Classification Methods for Speaker Recognition[*]

D.E. Sturim, W.M. Campbell, and D.A. Reynolds

Massachusetts Institute of Technology
Lincoln Laboratory
244 Wood Street
Lexington, MA 02420, USA
{sturim, wcampbell, dar}@ll.mit.edu

Abstract. Automatic speaker recognition systems have a foundation built on ideas and techniques from the areas of speech science for speaker characterization, pattern recognition and engineering. In this chapter we provide an overview of the features, models, and classifiers derived from these areas that are the basis for modern automatic speaker recognition systems. We describe the components of state-of-the-art automatic speaker recognition systems, discuss application considerations and provide a brief survey of accuracy for different tasks.

1 Introduction

The development of automatic speaker recognition systems is one example in the field of speech processing that brings together the areas of speech science for speaker characteristization, pattern recognition and engineering. From speech science comes the insights into how humans produce and perceive speaker-dependent information in the speech signal as well as signal processing techniques for analyzing acoustic correlates conveying this information. The area of pattern recognition provides algorithms for effectively modeling and comparing speaker characteristics from salient features. Finally, engineering is used to both realize working systems based on the above ideas and to handle real-world variability that arise in applications. In this chapter we provide an overview of the features, models, and classifiers derived from these areas that are the basis for modern automatic speaker recognition systems.

In Figure 1, we show the basic framework and components of speaker recognition systems. We are using the general term of speaker recognition to encompass the underlying tasks of speaker identification (which one of a set of speakers is talking?) and speaker detection or verification (is this particular speaker talking?). We will note throughout this chapter when particular comments refer

[*] This work was sponsored by the Department of Justice under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

C. Müller (Ed.): Speaker Classification I, LNAI 4343, pp. 278–297, 2007.

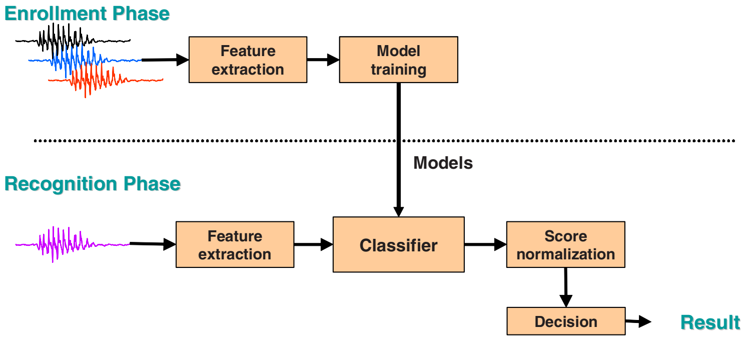

Fig. 1. Structure of a speaker recognition system

to identification or detection. As with any pattern recognition system, speaker recognition systems consist of two distinct phases: enrollment (also called training) and recognition (also called testing).

The first step, common to both enrollment and recognition phases, is the extraction and conditioning of a set of features from the input signal believed to convey information about the speaker. In Section 2, we review some of the commonly used methods for feature extraction.

Features from speech samples by a speaker are used in the enrollment phase to build or train parameters for a model which represents the specific characteristics of that speaker. During the recognition phase, features from the test speech sample are compared to one or more of the speaker models, depending on the task, by the classifier to produce match scores. In Section 3, we review the most successful models and classifiers found in automatic speaker recognition systems.

These scores are optionally normalized to add robustness or to map them to a desired dynamic range (e.g., 0 to 1). This and other forms of normalization and compensation are discussed in Section 3.6.

Finally, the decision component either compares the score to a threshold to decide to accept or reject, in the case of speaker detection, or reports out the highest scoring model, in the case of speaker identification. The decision could also compare the score of the highest scoring model to a threshold and decide to report “none-of-the-above.” This is a merger of speaker identification and detection known as *open-set* identification.

2 Feature Extraction

Feature processing for speaker recognition systems consists of extracting speaker dependent information in a form which can be effectively and efficiently used for model building and recognition. Broadly speaking, features used for speaker recognition can be categorized by three key attributes:

- Temporal span
- Discrete vs. continuous values
- Information level

The attributes of features will impact the models and classifiers that are appropriate to use.

The information in speech signals occurs at several different time spans and rates. Thus, features used to capture this information also occur with different time spans and rates. Features that aim to capture information about a person's vocal tract information as seen through the frequency spectrum of speech, will operate using short-time spans (∼20-30ms) so as to analyze quasi-stationary snapshots of the vocal apparatus. Prosodic information, such as a person's average pitch inflection per sentence, is an example of a feature derived by looking at a longer time span (∼1-2 s). Further, the feature time span and rate may be variable, for example, when examining aperiodic, variable duration events like speech pathologies, phonemes, or words.

The value of the speech measurements used in the features can be discrete or continuous. Features consisting of speech frequency spectrum samples are an example of continuous valued measurements. Features counting the number of occurrences of events in speech, such as word usage counts, are an example of discrete values measurements. There is, of course, a continuum between continuous and discrete measurements since one can quantize continuous values for efficiency or use a probability of occurrence that is < 1.0 when counting events.

The third attribute is the information level features represent. Speech conveys many levels of information, from semantic meaning, via the words spoken, to the speaker's physical vocal apparatus, via the acoustic sound of the speech (i.e., bass vs treble). Speaker recognition features can be focused to capture speaker dependent characteristics from these different levels. Features aimed at low-level information tend to extract measurements about the acoustic characteristics related to vocal production, such as frequency spectrum or short time pitch estimates. Features aimed at higher-level information, such as pronunciations and word usage (idiolect), require the output of some other speech recognition tool such as a phone or word recognition system.

We pictorially depict this feature attribute space in Figure 2. Typically, features related to high-level speaker information consist of longer time span, variable rate analysis of discrete events, such as phones or words. Features related to low-level speaker information consist of short time span, fixed rate analysis of continuous phenomenon, such as spectra. We next review some common features used in automatic speaker recognition systems indicating their attributes. Figure 3 shows where these features lie in the attribute space.

Mel Frequency Cepstral Coefficients (MFCCs) [1,2]: MFCCs are the most commonly used features in modern speaker recognition systems[3]. MFCC temporal processing uses a fixed analysis window on the order of ∼20 millisecond. MFCCs are represented by a real valued N-dimensional vector. The coefficients are a parameterization of the spectrum which have some dependency on the

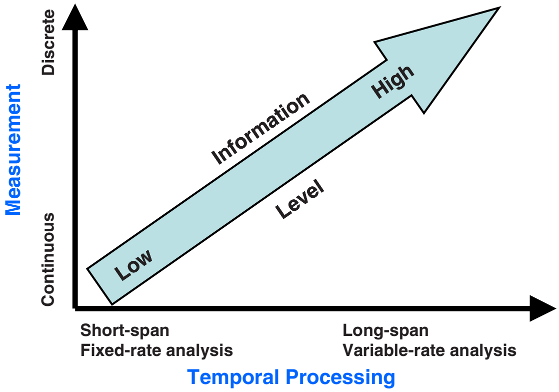

Fig. 2. Relation of attributes for features used in automatic speaker recognition systems

physical characteristics of the speaker. MFCCs are considered to be low-level information.

Linear Prediction-based Cepstral Coefficients (LPCCs) [4,2]: LPCCs are often used in speaker recognition systems, although their susceptibility to noisy environments have made them more undesirable as speaker recognition systems are applied to more challenging channels. Like MFCCs, the LPCC processing uses a fixed analysis window (∼20 millisecond) and are of the continuous measurement type. LPCCs are dependent on the spectral envelope and are considered to be low-level information.

Codebook quantized spectral entries [5]: These features measure the approximate location of the spectrum in acoustic space. Rather than use the continuous representation of cepstral features, the features can be quantized either using a VQ codebook or a Gaussian mixture model (GMM). The feature in this case is the index in the corresponding VQ codebook or the mixture index in the GMM.

Pitch and Energy [6]: The goal is to learn pitch and energy gestures by modeling the joint slope dynamics of pitch and energy. When these features are combined with a short phrases, the analysis window will be variable spanning the duration of the short phrase.

Prosodic Statistics [7]: Are based on various measurements of energy, duration and pitch derived over large speech segment. The goal is to capture the prosodic idiosyncrasies of individual speakers. The feature type will be continuous since the prosodic statistical measures are reported in continuous values. The level of information is considered low-middle since these features are measuring prosodic inflections and patterns.

Word and Phone Tokenization [8,9,10,11,12]: These are a more recent addition to feature sets used in speaker recognition systems. The analysis window is variable, since it is based on the expected duration of the word or phone units.

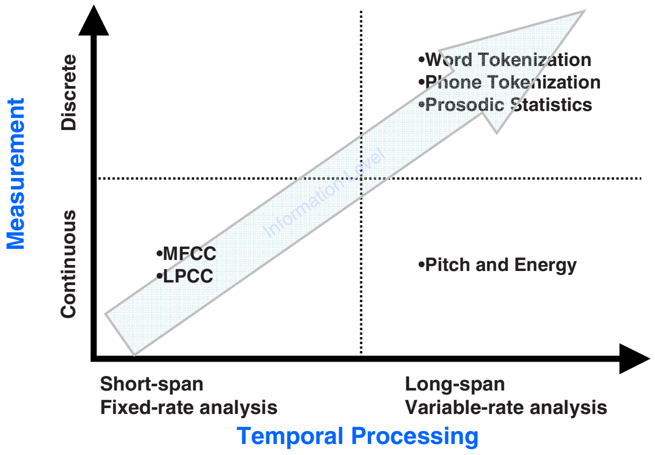

Fig. 3. Approximate location of common feature in the feature attribute space

Further counts of word pairs or triples cover longer time spans. Since counts of discrete words and phone are often used as features, the value type would be discrete. Word and phone models in speaker recognition both try to represent the pronunciation differences of talkers and are considered high-level information.

3 Models and Classifiers

Speaker models and classifiers are tied not only to the features used, but also to the task being addressed. The two tasks of speaker recognition are 1) speaker identification and 2) speaker verification. The speaker identification task is closed-set recognition, where all of the talkers that will be seen by the system are pre-enrolled and known. Figure 4 shows the general structure of a speaker identification system. The applications of closed-set identification are limited since most real-world scenarios must usually handle out-of-set speakers. Performance is a function of the number of speaker in the identification set and the speech used.

The speaker verification task, in contrast, is a binary decision of whether the unknown speaker is the same as the hypothesized (or claimed) speaker. While ostensibly an easier task than classifying among a set on N speakers, verification must potentially be able to effectively reject the open-set of speakers that could act as impostors. This open-set is usually dealt with by using some general impostor model. The general structure of the speaker verification system is presented in Figure 5. Speakers verification addresses a more general problem and has wider application in the speaker recognition community, so it is a more common focus for classifier design and evaluation.

For both the identification and verification structure, there are many types of models and classifiers that have been used. We will mainly focus on those aimed at solving the more general open-set verification task (although they are

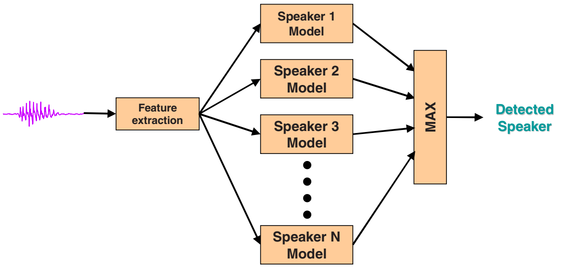

Fig. 4. General classifier structure for speaker identification system

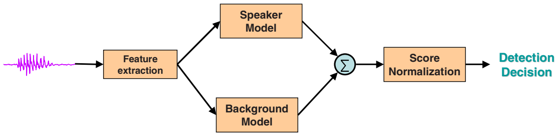

Fig. 5. General classifier structure for speaker verification system

often similarly used for identification). Early methods for speaker recognition included non-parametric techniques (vector quantization and dynamic time warping). Classification methods for speaker recognition in recent years have centered on statistical approaches. The structure and choice of a classifier depends on the application and the features used. In this section we review a subset of classifiers that have been successfully used in automatic speaker recognition systems.

3.1 Gaussian Mixture Modeling (GMM)

The Gaussian mixture modeling (GMM) approach has become one of the mainstay modeling techniques in *text-independent* speaker recognition systems. Consider the verification structure shown in Figure 5. In GMM speaker verification, the impostor model is more commonly known as a background model. In addition, the detection decision or score is normalizated to refine detection decision. The resulting structure is presented in Figure 5.

Figure 5 is realized in the framework of a likelihood ratio detector. In the approach of [3,13,14], we can consider the two hypotheses for a given segment of speech Y:

λ_{hyp}: Speech segment Y is from speaker S
$\lambda_{\overline{hyp}}$: Speech segment Y is not from speaker S

To decide between these two hypotheses we form the following likelihood ratio test:

$$\Lambda(Y) = \frac{p\left(Y|\lambda_{hyp}\right)}{p\left(Y|\lambda_{\overline{hyp}}\right)} \begin{cases} \geq \Theta & \text{Accept hypothesis } \lambda_{hyp} \\ \leq \Theta & \text{Reject hypothesis } \lambda_{hyp} \end{cases} \tag{1}$$

where $p\left(Y|\lambda\right)$ is the probability density function (pdf) of the observed speech segment Y, given the hypothesis λ, or likelihood function. The decision threshold, Θ, determines accepting or rejecting the hypotheses. Let X represent the set of feature vectors generated from the front-end processing of the speech segment Y. The set of features, X, usually MFCCs or LPCCs, are per frame speech-frame vectors: $\{\boldsymbol{x_1}, \cdots, \boldsymbol{x_T}\}$. The frame-based likelihood function can be written as $p\left(\boldsymbol{x}|\lambda\right)$.

In the GMM approach, the choice of the likelihood function is a mixture of M Gaussians:

$$p(\boldsymbol{x}|\lambda) = \sum_{i=i}^{M} w_i p_i(\boldsymbol{x}) \tag{2}$$

where $p_i(\boldsymbol{x})$ is the individual Gaussian density function,

$$p_i(\boldsymbol{x}) = \frac{1}{\left(2\pi\right)^{n/2}\left|\Sigma_i\right|^{1/2}} \times \exp\left\{-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}_i\right)' \Sigma_i^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}_i\right)\right\}. \tag{3}$$

The parameters of the model are: w_i, the mixture weight, $\boldsymbol{\mu}_i$, the N-dimensional mean vector, and Σ_i, the N by N dimensional covariance matrix. The model parameters can be succinctly written as: $\lambda = (w_i, \mu_i, \Sigma_i)$ where $i = [1 \cdots M]$. Equation (2) is just a linearly weighted sum of M individual Gaussians which will be used the likelihood calculation for a detection decision. The weights also satisfy the relation $\Sigma_{i=1}^{M} w_i = 1$. The general form of a Gaussian mixture allows for a fully populated covariance matrix. It has been shown that the diagonal covariance matrix is sufficient for text-independent speaker-verification modeling [3].

Once a model is trained then (2) can be used to evaluate the log-likelihood of model λ for an input test set of feature vectors, X :

$$\log p\left(X|\lambda\right) = \sum_{t=1}^{T} \log p\left(\boldsymbol{x_i}|\lambda\right) \tag{4}$$

Impostor modeling is crucial in producing good speaker recognition performance. Current methods form an universal background model, $p\left(\boldsymbol{x}|\lambda_{\overline{hyp}}\right)$, from a set of background model speakers [15]. The background speakers are chosen from a similarly recorded channel/conditions that will be seen in detection. The number of speakers used to train the background model should be large enough to model the acoustic space of the impostors. There is also a dependency on the number of Gaussians (M) used to model the space. A larger number of Gaussians will require more data to realize the mixture model. The size of M, will depend on channel, application, acoustic variation and amount of speech data seen at each phase. M may range from 64 to 2048. In the telephone speaker-verification task, with 2.5 minutes of enrollment speech and 30 second of verification speech, we have seen good performance with the number of mixtures $M = 512$ and 1-2 hours of background model training speech from over one hundred talkers.

Current state-of-the-art text-independent GMM speaker verification systems obtain background model parameter estimates in an unsupervised manner by using an expectation-maximization (EM) algorithm [16]. Feature vectors generated from a background speaker set provide the training data. The EM algorithm iteratively refines model parameter estimates to maximize the likelihood that the model matches the distribution of the training data. Model parameters converge to a final solution in a few iterations (5-10)[3].

Speaker model training is accomplished by adapting the background model to each enrollment speaker through *Maximum A Posteriori* (MAP) estimation [17,18]. This approach couples the speaker model to the background model and yields better results over the methods using unrelated models. Adapting from the background model utilizes the well trained parameters, $\{w_i, \boldsymbol{\mu}_i, \Sigma_i\}$, from the EM algorithm. The large amount of data used to train the background model allows for a well modeled cepstral space. Speaker models are adapted in turn from this richly populated space. Even though all the parameters of the model can be adapted, it has been shown that best performance results when only the means ($\boldsymbol{\mu}_i$) are adapted.

The speaker and background models can be applied to the likelihood ratio (1) and (4) to get the likelihood-ratio score,

$$\Lambda(X) = \log p\left(X|\lambda_{hyp}\right) - \log p\left(X|\lambda_{\overline{hyp}}\right) \tag{5}$$

Equation 5 is sufficient to form a detection decision, however better performance is achieved through refinement of the likelihood-ratio score with normalization. We will discuss normalization techniques in Section 3.6.

It should be noted the similarities in the organization of the GMM and the vector quantization (VQ) approach for speaker recognition. In the method of [19,20], the VQ codebook is a partitioning of the cepstral space. The VQ codebook can be weakly considered a quantized version of a Gaussian mixture model.

A support vector machine (SVM) is a versatile classifier that has gained considerable popularity in recent years. An SVM is discriminative and models the boundary between a speaker and a set of impostors. The typical method employed in SVM speaker recognition is based upon comparing speech utterances using sequence kernels. Rather than characterize features from individual frames of speech, these methods model entire sequences of feature vectors. Approaches include the generalized linear discriminant sequence kernel [21], Fisher kernel methods [22,23], n-gram kernels [24], MLLR transform kernels [25], and GMM supervector kernels [26].

Basic SVM Theory. An SVM [27] models two classes using sums of a kernel function $K(\cdot,\cdot)$,

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i t_i K(\mathbf{x}, \mathbf{x}_i) + d, \tag{6}$$

where the t_i are the ideal outputs, $\sum_{i=1}^{N} \alpha_i t_i = 0$, and $\alpha_i > 0$. The vectors $\mathbf{x}_i$ are support vectors and obtained from the training set by an optimization

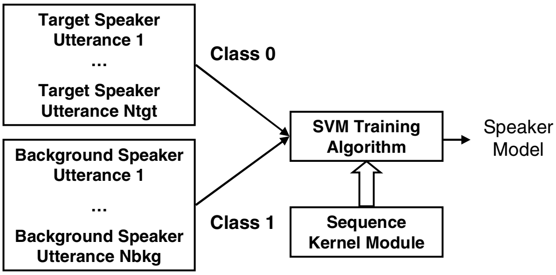
Target Speaker Utterance 1
...
Target Speaker Utterance Ntgt
Background Speaker Utterance 1
...
Background Speaker Utterance Nbkg
Class 0
Class 1
SVM Training Algorithm
Sequence Kernel Module
Speaker Model

Fig. 6. Setup for training an SVM classifier for speaker verification

process [28]. The ideal outputs are either 1 or -1, depending upon whether the corresponding support vector is in class 0 or class 1, respectively. For verification, a class decision is based upon whether the value, $f(\mathbf{x})$, is above or below a threshold.

The kernel $K(\cdot,\cdot)$ is typically constrained to have the Mercer condition, so that $K(\cdot,\cdot)$ can be expressed as

$$K(\mathbf{x},\mathbf{y}) = \mathbf{b}(\mathbf{x})^t\mathbf{b}(\mathbf{y}), \tag{7}$$

where $\mathbf{b}(\mathbf{x})$ is a mapping from the input space (where $\mathbf{x}$ lives) to a possibly infinite-dimensional *expansion space*. Optimization of an SVM relies upon a maximum margin concept. For separable data, the system places a hyperplane in a high dimensional space so that the hyperplane has maximum margin. The data points from the training set lying on the boundaries are the support vectors in equation (6).

Application of Support Vector Machines to Speaker Recognition. Figure 6 indicates the basic training strategy for SVMs using sequence kernels. We train a target model with target speaker utterances and a set of example speakers' utterances that have characteristics of the impostor population—a background speaker set. Each utterance from a target or background speaker becomes a point in the SVM expansion space. We implement a sequence kernel module for comparing two utterances and producing a kernel value. The kernel module is connected into a standard SVM training tool which then produces a speaker model. We keep the background speaker set the same as we enroll different target speakers.

Sequence Kernels for Speaker Recognition—General Structure. To apply an SVM, $f(\mathbf{X})$, in a speaker recognition application, we need a method for calculating kernel values from sequences of features (e.g., MFCC feature vectors). Two general methods have emerged—linearized train/test kernels and adapted model comparison.

The idea of a train/test sequence kernel is shown in Figure 7. The basic approach is to compare two speech utterances, *utt 1* and *utt 2* by training a model on one utterance and then scoring the resulting model on another utterance.

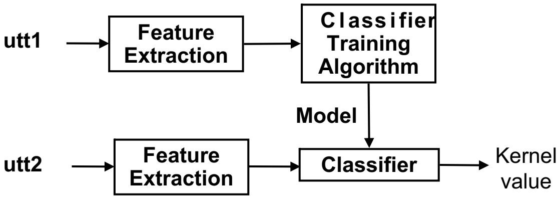

Fig. 7. Constructing a sequence kernel using a train/test strategy

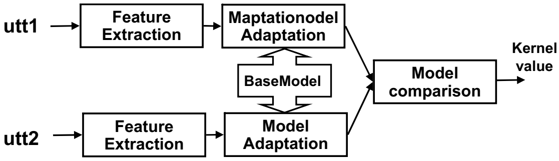

Fig. 8. Constructing a kernel using a base generative model

This process produces a value that measures the similarity between the two utterances. Although in general this comparison is not a kernel, it doesn't satisfy the Mercer condition, in many cases linearization will produce a kernel—see the next section.

The second basic method for constructing sequence kernels is shown in Figure 8. In this setup, we adapt a base model to obtain probability distributions which represent the utterances. We then apply a model comparison algorithm to get a measure of similarity. This approach has the useful property that it is naturally symmetric as long as the comparison calculation is symmetric.

3.2 Sequence Kernels for Speaker Recognition—Specific Examples

For the train/test kernel shown in Figure 7, a typical approach is the generalized linear discriminant sequence (GLDS) kernel [21]. In this method, the classifier is taken to be a polynomial discriminant function. Suppose we have two sequences of feature vectors, $\mathbf{X} = \{\mathbf{x}_i\}$ and $\mathbf{Y} = \{\mathbf{y}_j\}$. If we train a polynomial discriminant using mean-squared error, then the resulting kernel is given by

$$K(\mathbf{X}, \mathbf{Y}) = \bar{\mathbf{b}}_x \bar{\mathbf{R}}^{-1} \bar{\mathbf{b}}_y. \tag{8}$$

In (8),

$$\bar{\mathbf{b}}_x = \frac{1}{N_x} \sum_i \mathbf{b}(\mathbf{x}_i); \tag{9}$$

i.e., $\bar{\mathbf{b}}_x$ is the average expansion over all frames. A similar expansion is used for $\mathbf{Y}$. The matrix, $\bar{\mathbf{R}}$ is the correlation matrix of a background data set; typically, it is approximated with only diagonal terms. For details on the derivation of these equations, we refer to [21]. An interesting generalization of the GLDS kernel is to replace the polynomial expansion by a general kernel using the kernel trick [29,30].

For the generative model sequence kernel, several methods have been proposed. Current methods are based upon adapting from a GMM or HMM base model. In [25], adaptation of an HMM from a speech-to-text system is performed using maximum-likelihood linear regression (MLLR). The MLLR adaptation parameters are then compared with a weighted linear inner product. In [26], the adaptation is performed via MAP adaptation of a GMM. The GMMs are compared using either an approximation to the KL divergence or an integral inner product.

SVMs can also be applied to high-level features [11,24]. A token-sequence comparison kernel can be derived by using the train/test kernel framework in Figure 7. In this case, the classifier in the figure is taken to be the standard language model likelihood ratio using n-gram probabilities. The resulting kernel is of the form

$$K(T_1,T_2) = \sum_k D_k^2 p(d_k|T_1)p(d_k|T_2) \tag{10}$$

where the T_j are token sequences, D_k is a weighting function, $p(d_k|T_j)$ is the probability of a particular n-gram, d_k, occuring in token sequence T_j. A typical choice is something of the form

$$D_k = \min\left(C_k, g_k\left(\frac{1}{p(d_k|\text{background})}\right)\right) \tag{11}$$

where $g_k(\cdot)$ is a function which squashes the dynamic range, and C_k is a constant [24]. The probability $p(d_k|\text{background})$ in (11) is calculated from a large population of speakers. Typical choices for g_k are $g_k(x) = \sqrt{x}$ and $g_k(x) = \log(x) + 1$. The kernel (10) is closely related to methods in information retrieval; we refer to [24] for details.

3.3 Support Vector Machine (SVM)

3.4 Hidden Markov Modeling (HMM)

The GMM-UBM system described in Section 3.1 models the entire acoustic space. However, in text-dependent applications the system has prior knowledge of what will be said and template-matching techniques become advantageous. The first template matching methods were dynamic time warping (DTW) algorithms [31]. However DTW methods proved to be inefficient and methods gave way to a stochastic modeling of each talker's speech where the underlying stochastic processes is not observable of hidden (Hidden Markov Model). Early approaches in applying Hidden Markov Models (HMMs) to text-dependent and text-independent speaker recognition were developed by [15,32,10] and have been continued [33,34,35].

HMMs can efficiently model statistical variations in spectral features. Rather then modeling the entire acoustic space the HMM only models a progression of limited regions of acoustic space. These limited acoustic regions can be defined as states of finite time. These states can be described with a PDF, $p(\boldsymbol{x}_t|s)$ is the probability of per-frame feature vector, $\boldsymbol{x_t}$, given you are in state, s. Transitioning between states, (e.g.: from $t-1$ to t) is defined with a state transition probability, $p(s_t|s_{t-1})$.

The likelihood of T frames of speech occurred given a hypothesis, λ, is:

$$p(X|\lambda) = \sum_{\substack{\text{all} \\ \text{states}}} \prod_{t=1}^{T} p(s_t|s_{t-1})p(x_t|s_t) \tag{12}$$

Which is the Baum-Welch decoding [36,37,38]. Equation (12) can be employed in a similar manner as (5). The likelihood ratio can be constructed from a target likelihood $p(X|\lambda_{hyp})$over the an impostor/background likelihood $p(X|\lambda_{\overline{hyp}})$ as in (1).

The first step in HMM modeling is to form a representation of the impostors. Here the concept of the background model is to form a model of the world of all possible speakers. HMM background models can then be trained through the use of a full large vocabulary continuous speech recognition (LVCSR) system as in [35,39]. There are also approaches that use segmental K-means clustering procedure [33] or limited vocabulary phoneme-based methods were implemented in [40].

The speaker model, $p(X|\lambda_{hyp})$, can be formed by Baum-Welch adaptation from the background model [35]. [33] relies on segmental K-means clustering for training of the target model, but utilizes the speaker independent background model for the segmentation. This can be considered a general form of the GMM approach presented in Section 3.1. The GMM can be thought of as a single state hidden Markov model.

The HMM implementation of [35,39] can either be applied in text-independent or text-dependent applications. For text-independent applications, the language model of the LVCSR system has to be broad enough to span the speech that may be seen by the system.

The actual structure of a text-dependent system will depend greatly on the application. Speaker recognition accuracy is dependent on the performance of the system, but can also be controlled by limiting the vocabulary of the domain. Limiting the talkers to alpha-digits is a common domain. System accuracy may also be influenced by gathering more speech from cooperative speaker by prompting them with a series of random phrases.

3.5 Artificial Neural Networks

Artificial neural networks (ANNs) model continuous features using nonlinear modeling inspired by biological neural networks. A typical artificial neural network is a two-layer perceptron, $m(\mathbf{x})$, of the form

$$m(\mathbf{x}) = \tilde{g}\left(\mathbf{w}^t g(\mathbf{A}\mathbf{x} + c) + d)\right) \tag{13}$$

where $\mathbf{x}$ is the input, $g(\cdot)$ and $\tilde{g}$ are squashing functions, $\mathbf{A}$ is a matrix, $\mathbf{w}$ is a vector, and b and c are bias terms. Artificial neural networks were one of the first methods to be successfully used in discriminative speaker recognition [41].

ANNs, when trained with mean-squared or cross-entropy criteria [42], model the posterior probability, $p(\text{spk}|\mathbf{x}_i)$. Here, $\mathbf{x}_i$ is typically a continuous feature vector such as MFCCs. A typical scoring criterion is to take the average weighted posterior (or log posterior) across all frames of an input utterance.

Because an ANN models a posterior rather than a likelihood, typically cohort normalization or background normalization is not needed to achieve good perfromance. This property is expected since the ANN is a discriminative technique. But, as with most speaker recognition methods, techniques such as TNorm can stabilize thresholds.

Training for an ANN is accomplished in a similar manner to the SVM setup shown in Figure 6 except it is performed with frame level features. Feature vectors for the target speaker are extracted and placed in one class (with ideal output 1). Feature vectors for a background speaker set are placed in another class (with ideal output 0). Then, training with a backpropagation algorithm algorithm is performed.

Note that prior balancing is a critical part of ANN training. Because the target speaker training set size is typically significantly smaller than the background training set, the prior of the target is usually small. Since the output of the ANN approximates a posterior, the target prior is a factor in the ANN output. Compensation for this prior can be performed in training via, e.g. random sampling with prior equalization, or in testing by scaling the output by the target prior.

A successful extension of ANNs is the neural tree network [41] (NTN). NTNs are a combination of tree methods (such as CART) and neural networks. At each node in the tree, a neural network is used to determine which branch is taken. Scoring and training are an extension of standard ANN and tree methods. NTNs were successfully used for many years in a commercial system for text dependent speaker recognition.

Other connectionist methods for speaker recognition include radial basis functions (RBF) and elliptical basis functions (EBF), e.g. [43]. These approaches were only moderately successful and are subsumed by the more general training and modeling approach of GMMs.

3.6 Normalization Techniques

Ideally, score variability should only depend on speaker differences. Other factors may contribute to score variability such as transmission channel, environmental background effects, linguistic variation and session variation. There are many methods to stabilize score variation to make the threshold setting, Θ, more robust. Compensation methods have been developed in the feature domain, model domain, and score domain.

Feature Domain Normalization. Feature domain normalization transforms a base set of features, such as MFCCs, to a new set of features that are more robust to channel and noise effects. Typically, these methods have been based on signal processing and data-driven techniques.

Common feature transformations used to remove channel effects are RASTA [44] and cepstral mean subtraction (CMS) [45]. These methods rely on homomorphic signal processing techniques—filtering a signal in the time domain induces an additive bias in the cepstral domain.

Feature transformations that compensate for noise or other nonlinear distortions include cepstral variance normalization (CVN) and feature warping. CVN, in part, is based upon the fact that additive noise reduces the variance of cepstral coefficients [46]; compensation is realized by renormalizing the cepstral coefficients to unit variance. Feature warping [47] further extends this technique by remapping features to fit some predefined distribution.

More recent feature compensation methods have used supervised data-driven methods. For example, feature mapping [48], uses knowledge of channel types to remap features to a channel neutral model.

Model Domain Normalization–GMM. For GMM based classifiers, techniques that treat the undesired variability as a bias to the mean vectors have been successful. If we stack the means from a GMM into a *supervector* this can be written as

$$\boldsymbol{m}_j(s) = \boldsymbol{m}(s) + \boldsymbol{c}(s) \tag{14}$$

where $\boldsymbol{m}_j(s)$ is the supervector from speaker s's j-th enrollment session, $\boldsymbol{m}(s)$ is the desired compensated supervector for speaker s and $\boldsymbol{c}(s)$ is the undesired variability supervector.

The main difference in the compensation techniques is in how they estimate and remove the variability vector $\boldsymbol{c}(s)$. In Speaker Model Synthesis (SMS) [49], the difference between bias vectors from a set of pre-defined channel types is used to synthetically generate a library of channel-dependent speaker models so as to allow matched-channel likelihood ratio scoring during recognition. More recent latent factor analysis (LFA) based techniques [50,51], model the supervector bias as a low-dimensional normally distributed bias,

$$\boldsymbol{c}(s) = U\boldsymbol{n}(s) \tag{15}$$

where U is the low-rank session loading matrix. The LFA techniques are aimed specifically at compensation of session variability and do not require prior channel detectors or parameters.

Model Domain Normalization–SVM. As with the GMM, compensations with SVM classifiers can also be applied directly in the model domain. The SVM nuisance attribute projection (NAP) method [52] works by removing subspaces that cause variability in the kernel. NAP constructs a new kernel,

$$\begin{aligned} K(\{\mathbf{x}_i\},\{\mathbf{y}_j\}) &= \left[\mathbf{P}\bar{\mathbf{b}}_x\right]^t \left[\mathbf{P}\bar{\mathbf{b}}_y\right] \\ &= \bar{\mathbf{b}}_x^t \mathbf{P}\bar{\mathbf{b}}_y \\ &= \bar{\mathbf{b}}_x^t(\mathbf{I} - \mathbf{v}\mathbf{v}^t)\bar{\mathbf{b}}_y \end{aligned} \tag{16}$$

where $\mathbf{P}$ is a projection ($\mathbf{P}^2 = \mathbf{P}$), $\mathbf{v}$ is the direction being removed from the SVM expansion space, $\mathbf{b}(\cdot)$ is the SVM expansion, and $\|\mathbf{v}\|_2 = 1$. NAP can be applied to both low-level and high-level features.

Score Normalization. Typically, score normalization techniques remap target speaker scores based on some reference set of models, utterances, or channels. One of the most effective score normalization techniques, TNorm (test-normalization) was introduced in [53]. TNorm transforms a target model score, s, to

$$\frac{s - \mu}{\sigma} \tag{17}$$

where μ and σ are the mean and standard deviation of scores from a set of reference speakers' models scored on the input utterance. Other score normalization techniques include Z-Norm [54] (based on normalizing to a reference set of utterances) and H-Norm (based on normalizing to a reference set of channels) [55].

4 Classifier Choice

The choice of classifier to be used is greatly dependent on the application. Examples of application constraints that influence the classifier choice and configuration include the following.

- *Level of user cooperation*
- *Required recognition/detection accuracy*
- *Expected channels*
- *Amount of speech available for enrollment and detection*
- *Available compute and memory resources*
- *How the output is used*

User cooperation will determine whether or not you can field an active or passive system. If the user is cooperative the system can actually prompt the user for additional input speech. The additional input speech will boost performance while at the same time verify that the incoming user is "live". However if the users are uncooperative the system has take to more of a passive role. In these applications the systems have no control over the data they process.

High *recognition/detection accuracy* may be a requirement in areas such as banking account access. Here, it is desirable to be very accurate in who gets access to a user's account. A text-dependent system is applicable in this case since it offers higher performance then text-independent techniques.

The *channel* consists of, the type of microphone used to record the speech, the way the speech is encoded/transmitted, as well has ambient noises. If the application has to deal with a wide variety of channel conditions the classifier could employ some form of channel compensation to boost performance.

The *amount of speech data available for enrollment and detection* will also help determine the classifier. If more data is available then classifiers that key off of high level information become feasible.

Applications may also be limited in *computation and memory resources*. Embedded devices have limited amounts of processing power and available memory. A cell phone will have very limited capabilities that will uniquely constrain the speaker recognizer.

The consumer of the *output of the system* will determine what information is presented to the end user. Certain forensic applications require that systems return word usage and phonotactic information. In this application a word or phone based recognition systems, as described in Section 3.3, may be required to generate the information needed by the user. Further the type of output may need to be a hard decision, a human interpretable score, or a relative score to used by another automatic process.

It is quite difficult to characterize the accuracy of speaker verification systems in all applications due to the complexities and differences in the enrollment/detection scenarios. Figure 9 attempts to provide a range of performance for some of the cases mentioned above. These numbers are not meant to indicate the best performance that can be obtained, but rather a relative ranking of some different scenarios. In Figure 9, we depict a detection error trade-off (DET) plot, which shows the trade-off between false-rejects, f_r, and false-accepts, f_a, as the decision threshold changes in a verification system. On this DET we show four equal error rate points (EER is a summary performance indicator where $f_r = f_a$) for four different verification application scenarios. One thing to note is that system performance improves as more constraints are placed on the application conditions (e.g., text-dependent vs. test-independent, increased speech for enrollment and verification, more benign channels).

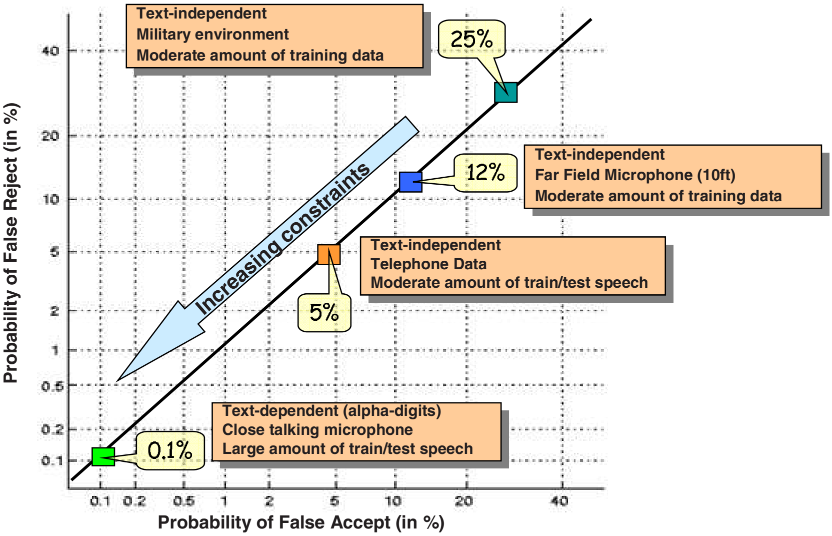

Fig. 9. Range of speaker verification performance

To examine some differences in classifiers, Figure 10 shows EER performance for a few of the text-independent systems described in section 3 for two conditions of enrollment data [56,57]. In the first condition about 2.5 minutes of speech is available for both enrollment and detection. In the other condition about 20 minutes of speech is available for enrollment and 2.5 minutes is available for detection. As expected, the trend is for performance to get better when more enrollment data is available. Further we see that spectral systems (GMM-LFA and SVM-GSV) perform better than high-level feature systems (SVM Word), but fusion of high and low level systems can produce some performance gains.

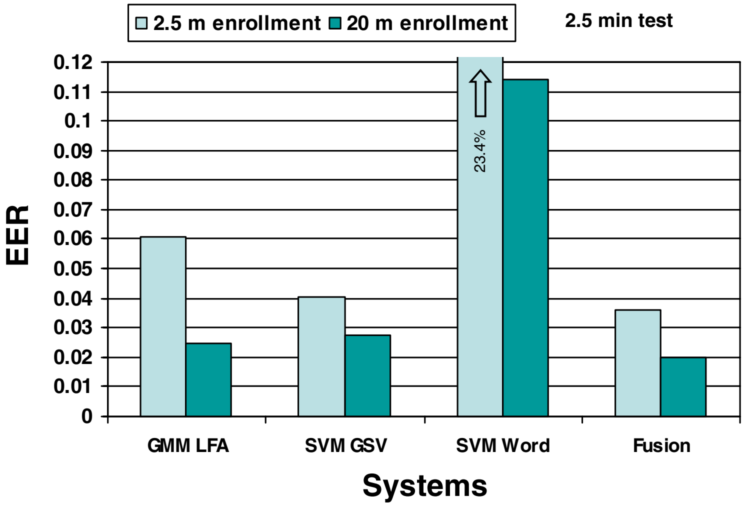

Fig. 10. The performance measure equal error rate for text-independent speaker verification systems

5 Conclusions

In this chapter, we have provided a brief overview of the classification methods used in speaker recognition. In Section 2, we presented some of the common feature extraction techniques that are currently being used in speaker recognition systems. In Section 3, we described classification methods that are representative of those currently being studied in research and used in application. We introduced common approaches for text-dependent and text-independent applications, as well as offering some historical evolution of how these classifiers came to be used.

Future work in speaker recognition will continue to exploit advances in speech science, classification, and engineering. Speech science continues to give insight into feature that characterize speakers—speaker idiolect, speaker dialect, as well as vocal characteristics (roughness, breathiness, etc.). More precise measurements and techniques for extracting these features will lead to more diverse and accurate speaker recognition systems.

Classification continues to be a strong component of the speaker recognition problem. Specialization of classification techniques to deal with speaker recognition challenges will no doubt lead to significant improvements. Current trends are methods that deal with channel variability, the continuum of feature types, and general mismatch.

Finally, engineering provides a feedback to all of the design techniques. Implementing and deploying technologies to different application domains—forensic, security, etc.—gives insight into robustness, computation, and fusion of speaker characterization techniques.

References

1. Davis, S.B., Mermelstein, P.: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. Acoust. Speech, Signal Processing, ASSP 28(4), 357–366 (1980)
2. Quatieri, T.: Discrete-Time Speech Signal Processing: Principles and Practice. Prentice-Hall, Englewood Cliffs (2001)
3. Reynolds, D.A., Quatieri, T.F., Dunn, R.: Speaker verification using adapted Gaussian mixture models. Digital Signal Processing 10(1-3), 19–41 (2000)
4. Tierney, J.: A study of LPC analysis of speech in additive noise. IEEE Trans. Acoust., Speech, Signal Processing, ASSP 28(4), 389–397 (1980)
5. Rabiner, L., Juang, B.-H.: Fundamentals of Speech Recognition. Prentice-Hall, Englewood Cliffs (1993)
6. Adami, A., Mihaescu, R., Reynolds, D.A., Godfrey, J.J.: Modeling prosodic dynamics for speaker recognition. In: Proc. ICASSP, pp. IV–788–IV–791 (2003)
7. Peskin, B., Navratil, J., Abramson, J., Jones, D., Klusacek, D., Reynolds, D., Xiang, B.: Using prosodic and conversational features for high-performance speaker recognition: Report from JHU workshop. In: Proc. ICASSP (2003)
8. Doddington, G.: Speaker recognition based on idiolectal differences between speakers. In: Proc. Eurospeech, pp. 2521–2524 (2001)
9. Navrátil, J., Jin, Q., Andrews, W.D., Campbell, J.P.: Phonetic speaker recognition using maximum-likelihood binary-decision tree models. In: Proc. ICASSP, pp. IV–796–IV–799 (2003)
10. Matsui, T., Furui, S.: Concatenated phoneme models for text-variable speaker recognition. In: Proc. ICASSP, vol. II, pp. 391–394 (1993)
11. Campbell, W.M., Campbell, J.P., Reynolds, D.A., Jones, D.A., Leek, T.R.: Phonetic speaker recognition with support vector machines. In: Thrun, S., Saul, L., Schölkopf, B. (eds.) Advances in Neural Information Processing Systems 16, MIT Press, Cambridge (2004)
12. Andrews, W.D., Kohler, M.A., Campbell, J.P., Godfrey, J.J., Hernandez-Cordero, J.: Gender-dependent phonetic refraction for speaker recognition. In: Proc. ICASSP, pp. I149–I153 (2002)
13. Bimbot, F., Bonastre, J.-F., Fredouille, C., Gravier, G., Meignier, S., Merlin, T., Ortega-Garc, J., Magrin-Chagnolleau, I., Petrovska-Delacretaz, D., Reynolds, D.A.: A tutorial on text-independent speaker verication. EURASIP Journal on Applied Signal Processing 4, 430–451 (2004)

14. Reynolds, D.A.: Speaker identification and verification using gaussian mixture speaker models. Speech Commun. 17(1-2), 91–108 (1995)
15. Carey, M., Parris, E., Bridle, J.: A speaker verification system using alpha-nets. In: Proc. ICASSP (1991)
16. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 39, 1–38 (1977)
17. Gauvain, J.-L., Lee, C.-H.: Maximum a posteriori estimation for multivariate Gaussian mixture observations of markov chains. IEEE Trans. Speech and Audio Processing 2(2), 291–298 (1994)
18. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. Wiley and Sons, New York (1973)
19. Soong, F., Rosenberg, A., Rabiner, L., Juang, B.: A vector quantization approach to speaker recognition. In: Proc. ICASSP, pp. 387–390 (1985)
20. Rosenberg, A., Soong, F.: Evaluation of a vector quantization talker recognition system in text independent and text dependent modes. In: Proc. ICASSP, pp. 873–876 (1986)
21. Campbell, W.M.: Generalized linear discriminant sequence kernels for speaker recognition. In: Proc. ICASSP, pp. 161–164 (2002)
22. Fine, S., Navrátil, J., Gopinath, R.A.: A hybrid GMM/SVM approach to speaker recognition. In: Proc. ICASSP (2001)
23. Wan, V., Renals, S.: SVMSVM: support vector machine speaker verification methodology. In: Proc. ICASSP, pp. 221–224 (2003)
24. Campbell, W.M., Campbell, J.P., Reynolds, D.A., Jones, D.A., Leek, T.R.: High-level speaker verification with support vector machines. In: Proc. ICASSP, pp. I–73–76 (2004)
25. Stolcke, A., Ferrer, L., Kajarekar, S., Shriberg, E., Venkataraman, A.: MLLR transforms as features in speaker recognition. In: Proc. Interspeech, pp. 2425–2428 (2005)
26. Campbell, W.M., Sturim, D.E., Reynolds, D.A., Solomonoff, A.: SVM based speaker verification using a GMM supervector kernel and NAP variability compensation. In: Proc. ICASSP, pp. I–97–I–100 (2006)
27. Cristianini, N., Shawe-Taylor, J.: Support Vector Machines. Cambridge University Press, Cambridge (2000)
28. Collobert, R., Bengio, S.: SVMTorch: Support vector machines for large-scale regression problems. Journal of Machine Learning Research 1, 143–160 (2001)
29. Louradour, J., Daoudi, K., Bach, F.: SVM speaker verification using an incomplete cholesky decomposition sequence kernel. In: IEEE 2006 Odyssey: The Speaker and Language Recognition Workshop (2006)
30. Mariéthoz, J., Bengio, S.: A max kernel for text-independent speaker verification systems. In: Second Workshop on Multimodal User Authentication (2006)
31. Soong, F.K., Rosenberg, A.E.: On the use of instantaneous and transitional spectral information in speaker recognition. In: Proc. ICASSP, pp. 877–880 (1986)
32. Matsui, T., Furui, S.: Speaker recognition using concatenated phoneme models. In: Proc. ICSLP (1992)
33. Rosenberg, A.E., Parthasarathy, S.: Speaker background models for connected digit password speaker verification. In: Proc. ICASSP, pp. 81–84 (1996)
34. Corrada-Emmanuel, A., Newman, M., Peskin, B., Gillick, L., Roth, R.: Progress in speaker recognition at dragon systems. In: Proc. ICSLP (1998)
35. Weber, F., Peskin, B., Newman, M., Corrada-Emmanuel, A., Gillick, L.: Speaker recognition on single- and multispeaker data. Digital Signal Processing 10, 75–92 (2000)

36. Rabiner, L.R., Juang, B.H.: An introduction to hidden markov models. IEEE ASSP Mag. 3 (1986)
37. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. of the IEEE 77(2), 257–285 (1989)
38. Campbell, J.P.: Speaker recognition: A tutorial. Proc. of the IEEE 85(9), 1437–1462 (1997)
39. Newman, M., Gillick, L., Ito, Y., McAllaster, D., Peskin, B.: Speaker verification through large vocabulary continuous speechrecognition. In: Proc. ICSLP (1996)
40. Matsui, T., Furui, S.: Likelihood normalization for speaker verification using phoneme- and speaker-independent model. In: Speech Communication (1995)
41. Farrell, K.R., Mammone, R.J., Assaleh, K.T.: Speaker recognition using neural networks and conventional classifiers. IEEE Trans. on Speech and Audio Processing 2(1), 194–205 (1994)
42. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
43. Oglesby, J., Mason, J.: Radial basis function networks for speaker recognition. In: Proc. ICASSP, pp. 393–396 (May 1991)
44. Hermansky, H., Morgan, N., Bayya, A., Kohn, P.: Compensation for the effect of communication channel in auditory-like analysis of speech (RASTA-PLP). In: Proc. Eurospeech, pp. 1367–1371 (1991)
45. Atal, B.: Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. Journal of the Acoustical Society of America 55(6), 1304–1312 (1974)
46. Mansour, D., Juang, B.: A family of distortion measures based upon projection operation for robust speech recognition. IEEE Trans. Acoust., Speech, Signal Processing, ASSP 37, 1659–1671 (1989)
47. Pelecanos, J., Sridharan, S.: Feature warping for robust speaker verification. In: Proc. of Speaker Odyssey Workshop, pp. 213–218 (2001)
48. Reynolds, D.A.: Channel robust speaker verification via feature mapping. In: Proc. ICASSP, vol. 2, pp. II–53–56 (2003)
49. Teunen, R., Shahshahani, B., Heck, L.: A model-based transformational approach to robust speaker recognition. In: Proc. ICSLP (2000)
50. Kenny, P., Boulianne, G., Dumouchel, P.: Eigenvoice modeling with sparse training data. IEEE Trans. Speech and Audio Processing 13(3), 345–354 (2005)
51. Vogt, R., Baker, B., Sriharan, S.: Modelling session variability in text-independent speaker verification. In: Proc. Interspeech, pp. 3117–3120 (2005)
52. Solomonoff, A., Campbell, W.M., Boardman, I.: Advances in channel compensation for SVM speaker recognition. In: Proc. ICASSP (2005)
53. Auckenthaler, R., Carey, M., Lloyd-Thomas, H.: Score normalization for text-independent speaker verification systems. Digital Signal Processing 10, 42–54 (2000)
54. Reynolds, D.A.: Comparison of background normalization methods for text independent speaker verification. In: Proc. Eurospeech, pp. 963–966 (1997)
55. Heck, L., Weintraub, M.: Handset-dependent background models for robust text-independent speaker recognition. In: Proc. ICASSP, pp. 1071–1074 (1997)
56. Campbell, W.M., Navratil, J., Reynolds, D.A., Shen, W., Sturim, D.E.: The MIT/IBM 2006 speaker recognition system:High-performance reduced complexity recognition. In: ICASSP (2007)
57. Reynolds, D.A., Campbell, W., Gleason, T., Quillen, C., Sturim, D., Torres-Carrasquillo, P., Adam, A.: The 2004 MIT Lincoln Laboratory speaker recognition system. In: ICASSP (2005)