, we found that 24% of the UBM Gaussians had zero distances compared to 13% for the female speakers. Since the UBM was constructed by agglomerating gender-dependent models, we also looked at the number of zero distances to the gender-dependent subpopulation models in the UBM. The male speakers had the following percentage of zero distances: 7% in the male portion of the UBM and 40% in the female portion of the UBM. The female speakers had 14% zero distances in the male portion of the UBM and 12% in the female portion of the UBM. Thus, the male speakers tended to only adapt mixtures from the male portion of the UBM and the female speakers tended to adapt both male and female Gaussians.

Results published in [8] using 1996 NIST SRE data and comparison to other systems not using adaptation in NIST SREs strongly indicate that the adaptation approach provides superior performance over a decoupled system

where the speaker model is trained independent of the UBM. One possible explanation for the improved performance is that the use of adapted models in the likelihood ratio is not affected by unseen acoustic events in recognition speech. Loosely speaking, if one considers the UBM as covering the space of speaker-independent, broad acoustic classes of speech sounds, then adaptation is the speaker-dependent tuning of those acoustic classes observed in the speaker's training speech. Mixture parameters for those acoustic classes not observed in the training speech are merely copied from the UBM. This means that during recognition, data from acoustic classes unseen in the speaker's training speech produce approximately zero log-likelihood ratio values that contribute evidence neither toward nor against the hypothesized speaker. Speaker models trained using only the speaker's training speech will have low likelihood values for data from classes not observed in the training data thus producing low likelihood ratio values. While this is appropriate for speech not from the speaker, it clearly can cause incorrect values when the unseen data occurs in test speech from the speaker.

3.5. Log-Likelihood Ratio Computation

The log-likelihood ratio for a test sequence of feature vectors X is computed as $\Lambda(X) = \log p(X|\lambda_{\text{hyp}}) - \log p(X|\lambda_{\text{ubm}})$. The fact that the hypothesized speaker model was adapted from the UBM, however, allows a faster scoring method than merely evaluating the two GMMs as in Eq. (6). This fast scoring approach is based on two observed effects. The first is that when a large GMM is evaluated for a feature vector, only a few of the mixtures contribute significantly to the likelihood value. This is because the GMM represents a distribution over a large space but a single vector will be near only a few components of the GMM. Thus, likelihood values can be approximated very well using only the top C best scoring mixture components.

The second observed effect is that the components of the adapted GMM retain a correspondence with the mixtures of the UBM, so that vectors close to a particular mixture in the UBM will also be close to the corresponding mixture in the speaker model. Using these two effects, a fast scoring procedure operates as follows: For each feature vector, determine the top C scoring mixtures in the UBM and compute UBM likelihood using only these top C mixtures. Next, score the vector against only the corresponding C components in the adapted speaker model to evaluate the speaker's likelihood. For a UBM with M mixtures, this requires only $M + C$ Gaussian computations per feature vector compared to $2M$ Gaussian computations for normal likelihood ratio evaluation. When there are multiple hypothesized speaker models for each test segment, the savings become even greater. In the GMM-UBM system, we use a value of $C = 5$.

3.6. Handset Score Normalization

It has been widely observed in the literature that handset [10] variability causes significant performance degradation in speaker recognition systems. Channel

[10] The term handset is used to imply the microphone used in the handset.

compensation in the front-end processing addresses linear channel effects, but there is evidence that handset transducer effects are nonlinear in nature and are thus difficult to remove from the features prior to training and recognition [25, 30]. Because the handset effects remain in the features, the speaker's model will represent the speaker's acoustic characteristics coupled with the distortions caused by the handset from which the training speech was collected. speaker same likelihood the same speaker. The effect is that log-likelihood ratio scores produced from different speaker models can have handset-dependent biases and scales. This is especially problematic when trying to use speaker-independent thresholds in a system, as is the case for the NIST SREs.

To develop and apply a handset-dependent score normalization, we first created a handset detector to label a speech segment as being either from a carbon-button microphone handset (CARB) or an electret microphone handset (ELEC). The handset detector is a simple maximum likelihood classifier in which handset dependent GMMs were trained using the Lincoln Laboratory Handset Database (LLHDB) [31, 32]. A 1024 mixture GMM was trained using speech from 40 speakers spoken over two carbon-button microphone handsets and another 1024 mixture GMM was trained using speech from the same 40 speakers spoken over two electret microphone handsets. Standard linear filtering compensation (cepstral mean subtraction and RASTA filtering) was applied to the features prior to model training. Since the models were trained with speech from the same speakers and had linear filtering effects removed, differences between the models should mainly be attributable to uncompensated transducer effects. A speech segment is then labeled by selecting the most likely model (CARB or ELEC) based on the models' likelihood values. This handset detector has been used by NIST to supply handset information to SRE participants as well as for analysis of results [7].

Using the handset labels, we then developed the handset score normalization known as HNORM. Since it is often problematic to obtain adequate speaker data for both training and development testing, an approach was sought to use only nonspeaker (or imposter) data to estimate normalization parameters. The basic approach is to estimate from development data handset-dependent biases and scales in the log-likelihood ratio scores and then remove these from scores during operation. First, we compute the log-likelihood ratio scores for a hypothesized speaker–UBM model pair from a set of imposter test segments coming from both CARB and ELEC handsets. We assume these scores have a Gaussian distribution and we estimate the handset-dependent means and standard deviations for these scores. To avoid bimodal distributions, the nonspeaker data should be of the same gender as the hypothesized speaker. The hypothesized speaker now has two sets of parameters describing his or her model's response to CARB and ELEC type speech:

$$\{\mu(\text{CARB}), \sigma(\text{CARB}), \mu(\text{ELEC}), \sigma(\text{ELEC})\}.$$

For the 1999 NIST SRE we used 200 30-s speech segments per handset type, per gender derived from the 1998 SRE test corpus. In general, the duration of

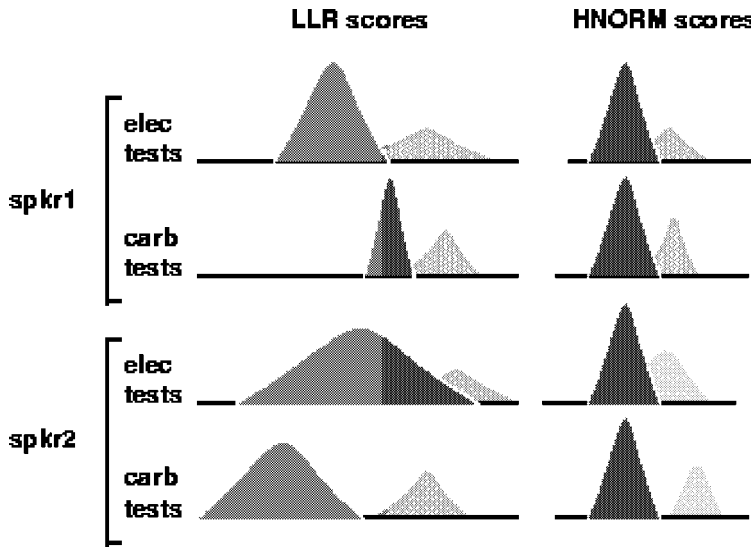


FIG. 4. Pictorial example of HNORM compensation. This picture shows log-likelihood ratio score distributions for two speakers before (left column) and after (right column) HNORM has been applied. After HNORM, the non-speaker score distribution for each handset type has been normalized to zero mean and unit standard deviation.

the speech segments used to estimate HNORM parameters should match the expected duration of the test speech segments.

During recognition, the handset detector supplies the handset type of the test segment, X, and HNORM is applied to the log-likelihood ratio score as

$$\Lambda^{\text{HNORM}}(X) = \frac{\Lambda(X) - \mu(HS(X))}{\sigma(HS(X))}, \tag{15}$$

where $HS(X)$ is the handset label for X.

The desired effect of HNORM is illustrated in Fig. 4. This figure shows log-likelihood ratio score distributions for two speakers before (left column) and after (right column) HNORM has been applied. The effect of removing the handset-dependent biases and scales is to normalize the nonspeaker score distributions such that they have zero mean and unit standard deviation for speech from both handset types. This results in better performance when using a single threshold for detection. In addition to removing handset bias and scales, HNORM also helps normalize log-likelihood scores across different speaker models, again resulting in better performance when using speaker-independent thresholds as in the NIST SREs. HNORM is in effect estimating speaker and handset specific thresholds and mapping them into the log-likelihood score domain rather than using them directly.

HNORM is a handset compensation technique that operates in the score domain. Other approaches to handset compensation operate in the signal domain [25] or in the model domain [16]. Since these techniques operate in different domains it is possible to combine them to potentially achieve even better compensation.

4. EXPERIMENTS

In this section, we present experiments and results using the GMM-UBM system. Experiments are conducted on the 1998 summer-development and 1999 NIST SRE corpora using the NIST evaluation paradigm. The NIST SRE plans detailing the evaluation paradigm can be found in [33]. A more complete description of the NIST SRE along with detailed analysis of results can be found in [7, 34]. The experiments presented here show the general effects on performance of various components and parameters of the GMM-UBM system.

The 1998 and 1999 NIST SRE one-speaker corpora are derived from the Switchboard-II corpus and consist of variable length test utterances (0.5–60 s) extracted from conversational telephone speech. For each speaker, the training data consist of two minutes of speech total, one minute from each of two conversations with the same phone number and the same handset type. Previous SRE data are available for training background models or other parameters. For each test file, 11 independent trials by 11 different hypothesized speakers are conducted. Each test file is processed independently. No cross-gender trials are performed; the gender of the hypothesized and test utterance speaker are the same. All scores where the hypothesized speaker is the speaker in the test utterance are pooled into a set of target scores. The other scores are pooled into a set of nontarget scores. This score pooling occurs over both male and female data. In the experiments below, results are from pooling all of the scores together and the results do not use NIST subpartitions of the scores, e.g., primary or secondary conditions, unless otherwise noted. A single, speaker-independent, threshold is swept over the two sets of scores and the probability of miss and probability of false alarm are computed for each threshold. The error probabilities are then plotted as detection error tradeoff (DET) curves [35] to show system performance.

The first set of experiments, which examined the composition of the UBM, were conducted on the 1998 summer-development data. Training data were selected from the 1997 SRE 30-s tests to have one hour of male speech and one hour of female speech, both equally distributed over the CARB and ELEC handset types. Three UBMs were constructed from these data. For the first case, a single 2048 GMM was trained by pooling all the training data together. In the second case, separate 1024 male and female UBMs were trained and used for male and female hypothesized speakers, respectively. Finally, a single 2048 UBM was formed by pooling the gender-dependent UBMs. HNORM was not applied. The DET curves for these three background models are shown in Fig. 5. While the difference is not large, the separate male and female UBMs performed slightly better than the gender-independent models. This is not surprising since there were no cross-gender trails and the gender-dependent models matched the expected nonspeaker composition. The UBM trained from pooled data performed slightly worse than the others and the UBM created by pooling gender-dependent models matches the performance of the other two systems for different operating regions. Overall, the performance of all three UBMs is comparable but the gender-

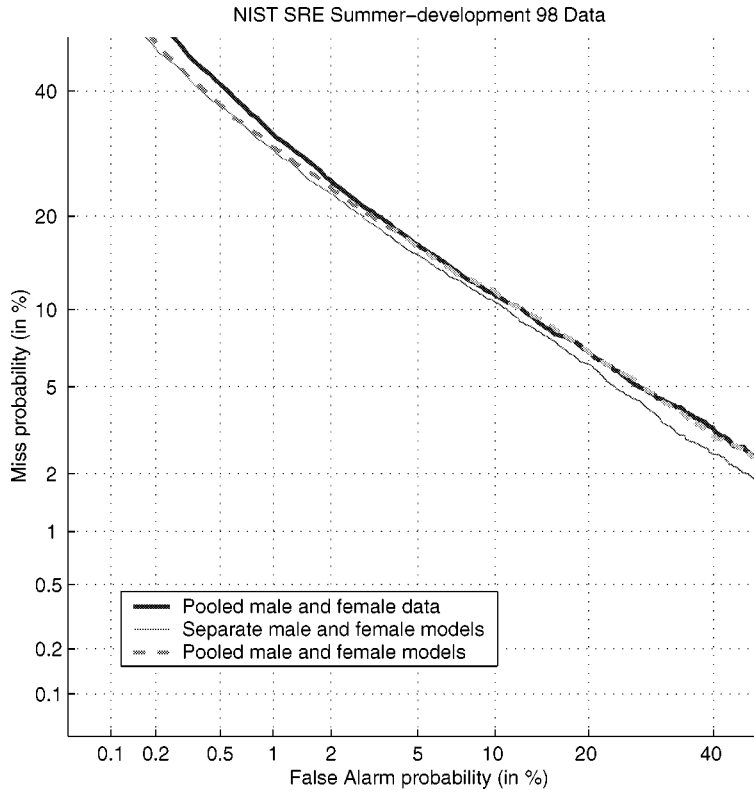


FIG. 5. DET curves for three UBM compositions: Pooled male and female data, separate male and female models, and pooled male and female models. Results are on the NIST 1998 summer-development single-speaker data using all scores.

independent models are perhaps more portable since they can better handle cross-gender nonspeakers. In fact, for the two-speaker detection task where there are cross-gender nonspeakers, the gender-dependent models perform poorly.

The next set of experiments examined the effect of model size on performance. These experiments were conducted on the 1998 summer-development data. Using the same training data as in the previous experiment and pooling the data, UBMs of sizes 16–2048 were trained and evaluated. Again, HNORM was not applied. The DET curves for the different model orders are shown in Fig. 6. From these DETs, it appears that the knee in the performance curve is somewhere around 512 mixtures.

On the 1998 summer-development data we also ran experiments to examine the effects of adapting different sets of parameters when creating hypothesized speaker models. In Fig. 7, we show DET curves for all combinations of parameter adaptations. The curves show that adapting the weights degrades performance in the operating region above 10% miss and below 10% false alarm. It is not clear at this time why adapting the weights degrades performance. The best overall performance is from adapting only the mean vectors.

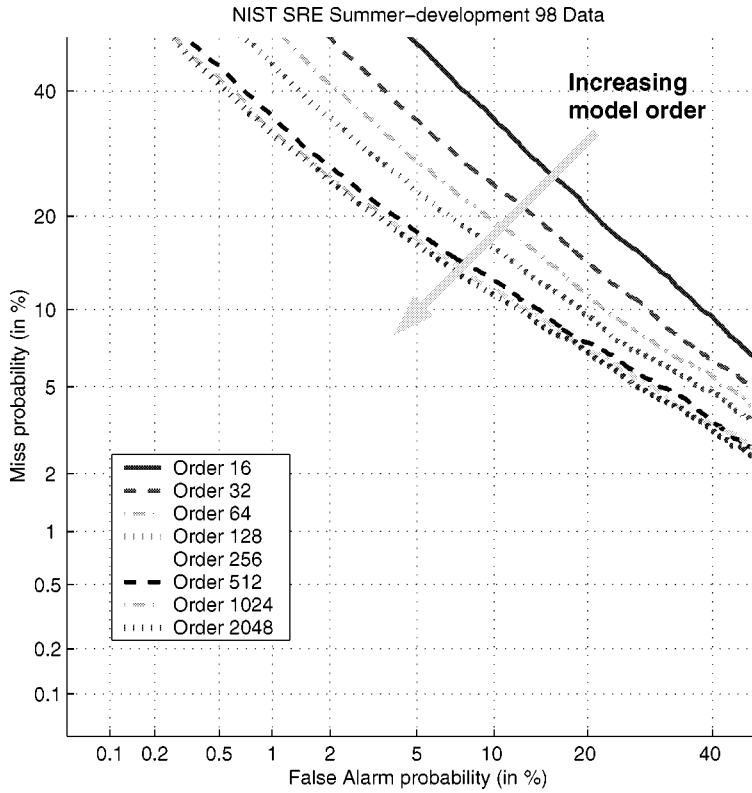


FIG. 6. DET curves for systems using UBMs with 16–2048 mixtures. Results are on the NIST 1998 summer-development single-speaker data using all scores.

In the last set of experiments we compared performance of the GMM-UBM with and without using HNORM. The data for this experiment are the 1999 NIST SRE single-speaker data. This is in contrast to the previous experiments that used the NIST 1998 summer-development data for designing the recognition system in preparation for the 1999 NIST SRE. The UBM used in these experiments is a 2048 mixture, gender-independent GMM created by pooling 1024 mixture gender-dependent models. The hypothesized speaker models were trained by adapting only the mean vectors. In Fig. 8 we show the DET curves for a system with and without HNORM scoring all test files in the 1999 NIST SRE one-speaker data set. The system using HNORM was the primary system for the 1999 NIST SRE single-speaker detection task submitted by MIT Lincoln Laboratory. As evidenced in the DET plot and confirmed by several other sites, HNORM provides a substantial performance increase over the baseline system at all operating points.

In Fig. 9, we show DET curves from a more detailed breakdown of the data set. We have pooled the files into three sets based on the phone number and handset type associated with the test file. In the same-number, same-type (SNST) set, only scores from trials where the test file had the same handset type as the training data for the hypothesized speaker are used. In addition, for

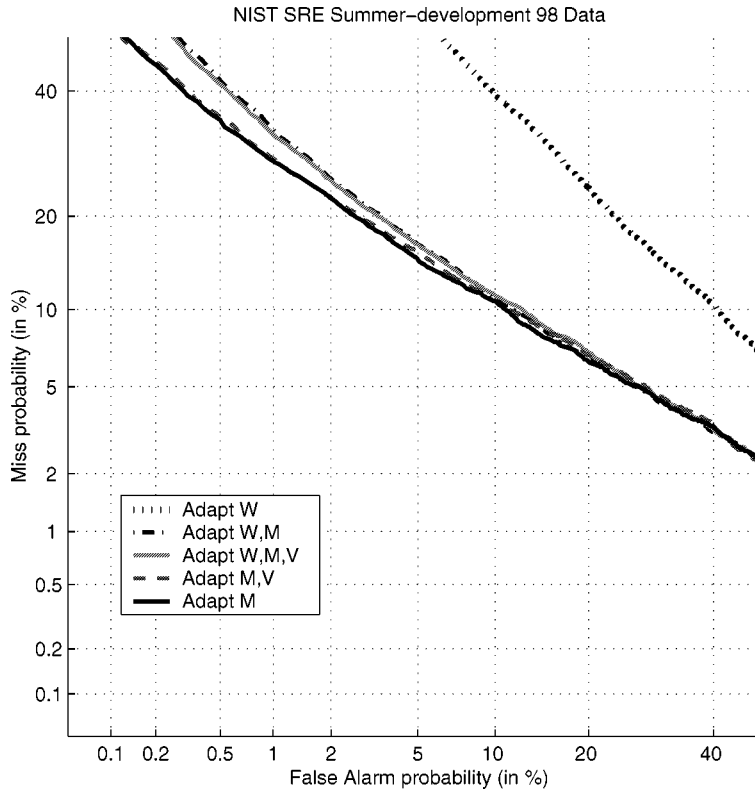


FIG. 7. DET curves for adaptation of different combinations of parameters. W = weights, M = means, V = variances. Results are on the NIST 1998 summer-development single-speaker data using all scores.

true trials (where the speaker in the test file was the same as the hypothesized speaker) only files originating from the same phone number as that used for the hypothesized speaker's training data are used. The SNST pooling shows very optimistic system performance, since all true trials came from identical phone numbers (and presumably identical handsets), while all false trails had same handset types but came from different phone numbers. Thus, microphone effects are actually aiding in correct detections. In the different-number, same-type (DNST) set, only scores from trials where the test file came from a different phone number but had the same handset type as the training data for the hypothesized speaker are used for both true and false trials. In the different-number, different-type (DNDT) set, only scores from trials where the test file came from a different phone number and a different handset type as the training data for the hypothesized speaker are used for both true and false trials. The DNDT is the most stringent evaluation since it has the most degree of mismatch of all poolings. For all three poolings, we show performance of the GMM-UBM system with and without using HNORM. It is clear that while HNORM improves performance under each condition, there remains a large performance gap between matched and mismatched conditions. With

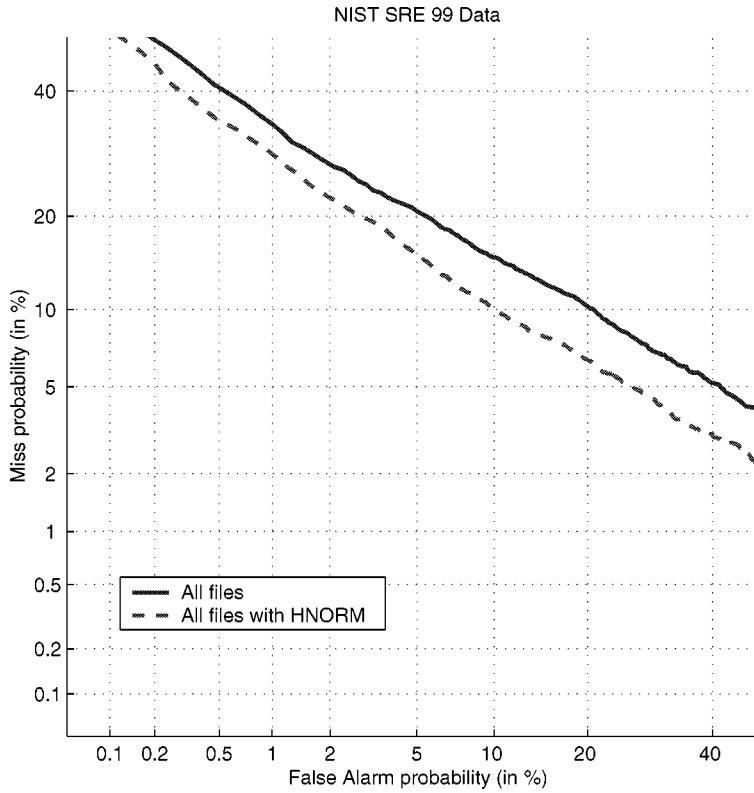


FIG. 8. Comparison of GMM-UBM system with and without HNORM. Results are on the NIST 1999 SRE single-speaker data using all scores.

progressively more degrees of mismatch occurring (from SNST to DNST the actual handset microphone is different for the true trials; from DNST to DNDT the handset types are different for all trials) performance drops off considerably. Although attributed to microphone variability effects, currently there are no precise theories of what effects the microphone transduction is producing in the speech to cause such a performance degradation.

5. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper we have described the major elements of the GMM-UBM system used for high-accuracy speaker recognition. The GMM-UBM system is built around the optimal likelihood ratio test for detection, using simple but effective Gaussian mixture models for likelihood functions, a universal background model for representing the competing alternative speakers, and a form of Bayesian adaptation to derive hypothesized speaker models. The use of a handset detector and score normalization to greatly improve detection performance, independent of the actual detection system, was also described and discussed. Finally, representative performance benchmarks and system

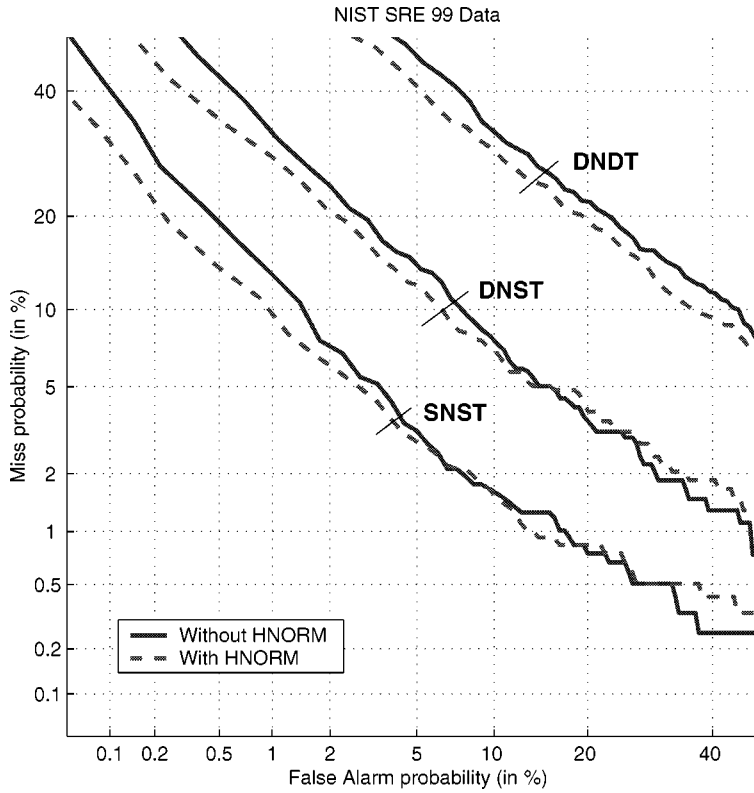


FIG. 9. Comparison of GMM-UBM system with and without HNORM, using different poolings of files in the 1999 NIST SRE single-speaker data set. SNST = Same-Number, Same-Type, DNST = Different-Number, Same-Type, DNDT = Different-Number, Different-Type.

behavior experiments on the 1998 summer-development and 1999 NIST SRE corpora were presented.

While the GMM-UBM system has proven to be very effective for speaker recognition tasks, there are several open areas where future research can improve or build on from the current approach. The first area is dealing better with mismatched conditions. The GMM-UBM system, and all current speaker state-of-the-art recognition systems, rely on low-level acoustic information. Unfortunately, speaker and channel information are bound together in an unknown way in the current spectral-based features and the performance of these systems degrades when the microphone or acoustic environment changes between training data and recognition data. Progress has been made in minimizing this frailty both in addressing linear channel distortion with cepstral mean subtraction and RASTA filtering and in addressing nonlinear effects by normalizing log-likelihood scores (HNORM) and by waveform compensation [25], but there still remains a tremendous performance gap to be bridged between matched and mismatched conditions.

The second area is incorporating higher levels of information, such as speaking style supra-segmental features, or word usage, into the decision

making process. Humans use several levels of information to recognize speakers from speech alone, but automatic systems are still dependent on the low-level acoustic information. The challenges in this area are to find, reliably extract, and effectively use these higher levels of information from the speech signal. It is likely that these higher levels of information will not provide good performance on their own and may need to be fused with more traditional acoustic-based systems. Techniques to fuse and apply high-level asynchronous, or event-based, information with low-level synchronous acoustic features need to be developed in a way that makes the two feature classes work synergistically.

REFERENCES

1. Rose, R. C. and Reynolds, D. A., Text-independent speaker identification using automatic acoustic segmentation. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* 1990, pp. 293–296.
2. Reynolds, D. A., *A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification.* Ph.D. thesis, Georgia Institute of Technology, September 1992.
3. Reynolds, D. A. and Rose, R. C., Robust text-independent speaker identification using Gaussian mixture speaker models, *IEEE Trans. Speech Audio Process.* **3** (1995), 72–83.
4. Reynolds, D. A., Speaker identification and verification using Gaussian mixture speaker models, *Speech Commun.* **17** (1995), 91–108.
5. Reynolds, D. A., Automatic speaker recognition using Gaussian mixture speaker models, *Lincoln Lab. J.* **8** (1996), 173–192.
6. Doddington, G., Przybocki, M., Martin, A., and Reynolds, D. A., The NIST speaker recognition evaluation—overview, methodology, systems, results, perspective, *Speech Commun.*, in press.
7. Martin, A. and Przybocki, M., The NIST 1999 speaker recognition evaluation—an overview, *Digital Signal Process.* **10** (2000), 1–18.
8. Reynolds, D. A., Comparison of background normalization methods for text-independent speaker verification. In *Proceedings of the European Conference on Speech Communication and Technology,* September 1997, pp. 963–966.
9. Dunn, R. B., Reynolds, D. A., and Quatieri, T. F., Approaches to speaker detection and tracking in conversational speech, *Digital Signal Process.* **10** (2000), 93–112.
10. Higgins, A., Bahler, L., and Porter, J., Speaker verification using randomized phrase prompting, *Digital Signal Process.* **1** (1991), 89–106.
11. Rosenberg, A. E., DeLong, J., Lee, C. H., Juang, B. H., and Soong, F. K., The use of cohort normalized scores for speaker verification. In *International Conference on Speech and Language Processing,* November 1992, pp. 599–602.
12. Matsui, T. and Furui, S., Similarity normalization methods for speaker verification based on a posteriori probability. In *Proceedings of the ESCA Workshop on Automatic Speaker Recognition, Identification and Verification,* 1994, pp. 59–62.
13. Carey, M., Parris, E., and Bridle, J., A speaker verification system using alphanets. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* May 1991, pp. 397–400.
14. Matsui, T. and Furui, S., Likelihood normalization for speaker verification using a phoneme- and speaker-independent model, *Speech Commun.* **17** (1995), 109–116.
15. Rosenberg, A. E. and Parthasarathy, S., Speaker background models for connected digit password speaker verification. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* May 1996, pp. 81–84.
16. Heck, L. P. and Weintraub, M., Handset-dependent background models for robust text-independent speaker recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* April 1997, pp. 1071–1073.
17. Dempster, A., Laird, N., and Rubin, D., Maximum likelihood from incomplete data via the EM algorithm, *J. Roy. Stat. Soc.* **39** (1977), 1–38.
18. Duda, R. O. and Hart, P. E., *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

19. Newman, M., Gillick, L., Ito, Y., McAllaster, D., and Peskin, B., Speaker verification through large vocabulary continuous speech recognition. In *Proceedings of the International Conference on Spoken Language Processing,* 1996, pp. 2419–2422.
20. Reynolds, D. A., Rose, R. C., and Smith, M. J. T., PC-based TMS320C30 implementation of the Gaussian mixture model text-independent speaker recognition system. In *Proceedings of the International Conference on Signal Processing Applications and Technology,* November 1992, pp. 967–973.
21. Soong, F. K. and Rosenberg, A. E., On the use of instantaneous and transitional spectral information in speaker recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* 1986, pp. 877–880.
22. Reynolds, D. A., Experimental evaluation of features for robust speaker identification, *IEEE Trans. Speech Audio Process.* **2** (1994), 639–643.
23. Hermansky, H., Morgan, N., Bayya, A., and Kohn, P., RASTA-PLP speech analysis technique. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* March 1992, pp. I.121–I.124.
24. Reynolds, D. A., The effects of handset variability on speaker recognition performance: Experiments on the switchboard corpus. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* May 1996, pp. 113–116.
25. Quatieri, T., Reynolds, D. A., and O'Leary, G., Magnitude-only estimation of handset nonlinearity with application to speaker recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* 1998, pp. 745–748.
26. Isobe, T. and Takahashi, J., Text-independent speaker verification using virtual speaker based cohort normalization. In *Proceedings of the European Conference on Speech Communication and Technology,* 1999, pp. 987–990.
27. Gauvain, J. L. and Lee, C.-H., Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, *IEEE Trans. Speech Audio Process.* **2** (1994), 291–298.
28. Vuuren, S., *Speaker Verification in a Time-Feature Space.* Ph.D. thesis, Oregon Graduate Institute, March 1999.
29. Fukunaga, K., *Introduction to Statistical Pattern Recognition.* Academic Press, San Diego, 1972.
30. Reynolds, D. A., Zissman, M., Quatieri, T. F., O'Leary, G., and Carlson, B., The effects of telephone transmission degradations on speaker recognition performance. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* May 1995, pp. 329–332.
31. Reynolds, D. A., HTIMIT and LLHDB: Speech corpora for the study of handset transducer effects. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing,* April 1997, pp. 1535–1538.
32. Linguistic Data Consortium (LDC), Philadelphia, PA. Website: www.ldc.upenn.edu.
33. NIST speaker recognition evaluation plans, Philadelphia, PA. Website: www.nist.gov/speech/test.htm.
34. Pryzbocki, M. and Martin, A., The 1999 NIST speaker recognition evaluation, using summed two-channel telephone data for speaker detection and speaker tracking. In *Proceedings of the European Conference on Speech Communication and Technology,* 1999, pp. 2215–2218.
35. Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Przybocki, M., The DET curve in assessment of detection task performance. In *Proceedings of the European Conference on Speech Communication and Technology,* 1997, pp. 1895–1898.

DOUGLAS REYNOLDS received the B.E.E. (with highest honors) in 1986 and the Ph.D. in electrical engineering in 1992, both from the Georgia Institute of Technology. He joined the Speech Systems Technology Group (now the Information Systems Technology Group) at the Massachusetts Institute of Technology Lincoln Laboratory in 1992. Currently, he is a senior member of the technical staff and his research interests include robust speaker identification and verification, language recognition, speech recognition, and general problems in signal classification. He is a senior member of the IEEE and a member of the IEEE Signal Processing Society Speech Technical Committee.

THOMAS F. QUATIERI received the B.S. (summa cum laude) from Tufts University, Medford, Massachusetts, in 1973, and the S.M., E.E., and Sc.D. from the Massachusetts Institute of

Technology (MIT), Cambridge, Massachusetts, in 1975, 1977, and 1979, respectively. He is currently a senior member of the technical staff at MIT Lincoln Laboratory, Lexington, Massachusetts, involved in digital signal processing for speech and audio modification, coding, and enhancement and for speaker recognition. His interests also include nonlinear system modeling and estimation. He has contributed many publications to journals and conference proceedings, written several patents, and co-authored chapters in numerous edited books. He holds the position of lecturer at MIT, where he has developed the graduate course *Digital Speech Processing*. Dr. Quatieri is the recipient of the 1982 Paper Award of the IEEE Acoustics, Speech, and Signal Processing Society for the paper, "Implementation of 2-D Digital Filters by Iterative Methods." In 1990, he received the IEEE Signal Processing Society's Senior Award for the paper, "Speech Analysis/Synthesis Based on a Sinusoidal Representation," and in 1994 won this same award for the paper, "Energy Separation in Signal Modulations with Application to Speech Analysis," which was also selected for the 1995 IEEE W.R.G. Baker Prize Award. He was a member of the IEEE Digital Signal Processing Technical Committee, he served on the steering committee for the biannual Digital Signal Processing Workshop from 1983 to 1992, and was Associate Editor for the *IEEE Transactions on Signal Processing* in the area of nonlinear systems. He is also a fellow of the IEEE and a member of Sigma Xi and the Acoustical Society of America.

ROBERT DUNN received a B.S. in electrical and computer engineering (with highest honors) from Northeastern University in 1991 and he received a S.M. in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT) in 1995. He joined the Speech Systems Technology Group (now the Information Systems Technology Group) at MIT Lincoln Laboratory in 1991, where he is currently a member of the technical staff. In 1997 and 1998 he worked on the development of speech coding technology for Voxware, Inc. His research interests include speaker identification, low rate speech coding, and audio signal enhancement.