1 Introduction

This chapter defines the term "speaker recognition" and looks at this technology from a high-level perspective. Then, it introduces the fundamental concepts of speaker recognition, including feature extraction, speaker modeling, scoring, and performance measures.

1.1 Fundamentals of Speaker Recognition

From time to time we hear that the information of millions of customers of remote services has been compromised. These security leaks cause concerns about the security of the remote services that everyone uses on a daily basis. While these remote services bring convenience and benefit to users, they are also gold mines for criminals to carry out fraudulent acts. The conventional approach to user authentication, such as usernames and passwords, is no longer adequate for securing these services. A number of companies have now introduced voice biometrics as a complement to the conventional username–password approach. With this new authentication method, it is much harder for the criminals to imitate the legitimate users. Voice biometrics can also reduce the risk of leaking customers' information caused by social engineering fraudulence. Central to voice biometrics authentication is speaker recognition.

Another application domain of voice biometrics is to address the privacy issues of smartphones, home assistants, and smart speakers. With the increasing intelligence capabilities of these devices, we can interact with them as if they were human. Because these devices are typically used solely by their owners or their family members and speech is the primary means of interaction, it is natural to use the voice of the owners for authentication, i.e., a device can only be used by its owner.

Speaker recognition is a technique to recognize the identity of a speaker from a speech utterance. As shown in Figure 1.1, in terms of recognition tasks, speaker recognition can be categorized into speaker identification, speaker verification, and speaker diarization. In all of these tasks, the number of speakers involved can be fixed (closed set) or varied (open set).

Speaker identification is to determine whether the voice of an unknown speaker matches one of the N speakers in a dataset, where N could be very large (thousands). It is a one-to-many mapping and it is often assumed that the unknown voice

https://doi.org/10.1017/9781108552332.002 Published online by Cambridge University Press

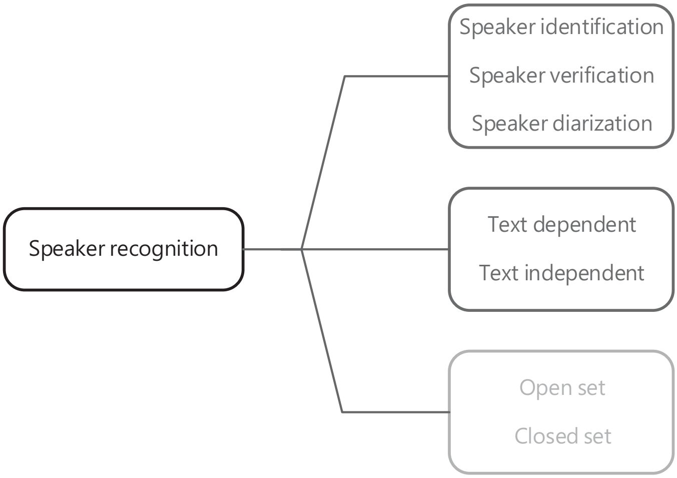


Figure 1.1 Categorization of speaker recognition in terms of tasks (top), text restriction (middle), and datasets (bottom).

must come from a set of known speakers – referred to as closed-set identification. Adding a “none of the above” option to closed-set identification gives us open-set identification.

Speaker verification is to determine whether the voice of an unknown speaker matches a *specific* speaker. It is a one-to-one mapping. In closed-set verification, the population of clients is fixed, whereas in open-set verification, new clients can be added without having to redesign the system.

Speaker diarization [5] is to determine when speaker changes have occurred in speech signals, which is an analogy to the speech segmentation task in speech recognition. When it is also needed to group together speech segments corresponding to the same speaker, we have speaker clustering. In both cases, prior speaker information may or may not be available.

There are two input modes in speaker recognition systems: text dependent and text independent. Text-dependent recognition systems know the texts that will be spoken by the speakers or expect legitimate users to speak some fixed or prompted phrases. The phrases are typically very short. Because the phrases are known, speech recognition can be used for checking spoken text to improve system performance. Text-dependent systems are mainly used for applications with strong control over user input, e.g., biometric authentication. On the contrary, in text-independent systems, there is no restriction on the spoken text. Typically, conversational speech is used as the input. So, the sentences are much longer than those in text-dependent systems. While the spoken text is unknown, speech recognition can still be used for extracting high-level features to boost performance. Text-independent systems are mainly used in applications with less control over user input, e.g., forensic speaker ID. Compared with text-dependent systems, text-independent systems are more flexible but recognition is more difficult.

https://doi.org/10.1017/9781108552332.002 Published online by Cambridge University Press

1.2 Feature Extraction

Speech is a time-varying signal conveying multiple layers of information, including words, speaker identities, acoustic features, languages, and emotions. Information in speech can be observed in the time and frequency domains. The most widely used visualization tool is the spectrogram in which the frequency spectra of consecutive short-term speech segments are displayed as an image. In the image, the horizontal and vertical dimensions represent time and frequency, respectively, and the intensity of each point in the image indicates the magnitude of a particular frequency at a particular time.

While spectrograms are great for visualization of speech signals, they are not appropriate for speech and speaker recognition. There are two reasons for this. First, the frequency dimension is still too high. For 1024-point fast Fourier transform (FFT), the frequency dimension is 512, which is far too large for statistical modeling. Second, the frequency components after FFT are highly correlated with each other, which do not facilitate the use of diagonal covariance matrices to model the variability in the feature vectors. To obtain a more compact representation of speech signals and to de-correlate the feature components, cepstral representation of speech is often used. The most widely used representation is the Mel-frequency cepstral coefficients (MFCCs) [6]. Figure 1.2 shows the process of extracting MFCCs from a frame of speech. In the figure, $s(n)$ represents a frame of speech, $X(m)$ is the logarithm of the spectrum at frequencies defined by the mth filter in the filter bank, and

$$o_i = \sum_{m=1}^{M} \cos\left[i \left(m - \frac{1}{2} \right) \frac{\pi}{M} \right] X(m), \quad i = 1, \ldots, P \tag{1.1}$$

are its MFCCs. In Figure 1.2, the symbols Δ and $\Delta\Delta$ represent velocity and acceleration of MFCCs, respectively. Denoting e as the log-energy, an acoustic vector corresponding to $s(n)$ is given by

$$\mathbf{o} = [e, o_1, \ldots, o_P, \Delta e, \Delta o_1 \ldots, \Delta o_P, \Delta\Delta e, \Delta\Delta o_1, \ldots, \Delta\Delta o_P]^{\mathsf{T}}. \tag{1.2}$$

In most speaker recognition systems, $P = 19$, giving $\mathbf{o} \in \Re^{60}$.

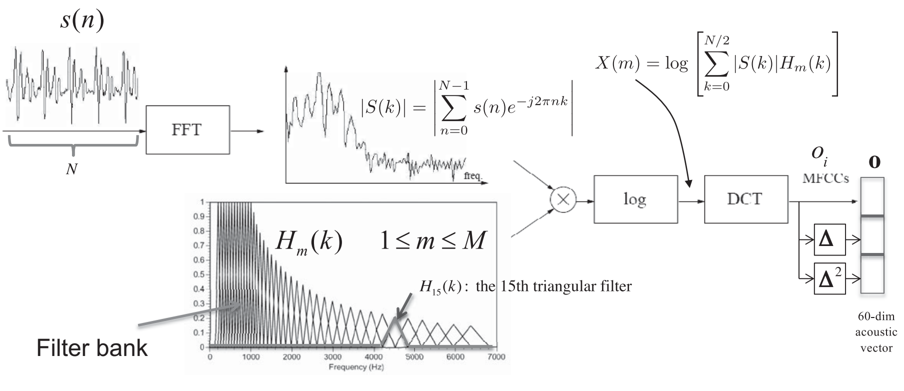


Figure 1.2 Process of extracting MFCCs from a frame of speech. Refer to Eq. 1.1 and Eq. 1.2 for o_i and $\mathbf{o}$, respectively.

https://doi.org/10.1017/9781108552332.002 Published online by Cambridge University Press

1.3 Speaker Modeling and Scoring

For text-independent speaker recognition, we assume that the acoustic vectors are independent. Therefore, speaker modeling amounts to modeling a batch of independent acoustic vectors derived from speech waveform as shown in Figure 1.3. The process can be considered as mapping the consecutive frames independently to an acoustic space (Figure 1.4(a)). In the most traditional approach, the distribution of these vectors is represented by a Gaussian mixture model (GMM) as shown in Figure 1.4(b).

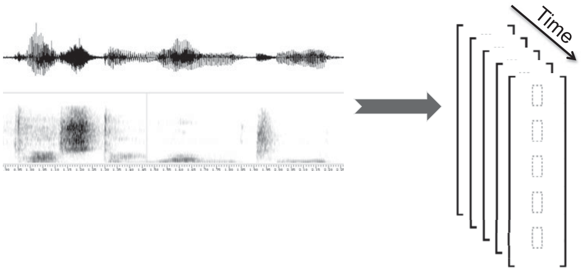
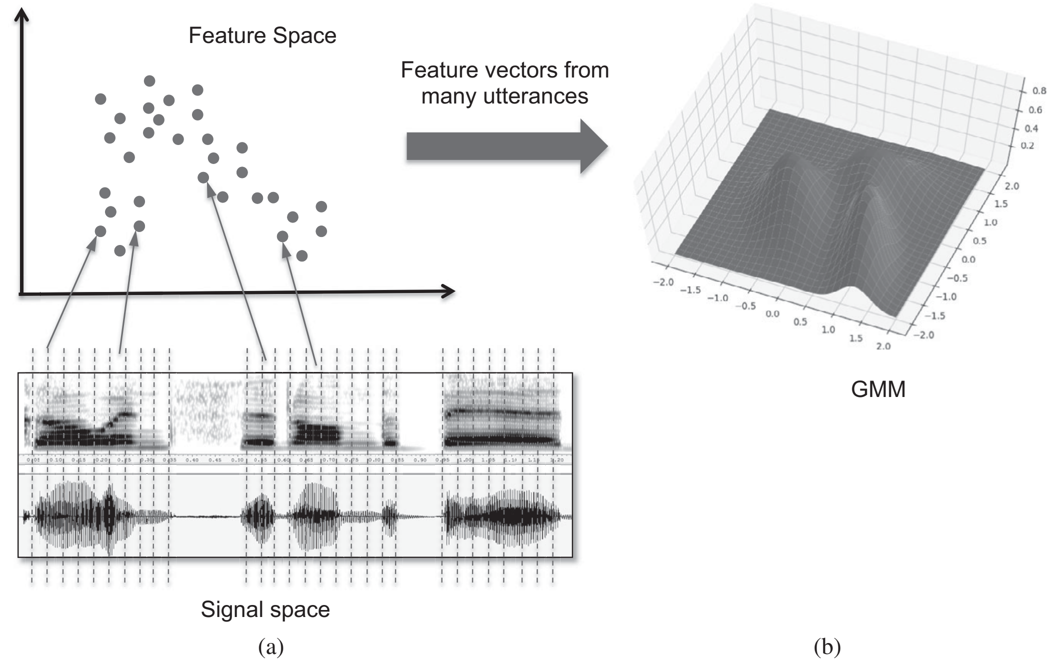


Figure 1.3 From waveform to a sequence of acoustic vectors.



(a) (b)

Figure 1.4 Modeling the statistical properties of acoustic vectors by a Gaussian mixture model.

https://doi.org/10.1017/9781108552332.002 Published online by Cambridge University Press

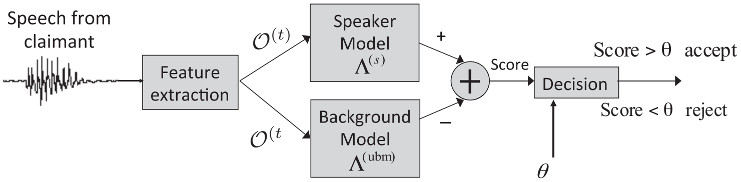


Figure 1.5 GMM–UBM scoring process.

1.3.1 Speaker Modeling

In 2000, Reynolds [7] proposed using the speech of a large number of speakers to train a GMM to model the acoustic characteristics of a general population. The resulting model is called the universal background model (UBM). The density of acoustic vectors **o**'s is given by

$$p(\mathbf{o}|\text{UBM}) = p(\mathbf{o}|\Lambda^{\text{ubm}}) = \sum_{c=1}^{C} \pi_c^{\text{ubm}} \mathcal{N}(\mathbf{o}|\boldsymbol{\mu}_c^{\text{ubm}}, \boldsymbol{\Sigma}_c^{\text{ubm}}). \tag{1.3}$$

The UBM parameters $\Lambda^{\text{ubm}} = \{\pi_c^{\text{ubm}}, \boldsymbol{\mu}_c^{\text{ubm}}, \boldsymbol{\Sigma}_c^{\text{ubm}}\}_{c=1}^{C}$ are estimated by the expectation-maximization (EM) algorithm [8] using the speech of many speakers. See Section 3.1.1 for the detail of the EM algorithm.

While the UBM represents the general population, individual speakers are modeled by speaker-dependent Gaussian mixture models. Specifically, for a target-speaker s, his/her GMM is given by

$$p(\mathbf{o}|\text{Spk } s) = p(\mathbf{o}|\Lambda^{(s)}) = \sum_{c=1}^{C} \pi_c^{(s)} \mathcal{N}(\mathbf{o}|\boldsymbol{\mu}_c^{(s)}, \boldsymbol{\Sigma}_c^{(s)})$$

where $\Lambda^{(s)} = \{\pi_c^{(s)}, \boldsymbol{\mu}_c^{(s)}, \boldsymbol{\Sigma}_c^{(s)}\}_{c=1}^{C}$ are learned by using a maximum *a posteriori* (MAP) adaptation [7]. See Section 3.1.3 for the details of a MAP adaptation.

1.3.2 Speaker Scoring

Given the acoustic vectors $O^{(t)}$ from a test speaker and a claimed identity s, speaker verification amounts to computing the log-likelihood ratio:

$$S_{\text{GMM–UBM}}(O^{(t)}|\Lambda^{(s)}, \Lambda^{\text{ubm}}) = \log p(O^{(t)}|\Lambda^{(s)}) - \log p(O^{(t)}|\Lambda^{\text{ubm}}), \tag{1.4}$$

where $\log p(O^{(t)}|\Lambda^{(s)})$ is the log-likelihood of $O^{(t)}$ given the speaker model $\Lambda^{(s)}$. Figure 1.5 shows the scoring process.

1.4 Modern Speaker Recognition Approaches

GMM–UBM is a frame-based approach in that the speaker models (GMMs) describe the distribution of acoustic frames. Since its introduction in 2000, it has been the

https://doi.org/10.1017/9781108552332.002 Published online by Cambridge University Press

state-of-the-art method for speaker verification for a number of years. However, it has its own limitations. One major drawback is that the training of the GMMs and the UBM is disjointed, meaning that contrastive information between target speakers and impostors cannot be explicitly incorporated into the training process. Another drawback is that suppressing nonspeaker information (e.g., channel and noise) is difficult. Although attempts have been made to suppress channel and noise variabilities in the feature [9, 10], model [11, 12], and score domains [13], they are not as effective as the modern approaches outlined below and explained in detail in this book.

In 2006, researchers started to look at the speaker verification problem from another view. Instead of accumulating the frame-based log-likelihood scores of an utterance, researchers derived methods to map the acoustic characteristics of the entire utterance to a high-dimensional vector. These utterance-based vectors live on a high-dimensional space parameterized by GMMs. Because the dimension is the same regardless of the utterance duration, standard machine learning methods such as support vector machines and factor analysis can be applied on this space. The three influential methods based on this idea are GMM–SVM, joint factor analysis, and i-vectors.

In GMM–SVM [14] (see Section 3.2), supervectors are constructed from MAP-adapted target-speaker GMMs. For each target speaker, a speaker-dependent SVM is then trained to discriminate his/her supervectors from those of the impostors. To reduce channel mismatch, the directions corresponding to nonspeaker variability are projected out. Scoring amounts to computing the SVM scores of the test utterances and decisions are made by comparing the scores with a decision threshold.

In joint factor analysis [15] (see Section 3.7), speaker and session variabilities are represented by latent variables (speaker factors and channel factors) in a factor analysis model. During scoring, session variabilities are accounted for by integrating over the latent variables, e.g., the channel factors.

In an i-vector system [16] (see Section 3.6), utterances are represented by the posterior means of latent factors, called the i-vectors. I-vectors capture both speaker and channel information. During scoring, the unwanted channel variability is removed by linear discriminant analysis (LDA) or by integrating out the latent factors in a probabilistic LDA model [17].

1.5 Performance Measures

For closed-set speaker identification, recognition rate (accuracy) is the usual performance measure:

$$\text{Recognition rate} = \frac{\text{No. of correct recognitions}}{\text{Total no. of trials}}.$$

Speaker verification, on the other hand, has a rich set of performance measures. Although different datasets have slightly different measures, their principles remain the same. The common measures include false rejection rate (FRR), false acceptance rate (FAR), equal error rate (EER), minimum decision cost function (minDCF), and actual decision cost function (DCF).

https://doi.org/10.1017/9781108552332.002 Published online by Cambridge University Press

1.5.1 FAR, FRR, and DET

The definition of FAR and FRR are as follows:

$$\begin{aligned}\text{False rejection rate (FRR)} &= \text{Miss probability}\\ &= \frac{\text{No. of true-speakers rejected}}{\text{Total no. of true-speaker trials}};\\ \text{False acceptance rate (FAR)} &= \text{False alarm probability}\\ &= \frac{\text{No. of impostors accepted}}{\text{Total no. of impostor attempts}}.\end{aligned}$$

Equal error rate (EER) corresponds to the operating point at which FAR = FRR. The concept of FAR, FRR, and EER can be explained by using the distributions of speaker scores and impostor scores in two speaker verification systems (System A and System B) shown in Figure 1.6. When the decision threshold θ is swept from low to high, FAR drops from 100 percent gradually to 0 percent but FRR gradually increases from 0 percent to 100 percent. When θ is very large, we have a secure but user-unfriendly system. On the other hand, when θ is very small, we have a user-friendly but nonsecure system. A system developer chooses a decision threshold such that the FAR and FRR meet the requirements of the application.

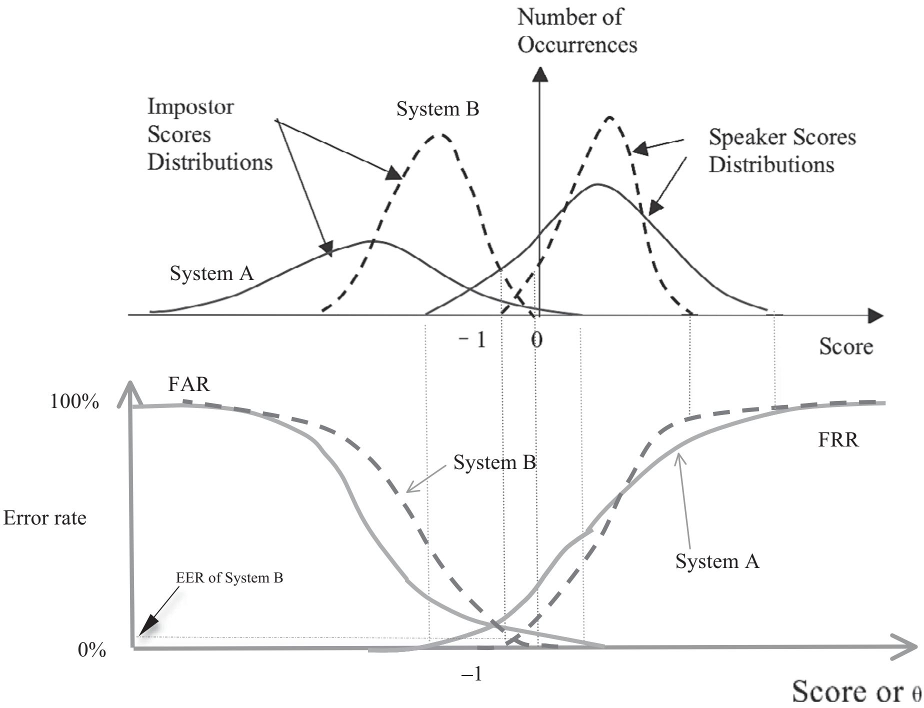


Figure 1.6 *Top:* Distributions of true-speaker scores and impostor scores of two speaker verification systems. *Bottom*: The FAR, FRR, and EER of the two systems.

https://doi.org/10.1017/9781108552332.002 Published online by Cambridge University Press

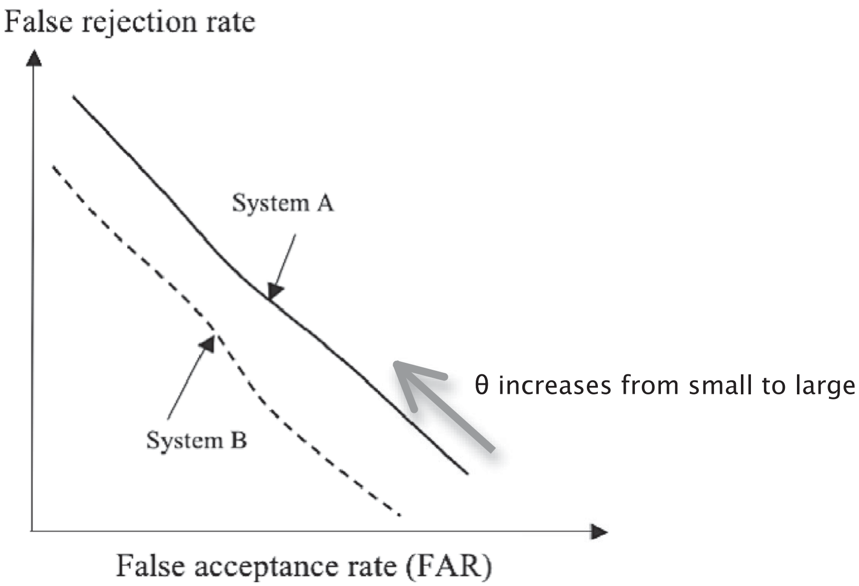


Figure 1.7 The DET curves correspond to system A and system B in Figure 1.6. System B performs better because its DET curve is closer to the origin.

Detection error tradeoff (DET) curves [18] are similar to receiver operating characteristic curves but with nonlinear x- and y-axis. The advantage of the nonlinear axes is that the DET curves of systems with verification scores following Gaussian distributions will be displayed as straight lines, which facilitate comparison of systems with similar performance. A DET curve is produced by sweeping the decision threshold of a system from low to high. Figure 1.7 shows the DET curves of System A and System B in Figure 1.6.

1.5.2 Decision Cost Function

The decision cost function (DCF) [18] is a weighted sum of the FRR ($P_{\text{Miss|Target}}$) and FAR ($P_{\text{FalseAlarm|Nontarget}}$):

$$\begin{aligned} C_{\text{Det}}(\theta) = C_{\text{Miss}} \times P_{\text{Miss|Target}}(\theta) \times P_{\text{Target}} + \\ C_{\text{FalseAlarm}} \times P_{\text{FalseAlarm|Nontarget}}(\theta) \times (1 - P_{\text{Target}}), \end{aligned}$$

where θ is a decision threshold, C_{Miss} is the cost of false rejection, $C_{\text{FalseAlarm}}$ is the cost of false acceptance, and P_{Target} is the prior probability of target speakers. With the DCF, a normalized cost can be defined:

$$C_{\text{Norm}}(\theta) = C_{\text{Det}}(\theta)/C_{\text{Default}},$$

https://doi.org/10.1017/9781108552332.002 Published online by Cambridge University Press

where

$$C_{\text{Default}} = \min \begin{cases} C_{\text{Miss}} \times P_{\text{Target}} \\ C_{\text{FalseAlarm}} \times (1 - P_{\text{Target}}). \end{cases} \tag{1.5}$$

The normalized DCF and the measures derived from it are the primary performance index of the NIST speaker recognition evaluations (SRE).[1] The parameters in Eq. 1.5 are set differently for different years of evaluations:

- SRE08 and earlier:

$$C_{\text{Miss}} = 10; \quad C_{\text{FalseAlarm}} = 1; \quad P_{\text{Target}} = 0.01.$$

- SRE10:

$$C_{\text{Miss}} = 1; \quad C_{\text{FalseAlarm}} = 1; \quad P_{\text{Target}} = 0.001.$$

In SRE12, the decision cost function has been changed to:

$$\begin{aligned} C_{\text{Det}}(\theta) = C_{\text{Miss}} \times P_{\text{Miss|Target}}(\theta) \times P_{\text{Target}} + C_{\text{FalseAlarm}} \times (1 - P_{\text{Target}}) \\ \times \left[P_{\text{FalseAlarm|KnownNontarget}}(\theta) \times P_{\text{Known}} \right. \\ \left. + P_{\text{FalseAlarm|UnKnownNontarget}} \times (1 - P_{\text{Known}})\right], \end{aligned} \tag{1.6}$$

where “KnownNontarget” and “UnknownNontarget” mean that the impostors are known and unknown to the evaluator, respectively. Similar to previous years’ SRE, the DCF is normalized, giving

$$C_{\text{Norm}}(\theta) = C_{\text{Det}}(\theta)/(C_{\text{Miss}} \times P_{\text{Target}}). \tag{1.7}$$

The parameters for core test conditions are set to

$$C_{\text{Miss}} = 1; \; C_{\text{FalseAlarm}} = 1; \; P_{\text{Target1}} = 0.01; \; P_{\text{Target2}} = 0.001; \; P_{\text{Known}} = 0.5.$$

Substituting the above P_{Target1} and P_{Target2} into Eq. 1.6 and Eq. 1.7, we obtain $C_{\text{Norm1}}(\theta_1)$ and $C_{\text{Norm2}}(\theta_2)$, respectively. Then, the primary cost of NIST 2012 SRE can be obtained:

$$C_{\text{Primary}}(\theta_1, \theta_2) = \frac{C_{\text{Norm1}}(\theta_1) + C_{\text{Norm2}}(\theta_2)}{2}. \tag{1.8}$$

In SRE16, the decision cost was changed to

$$\begin{aligned} C_{\text{Det}}(\theta) = C_{\text{Miss}} \times P_{\text{Miss|Target}}(\theta) \times P_{\text{Target}} + \\ C_{\text{FalseAlarm}} \times P_{\text{FalseAlarm|Nontarget}}(\theta) \times (1 - P_{\text{Target}}). \end{aligned} \tag{1.9}$$

The normalized DCF remains the same as Eq. 1.7, and the parameters for the core test conditions are:

$$C_{\text{Miss}} = 1; \; C_{\text{FalseAlarm}} = 1; \; P_{\text{Target1}} = 0.01; \; P_{\text{Target2}} = 0.001.$$

The primary cost is computed as in Eq. 1.8.

[1] www.nist.gov/itl/iad/mig/speaker-recognition

https://doi.org/10.1017/9781108552332.002 Published online by Cambridge University Press

Table 1.1 Cost parameters and prior of target-speakers in NIST 2018 SRE.

Speech Type	Parameter ID	C_{Miss}	$C_{\text{FalseAlarm}}$	P_{Target}
CTS	1	1	1	0.01
CTS	2	1	1	0.005
AfV	3	1	1	0.05

Note that in Eq. 1.8, the decision thresholds θ_1 and θ_2 are assumed unknown and can be set by system developers. When θ_1 and θ_2 are optimized to achieve the lowest C_{Norm1} and C_{Norm2}, respectively, then we have the minimum primary cost. Sometimes, researchers simply call it minDCF.

In practical applications of speaker verification, application-independent decision thresholds [19] are more appropriate. The goal is to minimize not only the EER and minDCF but also the actual DCF (actDCF) or C_{primary} at specific thresholds. For each SRE, NIST defined specific thresholds to evaluate how well systems were calibrated. For example in SRE16, θ_1 and θ_2 were set to 4.5951 and 5.2933, respectively. The primary cost using these two thresholds are called actual DCF or actual primary cost. Specifically, we have

$$C_{\text{actPrimary}} = \frac{C_{\text{Norm1}}(4.5951) + C_{\text{Norm2}}(5.2933)}{2}.$$

Any systems with scores that deviate significantly from these two thresholds will get a high $C_{\text{actPrimary}}$, which could be larger than 1.0.

The decision cost function in SRE18 is identical to that of SRE16 in Eq. 1.9. However, there are two types of speech: conversational telephone speech (CTS) and audio from video (AfV). The prior of target-speakers are different for these two types of speech, as shown in Table 1.1. With the three sets of parameters, the primary cost in Eq. 1.8 is extended to

$$C_{\text{Primary}}(\theta_1, \theta_2, \theta_3) = \frac{1}{2}\left[\frac{C_{\text{Norm1}}(\theta_1) + C_{\text{Norm2}}(\theta_2)}{2} + C_{\text{Norm3}}(\theta_3)\right]. \tag{1.10}$$

Using the P_{Target}'s in Table 1.1, the actual DCF in SRE18 is

$$C_{\text{Primary}} = \frac{1}{2}\left[\frac{C_{\text{Norm1}}(4.5951) + C_{\text{Norm2}}(5.2933)}{2} + C_{\text{Norm3}}(2.9444)\right].$$

https://doi.org/10.1017/9781108552332.002 Published online by Cambridge University Press