Recent Advancements in Automatic Speaker Authentication

Verbal-information and fixed-phrase approaches for identifying a person via voice are ready for real-world applications.

by QI LI, BIING-HWANG JUANG, CHIN-HUI LEE, QIRU ZHOU, and FRANK K. SOONG

Personal-identification numbers (PINs), passwords, and social-security numbers have been used extensively and have become an almost inseparable part of our modern daily life. They are used to ensure proper access to private information, personal transactions, and for security of computer and communication networks. To further enhance the security and to improve identification accuracy, biometric features such as signature, fingerprint, hand shape, eye iris, and voice have also been used. Among all biometric features, a person's voice is the most convenient one for personal-identification purposes because it is easy to produce, capture, and transmit over the telephone network.

Speaker or voice authentication is the process of authenticating a user via his/her spoken input. Voice authentication obviously can be done by human experts or operators. However, it will cost more and users may have to wait for services. How to automate the authentication procedure and maintain speed and high performance poses a serious technical challenge to speech researchers.

In this article, we focus exclusively on the use of voice for authentication applications and review recent advancements in this area. The technical components in speech recognition and verification systems are reviewed, and we then discuss a speech-verification (SV) system that utilizes stochastic matching to identify a person based on voice characteristics. We also discuss a newly proposed verbal-information verification (VIV) system that verifies identity through the content of the verbal information.

Two Methods of Speaker Authentication

As shown in Fig. 1, the approach to speaker authentication can be categorically divided into two groups: by a speaker's voice characteristics, which leads to speaker recognition, or by the verbal content of an utterance, which leads to VIV.

Speaker Recognition

Speaker recognition, according to its classification nature, includes SV and speaker identification (SID). SV is the process of verifying whether an unknown speaker is the person as claimed; i.e., a yes-no hypothesis testing problem. On the other hand, SID is the process of associating an unknown speaker with a

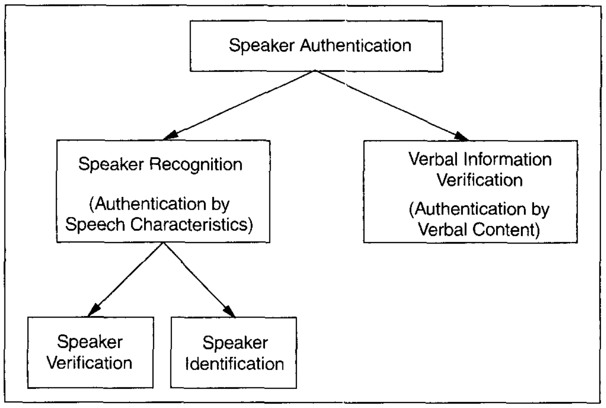
Speaker Authentication
Speaker Recognition
(Authentication by Speech Characteristics)
Verbal Information Verification
(Authentication by Verbal Content)
Speaker Verification
Speaker Identification

Figure 1. *Speaker-authentication approaches.*

1070-9932/99/$10.00©1999IEEE

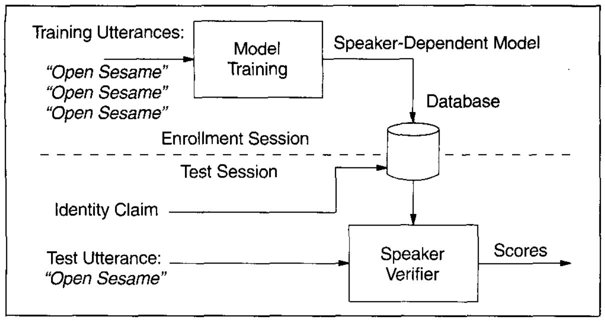

Figure 2. *A speaker-verification system.*

member in a population; i.e., a multiple-choice classification problem.

A typical SV system is shown in Fig. 2, which has two operating scenarios: enrollment and test sessions. A speaker needs to enroll first before he/she can use the system. In an enrollment session, the user's identity, like an account number, is assigned to the speaker, then the person is assigned or asked to select a pass-phrase (e.g., a connected digit string or a phrase, like "open sesame" shown in the figure). The system then prompts the speaker to utter the pass-phrase (the enrollment utterances) several times to allow training or constructing of a speaker-dependent (SD) model that registers the speakers speech characteristics. The speaker who has already enrolled can use the verification system in a future test. In a test, the user first claims his/her identity by entering or speaking the identity information, and the system then prompts the speaker to utter the pass-phrase. The pass-phrase utterance is compared against the already-trained SD model. A speaker is accepted if a decision score exceeds a preset threshold; otherwise, the speaker is rejected.

When the pass-phrases are the same in both training and test, the system is called a fixed pass-phrase system. Frequently, a connected-digit sequence of the telephone or account number is chosen as the fixed pass-phrase. Using a digit string for a pass-phrase has a distinctive difference from other nondigit choices. The high performance of the current connected-digit speech-recognition systems and embedded error-correcting possibilities of digit strings make it feasible that the identity claim can be made via spoken, rather than key-in, input. If such an option is installed, the spoken digit string is first recognized by an automatic speech recognizer (ASR) and the standard verification procedure then follows. Obviously, successful verification of a speaker relies upon a correct recognition of the input digit string.

A safety concern may be raised about using fixed pass-phrases since a spoken pass-phrase can be tape-recorded by impostors and used in later trials to get access to the system. A text-prompted SV system has been proposed to circumvent such a problem. A text-prompted system is made by first training a set of speaker-dependent word or subword models of a small vocabulary, such as digits. When the user tries to access the system, the system prompts the user to utter a randomized sequence of words in the vocabulary. The randomized word sequence is aligned with the pretrained word models and a verification decision is made based upon likelihood scores. Such a text-prompted system normally needs longer enrollment time in order to collect enough data to train SD word or subword models, compared to a fixed-phrase system. The performance of a text-prompted system is generally not as high as that of a fixed-phrase system. This is due to the fact that, unlike a fixed phrase, the co-articulation effect between words is usually undertrained unless enough training data can be collected during enrollment. Details on a text-prompted system and its performance can be found in [1].

The above systems are called text-dependent, or text-constrained, SV systems because the input utterance is constrained, either by a fixed phrase or by a fixed vocabulary. A verification system can also be text-independent. In a text-independent SV system, a speaker's model is trained on the general speech characteristics of a person's voice. Once such a model is trained, the speaker can be verified regardless of the underlying text of the spoken input. Such a system has wide applications in monitoring applications for verifying a speaker on a continuous basis. In order to characterize a speaker's general voice pattern without a text constraint, we normally need a large amount of phonetically or acoustically rich training data in the enrollment procedure. Also, without the text or lexical constraint, longer testing segments are usually needed to maintain satisfactory SV performance. Without a large training set and long testing segments, the performance of a text-independent system is usually inferior to that of a text-dependent system.

In evaluating an SV system, if it is both trained and tested by the same set of speakers, it is called a closed test; otherwise, it is called an open test. In a closed test, the impostors (i.e., all except the true speaker) in the population can be used to train high-performance, discriminant speaker models. However, as most SV applications are open test, to train the discriminant model against all possible impostors is not possible. As an alternative, a set of speakers whose speech characteristics are close to the speaker can be used to train the SD discriminant model, or speaker-independent (SI) models can be used to model impostors.

Verbal-Information Verification

Other than the conventional speaker recognition reviewed in the previous section, speaker authentication can also be approached by VIV. VIV is the process of verifying spoken utterances against the information stored in a given personal data profile. Compared to speaker recognition, VIV is a relatively new concept. We first proposed the idea of VIV in 1997 [2] and gave an update report in 1998 [3]. There are two ways to implement VIV. The input utterance can be verified either through ASR or utterance verification. With ASR, the spoken input is transcribed into a sequence of words. The transcribed words are then compared to the information prestored in the claimed speaker's personal data profile. A verification decision is then made. With utterance verification, the spoken

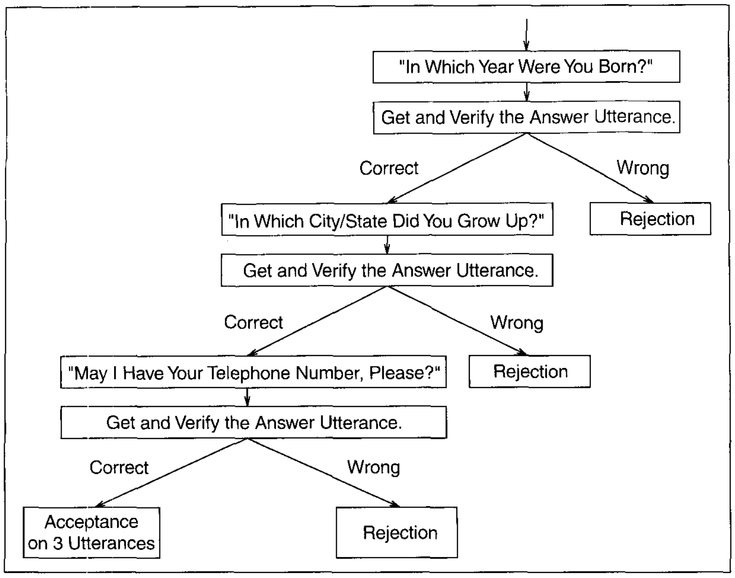
"In Which Year Were You Born?"
Get and Verify the Answer Utterance.
Correct
Wrong
"In Which City/State Did You Grow Up?"
Rejection
Get and Verify the Answer Utterance.
Correct
Wrong
"May I Have Your Telephone Number, Please?"
Rejection
Get and Verify the Answer Utterance.
Correct
Wrong
Acceptance on 3 Utterances
Rejection

Figure 3. *An example of verbal-information verification by asking sequential questions.*

input is verified against an expected sequence of word or subword models [4-8], which are taken from a personal data profile according to the identity claim made by the user.

An example of VIV is shown in Fig. 3. It is similar to a typical telebanking procedure: after an account number is provided, the operator verifies the user by asking some personal information questions, such as mother's maiden name, birth date, address, home telephone number, etc. The user must answer the questions correctly in order to gain access to his/her account. To automate the whole procedure, the questions can be prompted by a text-to-speech system (TTS) or as prerecorded message, then the spoken responses are automatically verified.

A major difference between speaker recognition and VIV in speaker authentication is that a speaker-recognition system utilizes a speaker's speech characteristics represented by the speech feature vectors, while a VIV system mainly inspects the verbal content in the speech signal.

The difference can be further addressed in the following three aspects. First, in a speaker-recognition system, for either SID or SV, we need to train SD models, while in VIV, we only use SI speech models. Second, a speaker-recognition system needs to enroll a new user to train the SD model, while a VIV system does not need such an enrollment. A user's personal data profile is created when the account is set up. Finally, in speaker recognition, the system has the ability to reject an imposter when the input utterances contain a legitimate pass-phrase but fail to match the pretrained SD model. In VIV, it is solely the user's responsibility to protect his/her own personal information because no speaker-specific voice characteristics are used in the verification process.

This dichotomy of VIV and speaker recognition, however, is not rigid. For example, VIV can be used for automatic enrollment of a speaker in a speaker-recognition system and the recorded enrollment data can be used to train SD models. It should also be apparent that by combining the two approaches, a hybrid system can be constructed and further enhancement in performance is possible [3].

Review Of Speaker- and Utterance-Verification Technology

In this section we review the basic building blocks of SV and VIV systems. For VIV, since the most important core technology is utterance verification, this particular module is reviewed in detail. The function blocks reviewed in this section include feature extraction, stochastic models, utterance segmentation, and statistical verification. The applications of these building blocks can be found in Figs. 2, 6, and 8. Feature extraction is always used for any input utterance. A stochastic model is constructed to characterize the feature vectors statistically. Utterance segmentation means to segment an utterance into a sequence of states for likelihood computation using the trained model. Statistical verification includes hypothesis testing for SV and utterance verification.

Feature Extraction

A significant amount of acoustic-phonetic and speaker information is embedded in the short-time spectral envelope of speech signals. This spectral information can be extracted by short-time spectral analysis and compactly represented by cepstral coefficients that are the cosine transform of the log spectrum. Due to its logarithmic nature for controlling the spectral dynamic range and the orthonormality of the cosine basis functions for decorrelating feature components, cepstral coefficients have been widely used as the standard features in both speaker and speech recognition. Given a speech signal sequence s, its cepstral vector $\mathbf{o}_s$ is represented as

$$\mathbf{o}_s = \mathcal{F}^{-1}(\log|\mathcal{F}(s)|), \tag{1}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse. The first 10 to 20 cepstral coefficients provide a compact representation of the succinct spectral properties of speech signals. In the following, we briefly introduce the LPC (linear predicative coding) front-end processor for its popular usage in speech processing, especially in modern digital coding of speech signals. LPC analysis and the corresponding cepstral coefficients are used in all experiments reported in this article.

As shown in Fig. 4, the speech signal is first digitized into a sequence of samples. A sampling rate of 8 kHz is used for telephone speech input. The digitized signal is then pre-emphasized using a first-order digital filter with a coefficient of 0.95 to spectrally flatten the signal and to make it less susceptible to finite precision effects in signal processing. The pre-emphasized speech samples are then blocked into frames

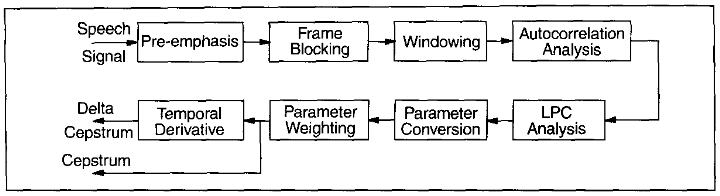

Figure 4. LPC front-end processor for feature extraction.

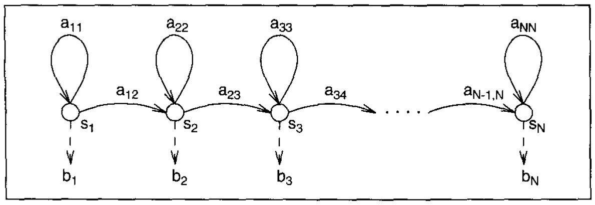

Figure 5. Left-to-right hidden Markov model.

of 30 ms, and frames are shifted every 10 ms. Equivalently, there is an overlap of 20 ms between any two successive frames. Each frame is multiplied by a Hamming window to minimize the signal discontinuities at both ends of each frame and 11 autocorrelation coefficients are derived from the windowed data. After an LPC short-time spectral analysis, which converts each frame of autocorrelations into a set of LPC coefficients, the LPC coefficients are converted to corresponding LPC cepstral coefficients in a closed form. The first 12 coefficients and the energy of frames are kept as features for recognition. LPC cepstral coefficients have been shown to be more appropriate than LPC coefficients for speech and speaker recognition. A bandpass-liftered window is used to weigh the cepstral coefficients for keeping only the most distinctive features in the spectral envelope. Also, in addition to the log energy and the 12 cepstral coefficients, the first and second order of the time derivatives of them, the so called "delta" and "delta-delta" coefficients, are included in the feature set.

Stochastic Models

The feature vectors collected from a speaker's enrollment session are used to construct a statistical model for characterizing a speaker's voice. Several methods have been used. The template method [9] is used to find a "prototypical" sequence of feature vectors, or the template, to represent the utterance of a pass-phrase. During a test, an utterance of the same pass-phrase is compared with the template using a dynamic time warp (DTW) alignment procedure. The vector-quantization (VQ) method [10] uses a speaker-dependent codebook to summarize prototypical feature vectors of a speaker's voice. A codebook is generated by a clustering procedure, which is performed based upon a predefined objective distortion measure for measuring the similarity between any two given vectors and a given training set of feature vectors. A set of centroids (prototypical vectors) are generated by minimizing the total distortions between all training vectors and their corresponding nearest centroids. In a test session, input vectors are compared with the nearest codebook centroids and corresponding distortions are measured to make a recognition decision. The VQ method is simple but it ignores the (time) evolution of the vectors in the feature space or the temporal structure of the underlying utterance. The hidden Markov model (HMM) method can characterize both the temporal structure of the vector sequence and the corresponding statistical variations along the trajectory of an utterance. This is why HMM has become widely used for speech and speaker recognition. We review the HMM as follows.

An HMM is a parametric statistical model. In speech and speaker recognition, an HMM is trained (i.e., parameter estimated) to represent the acoustic pattern of a subword, a word, or a whole pass-phrase. There are many variants of HMMs. The simplest kind is an N-state, left-to-right model without a state skip, as shown in Fig. 5. This is the model used in all the experiments reported in this article. The figure shows a Markov chain with a sequence of states that models the evolution of speech signals. Within each state, an output probability-density function (pdf) is used to statistically characterize the observed speech feature vectors as a multivariate distribution. There are two major forms to model the underlying pdf's: discrete and continuous pdf's. Currently, modeling the output pdf as a mixture of multivariate Gaussian densities is adopted for its better mathematical tractabilities and more parsimonious parameterization.

An HMM λ can be completely characterized by a triple of state transition probabilities A, observation densities B, and initial state probabilities Π, as shown in the following notation:

$$\lambda = \{A,\ B,\ \Pi\} = \{a_{i,j},\ b_i, \pi_i\},\ i, j = 1, \ldots, N, \qquad (2)$$

where N is the total number of states. Given an observation sequence (cepstral vectors) $\mathbf{O} = \{\mathbf{o}_t\}_{t=1}^{T}$, the continuous observation probability density for state j is characterized as a mixture of Gaussian probabilities,

$$b_j(\mathbf{o}_t) = \Pr(\mathbf{o}_t | j) = \sum_{m=1}^{M} c_{jm}\, \mathcal{N}(\mathbf{o}_t; \mu_{jm},\ R_{jm}), \qquad (3)$$

where

$$\mathcal{N}(\mathbf{o}_t; \mu_{jm},\ R_{jm}) = (2\pi)^{-d/2} |R_{jm}|^{-1/2} \cdot \exp\left\{\frac{1}{2}(\mathbf{o}_t - \mu_{jm})^T R_{jm}^{-1} (\mathbf{o}_t - \mu_{jm})\right\} \qquad (4)$$

where M is the total number of the Gaussian components and μ_{jm} and R_{jm} are the d-dimensional mean vector and covariance matrix of the mth component at state j. The mixture weights satisfy the stochastic constraint, $\sum_{m=1}^{M} c_{jm} = 1$.

The model parameters $\{A, B, \Pi\}$ of λ can be trained by iterative methods to satisfy a given optimization criterion. Normally the model parameters are trained to maximize the likelihood, $\Pr(\mathbf{O}|\lambda)$ (see next section for detail), based on a training data set. Iterative procedures such as the Baum-Welch method (also known as the expectation-maximization (EM) method) [11,12] have been used. Other than the maximum-likelihood (ML) criterion, the model can also be trained by optimizing a discriminant function. For example, the minimum-classification-error (MCE) criterion [13,14] was proposed along with a corresponding generalized probabilistic descent (GPD) training algorithm [15,16]. Other criteria, such as maximum mutual information (MMI) [17,18], have also been tried. Instead of just modeling the distribution of the data set of the target class, the criteria also incorporate data sets of other classes. A discriminant model is thus constructed not only to model the underlying distribution of the target class, but to minimize the classification error or to maximize the mutual information between the target class and others. The discriminant training algorithms have been applied successfully to HMM-based speech recognition. The MCE algorithm can also be applied to speaker recognition [19-22]. Generally speaking, the models trained by discriminant objective functions yield better recognition and verification performance.

Authentication performance can be improved by starting with a VIV system, then gradually adapting to an SV system.

Speech Segmentation

Given an HMM λ and a sequence of observations $\mathbf{O} = \{\mathbf{o}_t\}_{t=1}^{T}$, the optimal state segmentation can be determined by evaluating the probabilities of all possible state sequences. This can be done efficiently using the Viterbi algorithm [23, 24]. The likelihood is

$$\Pr\{\mathbf{O}, s_{\max} | \lambda\} = \max_{\{s_t\}} \left\{ \prod_{t=1}^{T} a_{s_t s_{t-1}} b_{s_t}(\mathbf{o}_t) \right\}, \tag{5}$$

where $\max_{\{s_t\}}$ is the Viterbi search to segment the feature vectors into a sequence of states $s_{\max}$ optimally, in the sense of maximum likelihood.

In utterance verification, we assume that the expected word or subword sequence is known and the task is to verify whether the input spoken utterance matches it. Similarly, in SV, the text of the pass-phrase is known. The task is to verify whether the input spoken utterance matches the given sequence, using the model trained by the speaker's voice.

Statistical Verification

The purpose of speaker and utterance verification is to determine whether the given speech samples are from the expected target sources or the alternatives through hypothesis testing. The data from the target source and the data from the alternative source are used here to construct the target and alternative models, respectively, for the test.

In SV, the target model is a speaker-dependent model trained on the speaker's voice. In an open-set test, the alternative model can be an SI model trained from a different database. In a closed-set test, the alternative model can be trained from the data of all the impostors in the set. The alternative model is also called a general background model or cohort model [25].

In utterance verification, a target model is trained for a specific subword, word, or phrase. The alternative model is trained on the data selected from a set of subwords, words, or phrases that are easily confused with the target source in the statistical sense. In other words, the alternative subwords or words are in the neighborhood of the target one in the feature space.

Given the two sets of models, verification can be approached via a statistical hypothesis testing procedure. There are several decision rules for optimal hypothesis testing under different criteria [26, 27]. For the hypothesis testing problem, all the decision rules calculate *likelihood ratio* first, then a decision can be made by comparing the ratio with a preset threshold. Different decision rules lead to different ways of threshold setting. Hypothesis testing can be formulated as follows.

Let $\mathbf{o}_i$ be an observation vector and $p(\mathbf{o}_i|\lambda_t)$ be the conditional density function for the target class and $p(\mathbf{o}_i|\lambda_a)$ for the alternative class. The likelihood ratio and log-likelihood ratio are

$$r(\mathbf{o}_i) = \frac{p(\mathbf{o}_i|\lambda_t)}{p(\mathbf{o}_i|\lambda_a)}, \tag{6}$$

and

$$\mathcal{R}(\mathbf{o}_i) = \log p(\mathbf{o}_i|\lambda_t) - \log p(\mathbf{o}_i|\lambda_a). \tag{7}$$

A decision is made

$$\begin{cases} \text{Acceptance:} & \mathcal{R}(\mathbf{o}_i) \geq T; \\ \text{Rejection:} & \mathcal{R}(\mathbf{o}_i) < T, \end{cases} \tag{8}$$

where T is the *threshold value* for the decision.

For speaker and utterance verifications, the decision is made on a set of observation samples over a part of an utterance or the whole utterance $\mathbf{O} = \{\mathbf{o}_i\}_{i=1}^{n}$, and n is the total number of samples used in the final decision. Following the Neyman-Pearson lemma [28, 29], for a given sequence of n observation vectors, the likelihood ratio $r(\mathbf{O})$ or log-likelihood ratio $\mathcal{R}(\mathbf{O})$ are

$$r(\mathbf{O}) = \prod_{i=1}^{n} \frac{p(\mathbf{o}_i|\lambda_t)}{p(\mathbf{o}_i|\lambda_a)} = \frac{P(\mathbf{O}|\lambda_t)}{P(\mathbf{O}|\lambda_a)}, \tag{9}$$

where $P(\mathbf{O}|\lambda_t)$ and $P(\mathbf{O}|\lambda_a)$ are likelihood for the target and anti-models, and

$$\begin{aligned}\mathcal{R}(\mathbf{O};\lambda_t,\lambda_a) &= \sum_{i=1}^{n} \log\frac{p(\mathbf{o}_i|\lambda_t)}{p(\mathbf{o}_i|\lambda_a)}\\ &= \log P(\mathbf{O}\mid\lambda_t) - \log P(\mathbf{O}\mid\lambda_a).\end{aligned} \tag{10}$$

A decision is made as

$$\begin{cases}\text{Acceptance:} & \mathcal{R}(\mathbf{O}) \geq T;\\ \text{Rejection:} & \mathcal{R}(\mathbf{O}) < T,\end{cases} \tag{11}$$

where T is a threshold value, determined theoretically or experimentally.

There are two kinds of errors in a test: false rejection (rejecting the hypothesis when it is true) and false acceptance (accepting it when it is false). The equal-error rate (EER), or the error rate when the two errors are made equal by adopting the threshold value, in an aposterior sense, is widely used in evaluating verification performance.

A Speaker-Verification System with Stochastic Matching

In this section, we first present a channel-equalization algorithm [30] for SV, then introduce an SV system built on the technology components introduced above. Among the different SV systems introduced earlier, we focus on the fixed-phrase system [30, 31] and evaluate the system in an open-set test. This is due to three reasons. First, a short, user-selected phrase is easy to remember. Second, a fixed-phrase system usually has better performance than a text-prompted system [1]. Last, an open-set evaluation is more appropriate for real applications. In a large-scale, telephone-based banking application, it usually involves a large user population that changes on a daily basis. It is impossible and unrealistic to train each of the SD models using all the other users as impostors.

As shown in Fig. 2, the fixed-phrase system has two phases, enrollment and test. During enrollment, LPC cepstral feature vectors corresponding to the nonsilence portion of the enrollment pass-phrases are used to train an SD HMM, which characterizes the phrase. In addition to model training, the text of the phrase collected from the enrollment session is transcribed into a sequence of phonemes $\{S_k\}_{k=1}^{K}$, where S_k is the kth phoneme and K is the total number of phonemes in the sequence. The models and the transcription are saved in the database along with the channel statistical parameters for channel compensation during test [30], which will be introduced in the next session.

A block diagram of a test session is shown in detail in Fig. 6. It illustrates how the system verifies a spoken pass-phrase. After a speaker claims his or her identity, the system expects the user to speak the same phrase used in the enrollment session. The stochastic matching procedure [30] is first applied to compensate the channel difference between training and testing. The compensated feature vectors are used for computing the target and background scores as shown in the figure.

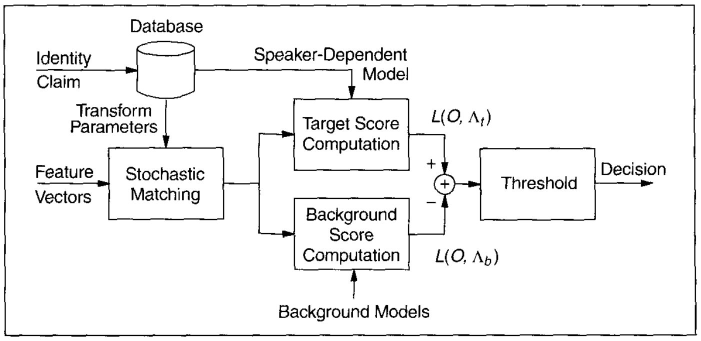

Figure 6. A fixed-phrase speaker-verification system

Fast Stochastic Matching for Channel Equalization

In speaker-recognition experiments via telephone lines, the user may switch between different telephone handsets and transmission lines from one call to another. Possible spectral mismatches between the training and test data can thus seriously deteriorate the recognition performance. This mismatch needs to be equalized. The distortion, as a good approximation, can be modeled as a linear convolution in the time domain, or equivalently, linearly additive in the cepstral domain. To equalize such a channel difference, a long-term cepstral average can be computed and subtracted from the cepstral feature vectors, in both the training and the testing data. The method, commonly called cepstral mean subtraction (CMS) [9, 32-34], has been shown effective and is widely used in both speech- and speaker-recognition systems. Maximum-likelihood approaches [35, 36] were also proposed to estimate the parameters of a linear transform to minimize such a mismatch. More sophisticated scaling and rotational equalization methods have been proposed [30] and they are used in conjunction with the average mean vector.

The mismatch can be modeled as a linear transform in the cepstral domain:

$$y = \mathbf{A}x + b, \tag{12}$$

where x is a vector of the cepstral frame of a test utterance; $\mathbf{A}$ and b are the matrix and vector that need to be estimated for every test utterance; and y is a transformed vector. Geometrically, b represents a translation and $\mathbf{A}$ represents both scaling and rotation. When **A** is diagonal, it is only a scaling operation. An analysis of the reason for using a linear transformation is beyond the scope of this article. Interested readers are referred to [37].

CMS is a fast, efficient technique for handling mismatch in both speaker and speech recognition. It estimates b and assumes **A** to be an identity matrix. In [32], the vector b was estimated by long-term average, short-term average, and an ML approach. In [35, 38], ML approaches were used to estimate b, a diagonal **A**, and model parameters for HMMs for stochastic matching. A least-squares solution of the linear transform parameters was briefly introduced in [37].

In [30], Li, Parthasarathy, and Rosenberg consider a general linear transform; i.e., **A** is a full matrix, and b is a vector. The approach is to have the overall distribution of test data match the overall distribution of training data. Then, an SD HMM trained on the training data is applied to evaluate the details of the test data. This is based on the assumption that differences between speakers are mainly on the details that have been characterized by HMMs. Compared to CMS and other bias-removal techniques [32, 36], the proposed linear transform is more general since CMS and others only consider the translation; compared to the ML approaches [32, 35, 36, 38], the algorithm is simpler and faster since iterative techniques are not required and the estimation of the linear transform parameters is separated from the HMM training and test.

We use Fig. 7 as a geometric interpretation of the proposed matching algorithm. In Fig. 7(a), the dashed line is a contour of training data. In Fig. 7(b), the solid line is a contour of test data. Due to different channels, noise levels, and telephone transducers, the mean of the test data is translated from the training data; the distribution is scaled [39] and rotated from the HMM training condition. Therefore, the mismatch may cause a wrong decision when using the trained HMM to score the mismatched test data. By applying the proposed algorithm, we first find a covariance matrix $\mathbf{R}_{\text{train}}$ from the training data, which characterizes the overall distribution approximately. Then, we find a covariance matrix $\mathbf{R}_{\text{test}}$ from the test data and estimate the parameters of the **A** matrix for the linear transform in Eq. (12). After applying the first transform, the overall distribution of the test data is scaled and rotated to be same as the training data except for the difference of the means, as shown in Fig. 7(c). In the second step, we find the difference of the means and translate the test data to the same location of the training data as shown in Fig. 7(d), where the contour of the transformed test data is more consistent with the contour of the training data.

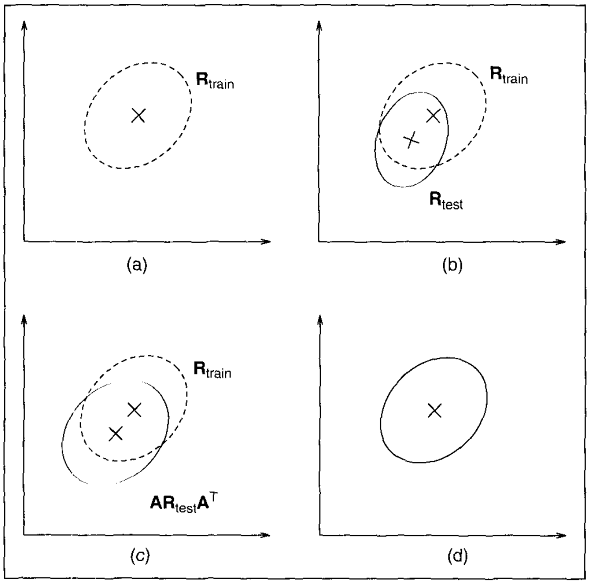

Figure 7. A geometric interpretation of the fast stochastic matching.

This technique attempts to improve mismatch whether the mismatch occurs because test and training conditions differ or because the test and training data originate from different speakers. It is reasonable to suppose that speaker characteristics are found mainly in the details of the representation. However, to the extent that they are also found in global features, this technique would increase the matching scores between true speaker models and imposter test utterances. Performance, then, could possibly degrade, particularly when other sources of mismatch are absent—that is, when test and training conditions are actually matched. However, the experiments in [30] showed that performances overall do improve.

In a training session, we collect multiple utterances with the same content and use a covariance matrix $\mathbf{R}_{\text{train}}$ and a mean vector m_{train} to represent the overall distribution of the training data of all the training utterances in a cepstral domain. They are defined as follows:

$$\mathbf{R}_{\text{train}} = \frac{1}{U} \sum_{i=1}^{U} \frac{1}{N_i} \sum_{j=1}^{N_i} (x_{i,j} - m_i)(x_{i,j} - m_i)^T, \tag{13}$$

and

$$m_{\text{train}} = \frac{1}{U} \sum_{i=1}^{U} m_i, \tag{14}$$

where $x_{i,j}$ is the jth nonsilence frame in the ith training utterance; U is the total number of training utterances; N_i and m_i are the total number of nonsilence frames and the mean vector of the ith training utterance, respectively; and m_{train} is the average mean vector of the nonsilence frames of all training utterances. The nonsilence frames are detected by an endpoint-detection algorithm that will be presented separately.

In a test session, only one utterance will be collected and verified at a time. The covariance matrix for the test data is

$$\mathbf{R}_{\text{test}} = \frac{1}{N_f} \sum_{j=1}^{N_f} (y_j - m_{\text{test}})(y_j - m_{\text{test}})^T, \tag{15}$$

where y_j and m_{test} are a nonsilence frame and the mean vector of the test data, and N_f is the total number of nonsilence frames.

The proposed criterion for parameter estimation is to have $\mathbf{R}_{\text{test}}$ match $\mathbf{R}_{\text{train}}$ through a rotation, scaling, and translation (RST) of the test data. For rotation and scaling, we have the following equation:

$$\mathbf{R}_{\text{train}} - \mathbf{A}\mathbf{R}_{\text{train}}\mathbf{A}^T = 0\,, \tag{16}$$

where $\mathbf{A}$ is defined as in Eq. (12), and $\mathbf{R}_{\text{train}}$ and $\mathbf{R}_{\text{test}}$ are defined as in Eqs. (13) and (15). By solving Eq. (16), we have the $\mathbf{A}$ matrix for Eq. (12),

$$\mathbf{A} = \mathbf{R}_{\text{train}}^{\frac{1}{2}}\mathbf{R}_{\text{test}}^{-\frac{1}{2}}. \tag{17}$$

Then, the translation term b of Eq. (12) can be obtained by

$$b = m_{\text{train}} - m_{rs} = m_{\text{train}} - \frac{1}{N_f}\sum_{j=1}^{N_f}\mathbf{A}x_j \tag{18}$$

where m_{train} is defined as in Eq. (14), m_{rs} is a mean vector of rotated and scaled frames, N_f is the total number of nonsilence frames of a test utterance, and x_j is the jth cepstral vector frame.

To verify a given test utterance against a set of true speaker's models (consisting of an SD HMM plus $\mathbf{R}_{\text{train}}$, m_{train}), first $\mathbf{R}_{\text{test}}$, $\mathbf{A}$, and b are calculated by using Eqs. (15), (17), and (18), then all test frames are transformed by Eq. (12) to reduce the mismatch.

Fixed-Phrase Verification

In the block of target-score computation of Fig. 6, the speech feature vectors are decoded into states by the Viterbi algorithm (Eq. (5)), using the whole-phrase model. A log-likelihood score for the target model (i.e., target score) is calculated as

$$L(\mathbf{O}, \Lambda_t) = \frac{1}{N_f}\log P(\mathbf{O}\;|\Lambda_t)\,, \tag{19}$$

where $\mathbf{O}$ is the set of feature vectors, N_f is the total number of vectors, Λ_t is the target model, and $P(\mathbf{O}|\,\Lambda_t)$ is the likelihood score from the Viterbi decoding.

In the block of the background score computation, a set of SI HMMs in the order of the transcribed phoneme sequence, $\Lambda_b = \{\lambda_1, \ldots, \lambda_K\}$, is applied to align an input utterance with the expected transcription using the Viterbi decoding algorithm. The segmented utterance is $\mathbf{O} = \{\mathbf{O}_1, \ldots, \mathbf{O}_K\}$, where $\mathbf{O}_i$ is the set of feature vectors corresponding to the ith phoneme S_i in the phoneme sequence. The background likelihood score is then computed by

$$L(\mathbf{O}, \Lambda_b) = \frac{1}{N_f}\sum_{i=1}^{K}\log P(\mathbf{O}_i\,|\,\lambda_i), \tag{20}$$

where $\Lambda_b = \{\lambda_i\}_{i=1}^{K}$ is a set of SI phoneme models in the order of the transcribed phoneme sequence, $P(\mathbf{O}_i\,|\,\lambda_{b_i})$ is the corresponding phoneme likelihood score, and K is the total number of phonemes.

The target and background scores are used for the following likelihood-ratio test [31]:

$$\mathcal{R}(\mathbf{O}; \Lambda_t, \Lambda_b) = L(\mathbf{O}, \Lambda_t) - L(\mathbf{O}, \Lambda_b), \tag{21}$$

where $L(\mathbf{O}, \Lambda_t)$ and $L(\mathbf{O}, \Lambda_b)$ are defined in Eqs. (19) and (20), respectively.

The system has been tested on a database consisting of fixed-phrase utterances. The database was recorded over long-distance telephone networks by 100 speakers, 51 male and 49 female. The fixed phrase, common to all speakers, was "I pledge allegiance to the flag" with an average utterance length of 2 seconds. Five utterances of each speaker recorded in one enrollment session (one telephone call) were used to construct an SD target HMM. For testing, we used 50 utterances recorded from a true speaker in different sessions (from different telephone channels and handsets at different times with different background noise), and 200 utterances recorded from 51 or 49 impostors of the same gender in different sessions.

In order to further improve the SD HMM, a procedure was employed for model adaptation. The second, fourth, sixth, and eighth test utterances, which were recorded at different times, from the true speaker were used to update the means and mixture weights of the SD HMM for verifying successive test utterances. For the above database, the average individual equal-error rate over 100 speakers was 2.6% without adaptation and 1.8% with adaptation respectively [30], as shown in Table 1. Usually, the longer the pass-phrase, the higher the accuracy. The response time depends on the hardware/software configuration. For most of the cases, SV time is less than uttering the pass-phrase.

We note that the same pass-phrase was used for all speakers in our evaluation. This should be the lower bound of the performance. The actual system performance would be better when users choose their own, and most likely different, pass-phrase. Also, to ensure the open-test nature, none of the impostor's data was used for training an SD target model by discriminant training.

Table 1. Experimental Results in Average Equal-Error Rates

	Without Adaptation	With Adaptation
Fixed Pass-Phrase Speaker Verification	2.61%	1.80%

(Tested on 100 speakers using one common pass-phrase.)

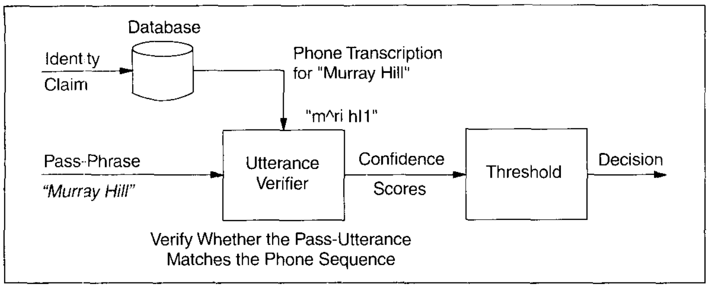

Figure 8. *Verification approach for VIV.*

VIV System

In this section, we briefly present a VIV system based on the utterance-verification approach. Details of the system can be found in [2, 3].

We implemented a VIV system that authenticates a speaker by asking three questions sequentially as shown in Fig 3. The answer to each of the three questions is verified by utterance verification. If all three questions are correctly answered, the speaker is accepted; otherwise, the speaker is rejected and no further questions are asked.

A block diagram of the utterance-verification approach is shown in Fig. 8. Similar to SV, the voice response to a question is first aligned with a sequence of corresponding transcribed phonemes of the correct answer using SI HMMs. Then, for each of the phonemes, the likelihood scores of the corresponding SI HMM and anti-HMMs are compared for hypothesis testing. Furthermore, a confidence measure is formed by combining the hypothesis test scores on each phoneme into an utterance-level score for decision. A confidence measure can be a linear or nonlinear function of the likelihood scores of each phoneme. An example of the confidence measure is

$$M = \frac{1}{K}\sum_{m=1}^{K}\frac{1}{N_m}[\log P(\mathbf{O}_m|\lambda_m) - \log P(\mathbf{O}_m|\bar{\lambda}_m)], \tag{22}$$

where K is the total number of phonemes, $\mathbf{O}_m$ is the segmented feature set for subword m with N_m feature vectors, and λ_m and $\bar{\lambda}_m$ are the target and anti-models for the mth phoneme. The anti-model is trained using the data that are near the target phoneme in the feature space as described in the "Stochastic Models" section [4]. Variations of confidence measures can be found in [2, 7].

The VIV system has been evaluated on a database of 100 speakers. Each speaker in the database was tested both as a true speaker and an imposter. Thus, for each speaker, we have three utterances from the true speaker and 99 × 3 utterances from all the other speakers as impostors. The experimental results are shown in Table 2. When a speaker is verified using three questions, the VIV system achieved 0% average individual equal-error rate, with a speaker-dependent threshold set for each individual information field.

Table 2. Experimental Results on Verbal-Information Verification

Approach	False Rejection on 3 Utterances	False Acceptance on 3 Utterances	Equal-Error Rate
Utterance Verification	0%	0%	0%
Tested on 100 speakers with 3 questions.			

In a VIV system, we assume that the user protects his or her personal information from impostors. In other words, an imposter can break into a VIV system by using the true speaker's information. To improve the security, a VIV system can randomly ask for a subset of personal information for each access. For example, the user registers six items in the profile, and each time the system randomly picks three for verification. Furthermore, the system can ask dynamic information registered in past transactions, such as the date or the amount of the last deposit. As proposed in [3], a speaker authentication system can also start with a VIV system, then adapt gradually to an SV system to further improve the authentication performance. The traditional SV enrollment, which requires the speaker to register multiple voice utterances before he/she can use the system, can thus be avoided.

Conclusions

The common technique used in the above SV and VIV systems is statistical hypothesis testing. By employing the background models [25] (Fig. 6) and/or the anti-models [4], the verification module tests two competing hypotheses, and the system performance is improved significantly over the systems without hypothesis testing.

SV and VIV are the two most practical approaches to speaker authentication. The performances of lab data indicate that both systems are ready for real-world applications. A system incorporating both SV and VIV, being able to provide higher flexibility and security, is even more attractive [3]. Also, different levels of security requirement can be provided by varying the system parameter, complexity, and the enrollment procedure.

Acknowledgment

The authors wish to thank S. Parthasarathy and Aaron E. Rosenberg for many useful discussions, their contributions to the stochastic matching algorithm, and providing the fixed-phrase SV system for experiments.

Keywords

Speaker authentication, speaker recognition, verbal-information verification, speaker verification, speaker identification.

References

[1] Q. Li, S. Parthasarathy, A.E. Rosenberg, and D.W. Tufts, "Normalized discriminant analysis with application to a hybrid speaker-verification system," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Atlanta, GA, May 1996.

[2] Q. Li, B.-H. Juang, Q. Zhou, and C.-H. Lee, "Verbal information verification," in *Proc. EUROSPEECH*, Ghode, Greece, Sept. 22-25 1997, pp. 839-842.

[3] Q. Li and B.-H. Juang, "Speaker verification using verbal information verification for automatic enrollment," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Seattle, WA, May 1998.

[4] R.A. Sukkar and C.-H. Lee, "Vocabulary independent discriminative utterance verification for non-keyword rejection in subword based speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, November 1996, pp. 420-429.

[5] R.A. Sukkar, A.R. Setlur, M.G. Rahim, and C.-H. Lee, "Utterance verification of keyword string using word-based minimum verification error (WB-MVE) training," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Atlanta, GA, May 1996, pp. 518-521.

[6] A.R. Setlur, R.A. Sukkar, and J. Jacob, "Correcting recognition errors via discriminative utterance verification," in *Proc. Int. Conf. Spoken Language Processing*, Philadelphia, PA, October 1996, pp. 602-605.

[7] T. Kawahara, C.-H. Lee, and B.-H. Juang, "Combining key-phrase detection and subword-based verification for flexible speech understanding," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Munich, Germany, May 1997, pp. 1159-1162.

[8] M.G. Rahim, C.-H. Lee, and B.-H. Juang, "Robust utterance verification for connected digits recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Detroit, MI, May 1995, pp. 285-288.

[9] S. Furui, "Cepstral analysis techniques for automatic speaker verification," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 27, pp. 254-277, April 1981.

[10] F. K. Soong, A. E. Rosenberg, and B.-H. Juang, "A vector quantization approach to speaker recognition," *AT&T Technical Journal*, vol. 66, pp. 14-26, March/April 1987.

[11] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Stat.*, vol. 41, pp. 164-171, 1970.

[12] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Society*, vol. 39, no. 1, pp. 1-38, 1977.

[13] S. Katagiri, C.-H. Lee, and B.-H. Juang, "New discriminative algorithm based on the generalized probabilistic descent method," in *Proc. IEEE Workshop on Neural Network for Signal Processing*, Princeton, NJ,September 1991, pp. 299-309.

[14] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Signal Processing*, vol. 40, pp. 3043-3054, December 1992.

[15] W. Chou, B.-H. Juang, and C.-H. Lee, "Segmental GPD training of HMM-based speech recognizer," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, San Francisco, CA, March 1992, pp. 473-476,.

[16] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 5, pp. 257-265, May 1997.

[17] L. R. Bahl, P.F. Brown, P.V. de Souza, and R.L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Tokyo, Japan, 1986, pp. 49-52.

[18] Y. Normandin, R. Cardin, and R.D. Mori, "High-performance connected digit recognition using maximum mutual information estimation," *IEEE Trans. Speech Audio Processing*, vol. 2, pp. 299-311, April 1994.

[19] C. S. Liu, C.-H. Lee, W. Chou, B.-H. Juang, and A.E. Rosenberg, "A study on minimum error discriminative training for speaker recognition," *J. Acoust. Soc. Amer.*, vol. 97, pp. 637-648, January 1995.

[20] F. Korkmazskiy and B.-H. Juang, "Discriminative adaptation for speaker verification," in *Proc. Int. Conf. on Spoken Language Processing*, vol. 3, Philadelphia, PA, 1996, pp. 28-31.

[21] A.E. Rosenberg, O. Siohan, and S. Parthasarathy, "Speaker verification using minimum verification error training," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Seattle, WA, May 1998, pp. 105-108.

[22] O. Siohan, A.E. Rosenberg, and S. Parthasarathy, "Speaker identification using minimum verification error training," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Seattle, WA, May 1998, pp. 109-112.

[23] A.J. Viterbi. "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Informa. Theory*, vol. IT-13, pp. 260-269, April 1967.

[24] G.D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268-278, March 1973.

[25] A.E. Rosenberg and S. Parthasarathy, "Speaker background models for connected digit password speaker verification," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Atlanta, GA, May 1996, pp. 81-84.

[26] K. Fukunaga, *Introduction to Statistical Pattern Recognition, Second Edition*. New York: Academic Press, 1990.

[27] J. Neyman and E.S. Pearson, "On the problem of the most efficient tests of statistical hypotheses," *Phil. Trans. Roy. Soc. A*, vol. 231, pp. 289-337, 1933.

[28] J. Neyman and E.S. Pearson, "On the use and interpretation of certain test criteria for purpose of statistical inference," *Biometrika*, vol. 20A, Pt I, pp. 175-240; 1928.

[29] A. Wald, *Sequential Analysis*. New York: Chapman & Hall, 1947.

[30] Q. Li, S. Parthasarathy, and A.E. Rosenberg, "A fast algorithm for stochastic matching with application to robust speaker verification," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Munich, Germany, pp. 1543-1547, April 1997.

[31] S. Parthasarathy and A.E. Rosenberg, "General phrase speaker verification using sub-word background models and likelihood-ratio scoring," in *Proc. ICSLP-96*, Philadelphia, PA, October 1996.

[32] A.E. Rosenberg, C.-H. Lee, and F. K. Soong, "Cepstral channel normalization techniques for HMM-based speaker verification," in *Proc. Int. Conf. on Spoken Language Processing*, Yokohama, Japan, 1994, pp. 1835-1838.

[33] B.S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *J. Acoust. Soc. Amer.*, vol. 55, pp. 1304-1312, 1974.

[34] B. S. Atal, "Automatic recognition of speakers from their voices," *Proc. IEEE*, vol. 64, pp. 460-475, 1976.

[35] A. Sankar and C.-H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 190-202, May 1996.

[36] M. G. Rahim and B.-H. Juang, "Signal bias removal by maximum likelihood estimation for robust telephone speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 19-30, January 1996.

[37] R.J. Mammone, X. Zhang, and R.P. Pamachandran, "Robust speaker recognition," *IEEE Signal Processing Magazine*, vol. 13, no. 5, pp. 58-71, Sept. 1996.

[38] A.C. Surendran, *Maximum-likelihood stochastic matching approach to non-linear equalization for robust speech recognition*. PhD thesis, Rutgers University, Piscataway, NJ, May 1996.

[39] D. Mansour and B.-H. Juang, "A family of distortion measures based upon projection operation for robust speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1659-1671, November 1989.

Qi Li received the B.S. degree from Hebei University of Science and Technology, Hebei, the M.S. degree from Northeastern University, Boston, and the Ph.D. degree from University of Rhode Island, Kingston, all in electrical engineering. From 1988 to 1994, he worked at F.M. Engineering and Research, Norwood, MA, where he engaged in research on patent-recognition algorithms and in real-time systems. In 1991, he attended Harvard University to study statistical theory and methods. In 1995, he joined Bell Laboratories, Murray Hill, NJ. He is currently a member of the technical staff in the Dialogue Systems Research Department. His research interests include speaker and speech recognition, fast search algorithms, stochastic modeling, robust features, fast discriminative learning, and neural networks. Dr. Li has been active as a reviewer for several journals including *IEEE Transactions on Speech and Audio Processing*, and as a local chair for the Workshop on Automatic Identification.

Dr. Biing-Hwang Juang is the head of Acoustics & Speech Research Department at Bell Labs, Lucent Technologies. His research activities include speech coding, speech recognition, and multimedia communications. He has published extensively and holds a number of patents in the area of speech communication and communication services. He is co-author of the book *Fundamentals of Speech Recognition* published by Prentice-Hall. He was an editor for the *IEEE Transactions on Acoustics, Speech, and Signal Processing* (1986-88), the *IEEE Transactions on Neural Networks* (1992-93), and the *Journal of Speech Communication* (1992-94). He has served on the Digital Signal Processing and the Speech Technical Committees as well as the conference board of the IEEE Signal Processing Society and was (1991-1993) Chairman of the Technical Committee on Neural Networks for Signal Processing. He is currently Editor-in-Chief of the *IEEE Transactions on Speech & Audio Processing* and member of the editorial board of the *IEEE Proceedings*. He also serves on international advisory boards outside the United States. He is a Fellow of the IEEE.

Chin-Hui Lee received a B.S. in electrical engineering from National Taiwan University, Taipei, in 1973; an M.S. in engineering and applied science from Yale University, New Haven; in 1977; and a Ph.D. in electrical engineering with a minor in statistics from the University of Washington, Seattle, in 1981. Since 1986, he has been with Bell Laboratories, Murray Hill, New Jersey, where he is now a Distinguished Member of the technical staff and head of the newly established Dialogue Systems Research Department. His current research include multimedia signal processing, speech and speaker recognition, speech and language modeling, adaptive and discriminative learning, spoken dialogue processing, biometric authentication, and information retrieval. His research scope is reflected in the recently edited book *Automatic Speech and Speaker Recognition: Advanced Topics*, published by Kluwer Academic Publishers in 1996. He is a member of the IEEE Signal Processing Society, the IEEE Communication Society, and the European Speech Communication Association. He is also a lifetime member of the Computational Linguistic Society in Taiwan. From 1991 to 1995, he was an associate editor for the *IEEE Transactions on Signal Processing and Transactions on Speech and Audio Processing* and a member of the ARPA Spoken Language Coordination Committee. He is currently the chairman of the SPS Speech Processing Technical Committee.

Qiru Zhou received his B.S. and M.S. from Northern Jiao-Tong University and Beijing University of Posts and Telecommunications, respectively, in electrical and computer engineering. He joined Bell Labs, AT&T in 1992. Currently he is a member of the technical staff at Bell Labs, Lucent Technologies, Murray Hill, New Jersey, with the Dialogue Systems Research Department. His research interests include speech- and speaker-recognition algorithms and software, speech and multi-modal dialogue system architecture, and real-time and distributed object-oriented software architecture for multimedia communications. Since 1992, he has been involved in various projects at AT&T and Lucent concerning the application of speech technologies into products. He is a technical leader in Lucent's speech software product development.

Frank K. Soong received his B.S., M.S., and Ph.D. from the National Taiwan University, the University of Rhode Island, and Stanford University, respectively, all in electrical engineering. In 1982, he joined Bell Labs as a member of the technical staff of the Acoustics Research Department. Currently, he is a Distinguished Member of the technical staff at Bell Labs, Lucent Technologies, Murray Hill, New Jersey, with the Dialogue Systems Research Department. His research interests include speech and speaker recognition; speech analysis and coding, particularly in optimal quantization; stochastic modeling; and fast search algorithms. The most current research that he is involved with is hands-free voice user interfaces, including directional microphone, noise suppression, room dereverberation, echo cancellation, speaker adaptation, and robust speech recognition. He was an invited researcher at the Electrical Communication Lab (ECL) of NTT, Musashino, Japan, from 1987 to 1988. He co-chaired the third International IEEE Speech Recognition Workshop in 1991. He has served as an associate editor of *IEEE Transactions on Speech and Audio Processing*.

Address for Correspondence: Qi (Peter) Li, Bell Labs, Lucent Technologies, Room 2C-572, 600 Mountain Avenue, Murray Hill, NJ 07974, Tel: (908) 582-6443, Fax: (908) 582-7308, E-mail: qli@bell-labs.com.