5 Robust Speaker Verification

5.1 DNN for Speaker Verification

Recently, deep learning and deep neural networks (DNNs) have changed the research landscape in speech processing [139–142]. This is mainly due to their superb performance. In speaker recognition, the most common strategy is to train a DNN with a bottleneck layer from which either frame-based features or utterance-based features can be extracted. The network can be trained to produce senone posteriors or speaker posteriors. For the latter, a pooling procedure is applied to convert variable-length utterances into a fixed-length feature vector. Then, standard back-end classifiers can be applied for scoring.

The relationship between i-vectors and background noise is not straightforward and may not be linear. Linear models such as PLDA and LDA cannot fully capture this complex relationship. Recently, a number of studies have demonstrated that DNNs are more capable of modeling this complex relationship. For instance, Isik et al. [143] extracted speaker vectors from i-vectors through disentangling the latent dependence between speaker and channel components, and [144, 145] used the weights of a stacked restricted Boltzmann machine (RBM) to replace the parameters of a PLDA model. In [146], noisy i-vectors were mapped to their clean counterparts by a discriminative denoising autoencoder (DDAE) using the speaker identities and the information in the clean i-vectors.

5.1.1 Bottleneck Features

DNNs have been used as frame-based feature extractors. This is achieved by reading the activations of the bottleneck layer of a DNN that is trained to produce the senone posteriors of a short segment of speech [147, 148]. The procedure of this approach is illustrated in Figure 5.1. The main idea is to replace the MFCC-UBM by the bottleneck-feature based UBM (BNF-UBM). As the DNN is trained to produce senone posteriors, the bottleneck features contain more phonetic information and the alignments with the BNF-UBM are more relevant and reliable for speaker verification. In addition to computing the zeroth-order statistics (alignment), the bottleneck features can also be used as acoustic features for i-vector extraction.

Alternatively, bottleneck features can be extracted at the conversation level by averaging the activation of the last hidden layer [149, 150]. This procedure leads to the

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

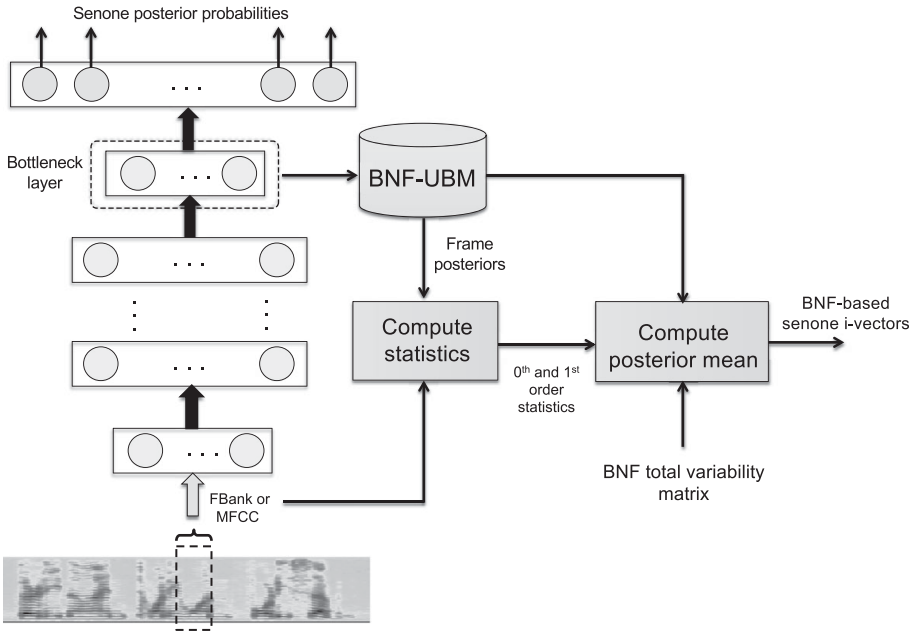


Figure 5.1 The procedure of extracting i-vectors from DNN-based bottleneck features. Instead of using an MFCC-based UBM for frame alignment, an UBM trained by DNN-based bottleneck features (BNF) is used.

d-vectors. The structure of the DNN is almost the same as Figure 5.1 except that the DNN is trained to output speaker posteriors instead of senone posteriors. This straightforward use of DNNs, however, can barely achieve significant performance gain, despite some success under reverberant environments [151] or with the help of a denoising autoencoder [2, 148]. The idea is illustrated in Figure 5.2.

5.1.2 DNN for I-Vector Extraction

One of the promising approaches is to use a DNN to compute the frame posteriors (senone posteriors) for i-vector extraction [3, 84, 85, 152–154] (see Section 3.6.10). In this method, the component posteriors of a GMM-based universal background model (UBM) are replaced by the output of a phonetically aware DNN. This means that an UBM is not required for frame alignment. Instead, the DNN estimates the posterior probabilities of thousands of senones given multiple contextual acoustic frames. However, as the i-vector extractor is still based on a GMM-based factor analyzer, the total variability matrix is still formed by stacking multiple loading matrices, one for each Gaussian. The resulting i-vectors are known as senone i-vectors or DNN i-vectors. Figure 5.3 shows the procedure of senone i-vector extraction.

The idea of replacing the UBM by a DNN is fascinating because the latter facilitates the comparison of speakers as if they were pronouncing the same context.

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

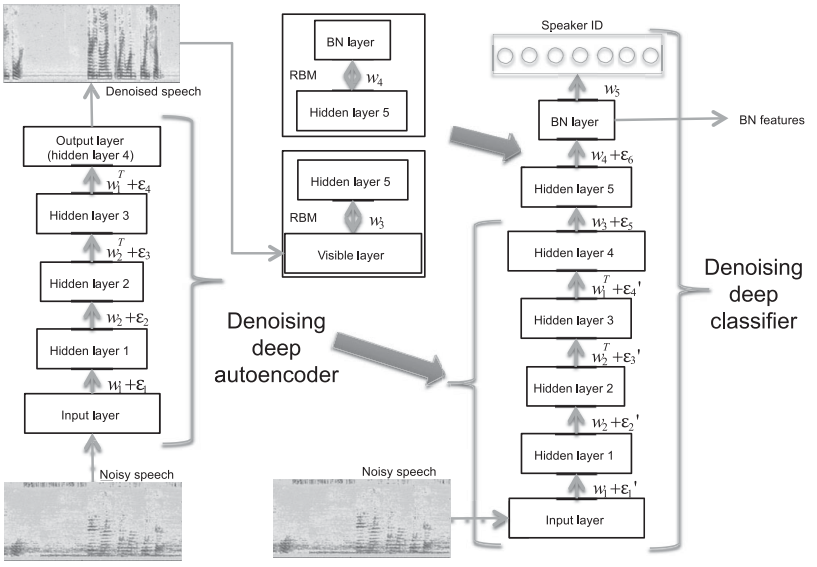


Figure 5.2 Procedure for creating a DNN classifier by stacking restricted Boltzmann machines and the extraction of bottleneck (BN) features from the BN layer. A softmax layer outputting speaker IDs is put on top of the denoising autoencoder. After backpropagation fine-tuning, BN features extracted from the BN layer can be used for speaker recognition [2].

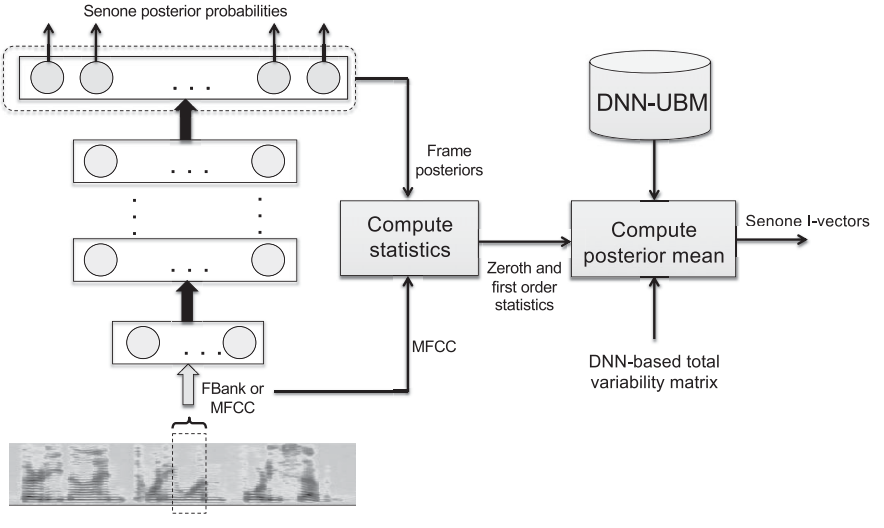


Figure 5.3 Extracting senone i-vectors. The DNN–UBM contains the mean vectors and covariance matrices computed from the frame posteriors and acoustic vectors (typically MFCCs) using Eq. 3.161 and Eq. 3.163, respectively. The DNN-based total variability matrix is obtained by Eq. 3.129 using the frame posteriors computed by the DNN. In some systems [3], the MFCCs are replaced by bottleneck features.

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

5.2 Speaker Embedding

The use of deep neural networks to create an embedded space that captures most of the speaker characteristics has attracted a lot of attention in the last few years. Strictly speaking, the d-vector [150] is also a kind of speaker embedding. However, the most promising approach today is the x-vectors [155, 156].

5.2.1 X-Vectors

Despite the outstanding performance of the senone i-vectors, training the DNN to output senone posteriors is computationally demanding and requires language-specific resources such as transcribed data. So far, senone i-vectors are limited to English only.

To remove the language constraint in senone i-vectors, Snyder et al. [155, 156] proposed to train a network to output speaker posteriors instead of senone posteriors. The concept is similar to the d-vector. However, unlike the d-vector, a time-delay neural network (TDNN) is used to capture the temporal information in the acoustic frames and statistical pooling is applied to aggregate the frame-level information into utterance-level information. Statistical pooling computes the mean and standard deviation of the TDNN's outputs across consecutive frames in the utterance. Speaker-dependent embedded features are then extracted at the layer(s) after the pooling. The authors name the resulting feature vectors as x-vectors. The TDNN and the pooling mechanism is shown to be very effective in converting acoustic frames into fixed-length feature vectors for PLDA scoring. Figure 5.4 shows the structure of the x-vector extractor.

In addition to the TDNN and the statistical pooling mechanism, another difference between the x-vector extractor and the d-vector extractor is that the former outputs the speaker posteriors at the utterance-level, whereas the latter outputs the speaker posterior at the frame level.

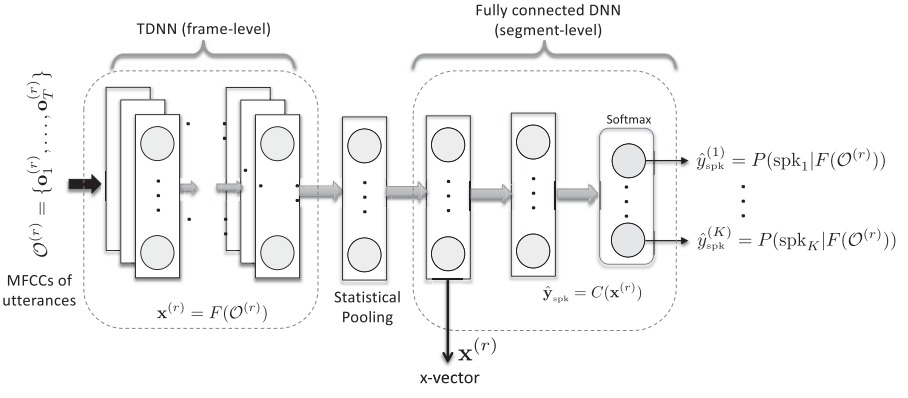


Figure 5.4 Structure of an x-vector extractor. The TDNN aims to capture the temporal information of contextual acoustic frames and the pooling layer aggregates the temporal information to form the utterance-level representation called the x-vector.

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

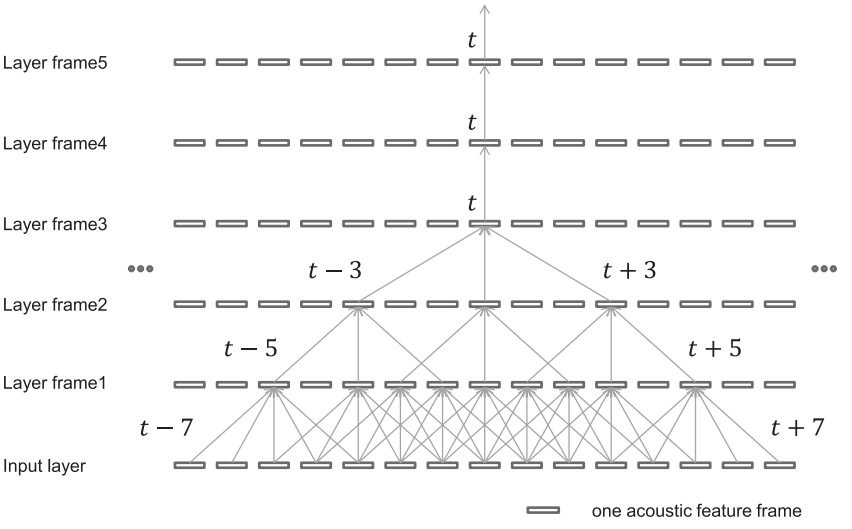


Figure 5.5 The frame context at different layers of an x-vector extraction DNN (courtesy of Youzhi Tu).

The TDNN structure allows neurons in the upper layers to receive signals that span across multiple frames. For example, in [155], at a particular frame t of an input utterance, the first hidden layer receives five frames between $t-2$ and $t+2$ from the input layer; the second hidden layer receives three frames at $t-2$, t, and $t+2$ from the first hidden layer; the third hidden layer receives three frames at $t-3$, t, and $t+3$ from the second hidden layer. As a result, each neuron in the third hidden layer will have a temporal span of 15 frames, as shown in Figures 5.5 and 5.6.

The authors in [156] shows that their x-vector system significantly outperforms their i-vector systems in the Cantonese part of NIST 2016 SRE. It was demonstrated that the performance of x-vectors can be further improved by increasing the diversity of training speech. This was done by adding noise, music, babble, and reverberation effect to the original speech samples. The combined speech samples are then used for training the DNN in Figure 5.4.

Very recently, the x-vector embedding has been enhanced by adding LSTM layers on top of the last TDNN layer, followed by applying statistical pooling on both the TDNN and LSTM layers [157, 158]. Another improvement is to add an attention mechanism to the statistical pooling process [159, 160], which are shown to be better than simply averaging the frame-level features.

5.2.2 Meta-Embedding

The idea of meta-embedding [62] is motivated by the difficulty of propagating the uncertainty of i-vectors and x-vectors to the PLDA models. Although it has been shown that the uncertainty of i-vectors (represented by the their posterior covariances) can

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

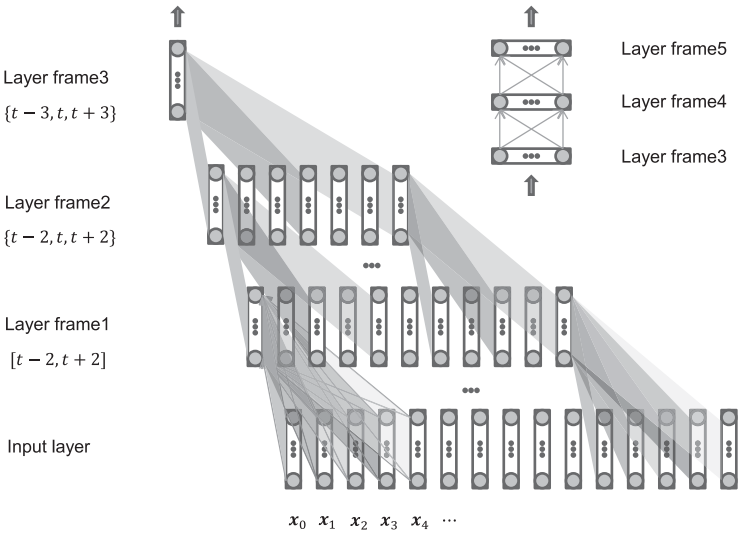


Figure 5.6 Architecture of the TDNN in the x-vector extractor. At the input layer, the vertical bars represent acoustic frames. In the hidden layers, the vertical bars comprises neurons with connections to contextual nodes in the layer below. (Courtesy of Youzhi Tu)

be propagated to the PLDA models [50, 80], the method increases the scoring time significantly [82, 83].

In the conventional i-vector/PLDA framework, speaker identity is represented by the a point estimate of the posterior density (Eq. 3.76) and the uncertainty of the point estimate is discarded. Meta-embedding overcomes this limitation by considering the likelihood function for the speaker identity variable:

$$f(\mathbf{z}) \propto P(r|\mathbf{z}), \tag{5.1}$$

where $\mathbf{z} \in \mathbb{R}^d$ is the speaker identity variable (which is hidden) and r is a representation of the audio recording, e.g., i-vector or MFCCs. Therefore, given the jth recording, instead of using the point estimate r_j to represent the speaker, we keep all of the information about the speaker in $f_j(\mathbf{z}) = k_j P(r_j|\mathbf{z})$ where k_j is an arbitrary constant. It is important to note that meta-embedding is the whole function f_j instead of some point estimates that live in $\mathbb{R}^d$.

Given the representation of two recordings, r_1 and r_2, we have two hypotheses:

$$H_1 : r_1 \text{ and } r_2 \text{ are obtained from the same speaker}$$
$$H_2 : r_1 \text{ and } r_2 \text{ are obtained from two different speakers}$$

To apply meta-embedding for speaker verification, we need to compute the likelihood ratio

$$\frac{p(r_1, r_2|H_1)}{p(r_1, r_2|H_2)} = \frac{\int_{\mathbb{R}^d} f_1(\mathbf{z}) f_2(\mathbf{z}) \pi(\mathbf{z}) \mathrm{d}\mathbf{z}}{\left[\int_{\mathbb{R}^d} f_1(\mathbf{z}) \pi(\mathbf{z}) \mathrm{d}\mathbf{z}\right] \left[\int_{\mathbb{R}^d} f_2(\mathbf{z}) \pi(\mathbf{z}) \mathrm{d}\mathbf{z}\right]}, \tag{5.2}$$

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

where $\pi(\mathbf{z})$ is the prior density of $\mathbf{z}$. In [62], $\pi(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$. The integrals in Eq. 5.2 are the expectation of f_j:

$$\int_{\mathbb{R}^d} f_j(\mathbf{z})\pi(\mathbf{z})\mathrm{d}\mathbf{z} = \mathbb{E}_{\mathbf{z}\sim\pi(\mathbf{z})}\{f_j(\mathbf{z})\}.$$

For Gaussian meta-embedding (GME), the expectation is given by [161]

$$\begin{aligned}
\mathbb{E}_{\mathbf{z}\sim\pi(\mathbf{z})}\{f(\mathbf{z})\} &= \int \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}) f(\mathbf{z})\mathrm{d}\mathbf{z} \\
&= \int \frac{1}{\sqrt{(2\pi)^d}} \exp\left\{-\frac{1}{2}\mathbf{z}^\mathsf{T}\mathbf{z}\right\} \cdot \exp\left\{\mathbf{a}^\mathsf{T}\mathbf{z} - \frac{1}{2}\mathbf{z}^\mathsf{T}\mathbf{B}\mathbf{z}\right\} \mathrm{d}\mathbf{z} \\
&= \int \frac{1}{\sqrt{(2\pi)^d}} \exp\left\{\mathbf{a}^\mathsf{T}\mathbf{z} - \frac{1}{2}\mathbf{z}^\mathsf{T}(\mathbf{B}+\mathbf{I})\mathbf{z}\right\} \mathrm{d}\mathbf{z} \\
&= |\mathbf{B}+\mathbf{I}|^{-\frac{1}{2}} \exp\left\{\frac{1}{2}\mathbf{a}^\mathsf{T}(\mathbf{B}+\mathbf{I})^{-1}\mathbf{a}\right\},
\end{aligned} \tag{5.3}$$

where $\mathbf{a}$ and $\mathbf{B}$ are the natural parameters of the Gaussian likelihood function $f_j(\mathbf{z})$.

5.3 Robust PLDA

One school of thought to enhance the robustness of speaker verification systems against background noise is to improve the robustness of the PLDA model. To this end, both the SNR and duration variabilities can be incorporated into the PLDA model so that these variabilities can be marginalized during the scoring stage. The method is called SNR- and duration-invariant PLDA [55, 162, 163]. Table 5.1 summarizes the nomenclature of various types of PLDA models to be discussed in this section.

5.3.1 SNR-Invariant PLDA

An i-vector $\mathbf{x}_{ij}$ is generated from a linear model of the form [164, 165]:

$$\mathbf{x}_{ij} = \mathbf{m} + \mathbf{V}\mathbf{h}_i + \mathbf{G}\mathbf{r}_{ij} + \boldsymbol{\epsilon}_{ij}, \tag{5.4}$$

Table 5.1 Abbreviations of various PLDA models.

Abbr.	Model Name	Formula
PLDA	Probabilistic LDA	$\mathbf{x}_{ij} = \mathbf{m} + \mathbf{V}\mathbf{h}_i + \boldsymbol{\epsilon}_{ij}$ (Eq. 5.5)
SI-PLDA	SNR-invariant PLDA	$\mathbf{x}_{ij}^k = \mathbf{m} + \mathbf{V}\mathbf{h}_i + \mathbf{U}\mathbf{w}_k + \boldsymbol{\epsilon}_{ij}^k$ (Eq. 5.6)
DI-PLDA	Duration-invariant PLDA	$\mathbf{x}_{ij}^p = \mathbf{m} + \mathbf{V}\mathbf{h}_i + \mathbf{R}\mathbf{y}_p + \boldsymbol{\epsilon}_{ij}^p$ (Eq. 5.7)
SDI-PLDA	SNR- and duration-invariant PLDA	$\mathbf{x}_{ij}^{kp} = \mathbf{m} + \mathbf{V}\mathbf{h}_i + \mathbf{U}\mathbf{w}_k + \mathbf{R}\mathbf{y}_p + \boldsymbol{\epsilon}_{ij}^{kp}$ (Eq. 5.29)

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

where $\mathbf{V}$ and $\mathbf{G}$ define the speaker subspace and channel subspace, respectively. $\mathbf{h}_i$ and $\mathbf{r}_{ij}$ are the speaker and channel factors, respectively, and $\boldsymbol{\epsilon}_{ij}$ is the residue with a Gaussian distribution, $\mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \boldsymbol{\Sigma})$. In most cases, $\boldsymbol{\Sigma}$ is a diagonal covariance matrix that represents the variation that cannot be described by $\mathbf{G}\mathbf{G}^{\mathsf{T}}$ and $\mathbf{V}\mathbf{V}^{\mathsf{T}}$.

The PLDA model in Eq. 5.4 has two components [61, 66], namely the speaker component ($\mathbf{m}+\mathbf{V}\mathbf{h}_i$) that represents the characteristics of speaker i and the channel component ($\mathbf{G}\mathbf{r}_{ij}+\boldsymbol{\epsilon}_{ij}$) that characterizes the speaker as well as the channel. Because the dimension of i-vectors is sufficiently low, the covariance $\mathbf{G}\mathbf{G}^{\mathsf{T}}$ can be absorbed into $\boldsymbol{\Sigma}$ provided that it is a full covariance matrix. As a result, the PLDA model reduces to [166]:

$$\mathbf{x}_{ij} = \mathbf{m} + \mathbf{V}\mathbf{h}_i + \boldsymbol{\epsilon}_{ij}, \tag{5.5}$$

where $\boldsymbol{\epsilon}_{ij} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma}$ being a full covariance matrix.

SNR-invariant PLDA (SI-PLDA) was developed [55, 162] to address the limitation of PLDA in modeling SNR and duration variabilities in i-vectors. To train an SNR-invariant model, we partition the training set into K groups based on the SNR of the training utterances so that each i-vector is assigned to one SNR group. Let $\mathbf{x}_{ij}^k$ represents the jth i-vector from the ith speaker in the kth SNR group. Extending the classical PLDA to modeling SNR and duration variabilities, we may write $\mathbf{x}_{ij}^k$ as

$$\mathbf{x}_{ij}^k = \mathbf{m} + \mathbf{V}\mathbf{h}_i + \mathbf{U}\mathbf{w}_k + \boldsymbol{\epsilon}_{ij}^k, \tag{5.6}$$

where $\mathbf{V}$ and $\mathbf{U}$ represent the speaker and SNR subspaces respectively, $\mathbf{h}_i$ and $\mathbf{w}_k$ are speaker and SNR factors with a standard normal prior, and $\boldsymbol{\epsilon}_{ij}^k$ is a residue term following a Gaussian distribution $\mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \boldsymbol{\Sigma})$. In [55, 162], the channel variability is modeled by the full covariance matrix $\boldsymbol{\Sigma}$.

The SI-PLDA (Eq. 5.6) and the classical PLDA (Eq. 5.4) are different. Specifically, in SI-PLDA, variability in i-vectors caused by noise level variation is modeled by $\mathbf{U}\mathbf{U}^{\mathsf{T}}$, whereas in classical PLDA, the channel variability is modeled by $\mathbf{G}\mathbf{G}^{\mathsf{T}}$. Therefore, the SNR factor ($\mathbf{w}_k$ in Eq. 5.6) depends on the SNR groups, whereas the channel factor ($\mathbf{r}_{ij}$ in Eq. 5.4) depends on the speaker and channel.

Figure 5.7 shows the use of speaker and SNR labels when training a PLDA model (a) and an SNR-invariant PLDA model (b). In PLDA, the SNR group labels are ignored during PLDA training, whereas, in SI-PLDA, the SNR labels are used to group the i-vectors according to the SNR of their respective utterances. These extra SNR labels enable the PLDA model to find an SNR subspace, which helps to reduce i-vector variability due to noise level variation.

5.3.2 Duration-Invariant PLDA

In some datasets, the duration of utterances varies widely. Figure 5.8 shows the histogram of utterance durations in NIST 2016 SRE. According to [167, 168], duration variability in the i-vectors can be modeled as additive noise in i-vector space. Adding a duration factor to the PLDA model results in a duration-invariant PLDA (DI-PLDA) [163].

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

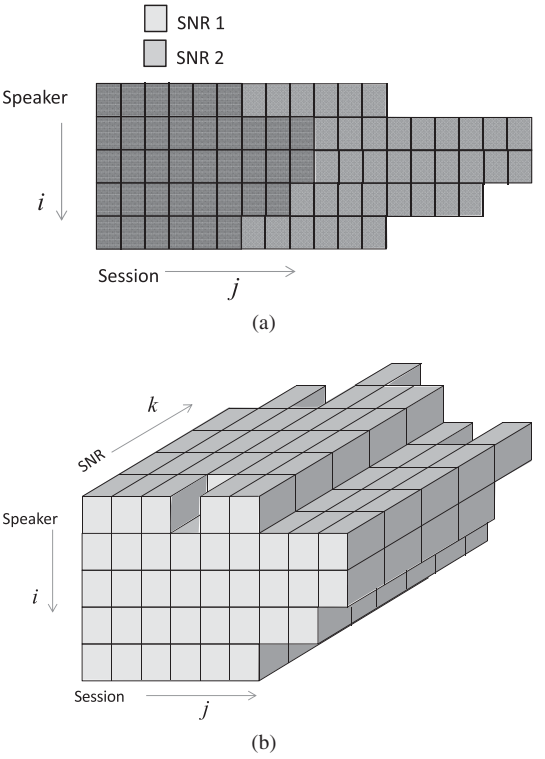


Figure 5.7 (a) Grouping of i-vectors under multi-condition training in conventional PLDA. The rectangles represents i-vectors. Although the training set contains two groups of i-vectors (with different SNRs), the training algorithm of PLDA will not consider this information and will sum over the statistics of both groups. (b) Grouping of i-vectors under SNR-invariant PLDA in which each cube represents an i-vector. There are $H_i(k)$ i-vectors from speaker i whose utterances fall into SNR group k. The training algorithm of SNR-invariant PLDA will take the SNR labels and speaker labels into consideration and will sum over the statistics within individual groups. [Reprinted from *Discriminative Subspace Modeling of SNR and Duration Variabilities for Robust Speaker Verification (Figure 1), N. Li, M.W. Mak, W.W. Lin and J.T. Chien, Computer Speech and Language, vol. 45, pp. 87–103, 2017, with permission of Elsevier*]

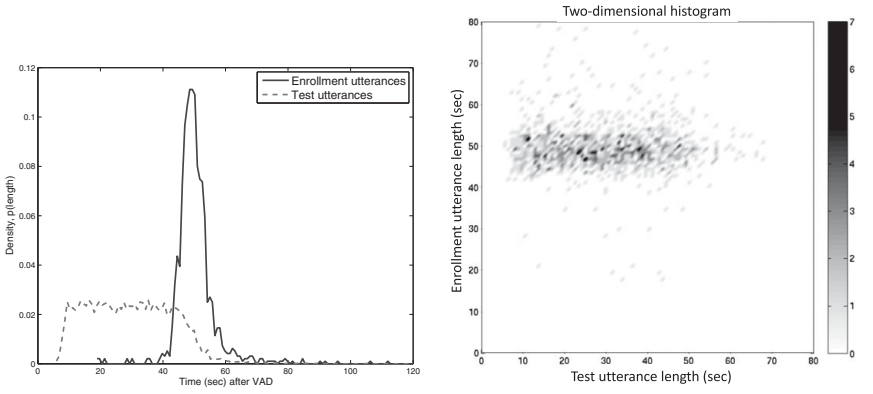
EM Formulations

Denote $\mathcal{X} = \{\mathbf{x}_{ij}^p | i = 1, \ldots, S; j = 1, \ldots, H_i(p); p = 1, \ldots, P\}$ as a set of i-vectors obtained from S speakers, where $\mathbf{x}_{ij}^p$ is utterance j from speaker i at duration group p. Speaker i posseses $H_i(p)$ i-vectors from duration group p. If the term associated with the SNR is replaced by a term related to duration, Eq. 5.6 becomes DI-PLDA, i.e.,

$$\mathbf{x}_{ij}^p = \mathbf{m} + \mathbf{V}\mathbf{h}_i + \mathbf{R}\mathbf{y}_p + \boldsymbol{\epsilon}_{ij}^p, \tag{5.7}$$

where the duration subspace is defined by $\mathbf{R}$ and $\mathbf{y}_p$ denotes the duration factor whose prior is a standard Gaussian. The meanings of other terms remain the same as those in Eq. 5.6.

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press



(a) (b)

Figure 5.8 (a) Duration distributions of enrollment and test utterances in NIST 2016 SRE. (b) A two-dimensional histogram illustrating the duration distribution of all possible target-test pairs.

To simplify computation, the latent factors $\mathbf{h}_i$ and $\mathbf{y}_p$ are assumed to be posteriorly independent in [55]. However, a better approach is to consider them posteriorly dependent, as in [163]. In the latter case, variational Bayes methods [1] can be used for deriving the EM algorithms for training the SI-PLDA and DI-PLDA models.

Suppose there are $N_i = \sum_{p=1}^{P} H_i(p)$ training utterances from speaker i and $B_p = \sum_{i=1}^{S} H_i(p)$ training utterances in duration group p. Denote $\boldsymbol{\theta} = \{\mathbf{m}, \mathbf{V}, \mathbf{R}, \boldsymbol{\Sigma}\}$ as the old estimate of the model parameters, a new estimate $\boldsymbol{\theta}'$ can be computed by maximizing the auxiliary function:

$$\begin{aligned} Q(\boldsymbol{\theta}'|\boldsymbol{\theta}) &= \mathbb{E}_{q(\underline{\mathbf{h}},\underline{\mathbf{y}})}\left\{\log p(\mathcal{X},\underline{\mathbf{h}},\underline{\mathbf{y}}|\boldsymbol{\theta}')\middle|\mathcal{X},\boldsymbol{\theta}\right\} \\ &= \mathbb{E}_{q(\underline{\mathbf{h}},\underline{\mathbf{y}})}\left\{\sum\nolimits_{ijp}\log\left[p(\mathbf{x}_{ij}^{p}|\mathbf{h}_i,\mathbf{y}_p,\boldsymbol{\theta}')p(\mathbf{h}_i,\mathbf{y}_p)\right]\middle|\mathcal{X},\boldsymbol{\theta}\right\}, \end{aligned} \tag{5.8}$$

where $\underline{\mathbf{h}} = \{\mathbf{h}_1, \ldots, \mathbf{h}_s\}$, $\underline{\mathbf{y}} = \{\mathbf{y}_1, \ldots, \mathbf{y}_p\}$, and $q(\underline{\mathbf{h}}, \underline{\mathbf{y}})$ is the variational posterior density of $\underline{\mathbf{h}}$ and $\underline{\mathbf{y}}$. $Q(\boldsymbol{\theta}'|\boldsymbol{\theta})$ can then be maximized by setting the derivative of $Q(\boldsymbol{\theta}'|\boldsymbol{\theta})$ with respect to the model parameters $\{\mathbf{m}, \mathbf{V}, \mathbf{R}, \boldsymbol{\Sigma}\}$ to $\mathbf{0}$, which results in

$$\mathbf{m} = \frac{1}{N}\sum_{i=1}^{S}\sum_{p=1}^{P}\sum_{j=1}^{H_i(p)}\mathbf{x}_{ij}^{p}, \tag{5.9a}$$

$$\mathbf{V}' = \left\{\sum_{i=1}^{S}\sum_{p=1}^{P}\sum_{j=1}^{H_i(p)}\left[(\mathbf{x}_{ij}^{p}-\mathbf{m})\langle\mathbf{h}_i|\mathcal{X}\rangle - \mathbf{R}\langle\mathbf{y}_p\mathbf{h}_i^{\mathsf{T}}|\mathcal{X}\rangle\right]\right\} \left[\sum_{i=1}^{S}\sum_{p=1}^{P}\sum_{j=1}^{H_i(p)}\langle\mathbf{h}_i\mathbf{h}_i^{\mathsf{T}}|\mathcal{X}\rangle\right]^{-1}, \tag{5.9b}$$

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

$$\mathbf{R}' = \left\{ \sum_{i=1}^{S} \sum_{p=1}^{P} \sum_{j=1}^{H_i(p)} \left[(\mathbf{x}_{ij}^p - \mathbf{m}) \langle \mathbf{y}_p | \mathcal{X} \rangle - \mathbf{V} \langle \mathbf{h}_i \mathbf{y}_p^{\mathsf{T}} | \mathcal{X} \rangle \right] \right\}, \left[\sum_{i=1}^{S} \sum_{p=1}^{P} \sum_{j=1}^{H_i(p)} \langle \mathbf{y}_p \mathbf{y}_p^{\mathsf{T}} | \mathcal{X} \rangle \right]^{-1} \tag{5.9c}$$

$$\boldsymbol{\Sigma}' = \frac{1}{N} \sum_{i=1}^{S} \sum_{p=1}^{P} \sum_{j=1}^{H_i(p)} \left[(\mathbf{x}_{ij}^p - \mathbf{m})(\mathbf{x}_{ij}^p - \mathbf{m})^{\mathsf{T}} - \mathbf{V} \langle \mathbf{h}_i | \mathcal{X} \rangle (\mathbf{x}_{ij}^p - \mathbf{m})^{\mathsf{T}} \right] - \frac{1}{N} \sum_{i=1}^{S} \sum_{p=1}^{P} \sum_{j=1}^{H_i(p)} \mathbf{R} \langle \mathbf{y}_p | \mathcal{X} \rangle (\mathbf{x}_{ij}^p - \mathbf{m})^{\mathsf{T}}, \tag{5.9d}$$

where $N = \sum_{i=1}^{S} N_i = \sum_{p=1}^{P} B_p$.

The global mean, Eq. 5.9(a), can be computed in one iteration. However, the other model parameters, Eqs. 5.9(b)–(d), which make up the M-step of the EM algorithm, require iterative update because the posterior distribution of $\mathbf{h}_i$ and $\mathbf{y}_p$ are not known. Below, we will explain how variational Bayes method can be used to compute these posteriors.

In variational Bayes, the true posterior $p(\underline{\mathbf{h}}, \underline{\mathbf{y}} | \mathcal{X})$ is approximated by a variational posterior $q(\underline{\mathbf{h}}, \underline{\mathbf{y}})$, and the marginal likelihood of $\mathcal{X}$ is written as

$$\begin{aligned} \log p(\mathcal{X}) &= \int \int q(\underline{\mathbf{h}}, \underline{\mathbf{y}}) \log p(\mathcal{X}) \mathrm{d}\underline{\mathbf{h}} \mathrm{d}\underline{\mathbf{y}} \\ &= \int \int q(\underline{\mathbf{h}}, \underline{\mathbf{y}}) \log \left[\frac{p(\underline{\mathbf{h}}, \underline{\mathbf{y}}, \mathcal{X})}{p(\underline{\mathbf{h}}, \underline{\mathbf{y}} | \mathcal{X})} \right] \mathrm{d}\underline{\mathbf{h}} \mathrm{d}\underline{\mathbf{y}} \\ &= \int \int q(\underline{\mathbf{h}}, \underline{\mathbf{y}}) \log \left[\frac{p(\underline{\mathbf{h}}, \underline{\mathbf{y}}, \mathcal{X})}{q(\underline{\mathbf{h}}, \underline{\mathbf{y}})} \right] \mathrm{d}\underline{\mathbf{h}} \mathrm{d}\underline{\mathbf{y}} + \int \int q(\underline{\mathbf{h}}, \underline{\mathbf{y}}) \log \left[\frac{q(\underline{\mathbf{h}}, \underline{\mathbf{y}})}{p(\underline{\mathbf{h}}, \underline{\mathbf{y}} | \mathcal{X})} \right] \mathrm{d}\underline{\mathbf{h}} \mathrm{d}\underline{\mathbf{y}} \\ &= \mathcal{L}(q) + \mathcal{D}_{\mathrm{KL}}(q(\underline{\mathbf{h}}, \underline{\mathbf{y}}) \| p(\underline{\mathbf{h}}, \underline{\mathbf{y}} | \mathcal{X})). \end{aligned} \tag{5.10}$$

In Eq. 5.10, $\mathcal{D}_{\mathrm{KL}}(q||p)$ denotes the KL-divergence between distribution q and distribution p; also,

$$\mathcal{L}(q) = \int \int q(\underline{\mathbf{h}}, \underline{\mathbf{y}}) \log \left[\frac{p(\underline{\mathbf{h}}, \underline{\mathbf{y}}, \mathcal{X})}{q(\underline{\mathbf{h}}, \underline{\mathbf{y}})} \right] \mathrm{d}\underline{\mathbf{h}} \mathrm{d}\underline{\mathbf{y}} \tag{5.11}$$

is the variational lower-bound of the marginal likelihood. Because of the non-negativity of KL-divergence, maximizing the lower bound with respect to $q(\underline{\mathbf{h}})$ will also maximize the marginal likelihood. The maximum is attained when $q(\underline{\mathbf{h}}, \underline{\mathbf{y}})$ is the same as the true posterior $p(\underline{\mathbf{h}}, \underline{\mathbf{y}} | \mathcal{X})$. Also, it is assumed that the approximated posterior $q(\underline{\mathbf{h}}, \underline{\mathbf{y}})$ is factorizable, i.e.,

$$\log q(\underline{\mathbf{h}}, \underline{\mathbf{y}}) = \log q(\underline{\mathbf{h}}) + \log q(\underline{\mathbf{y}}) = \sum_{i=1}^{S} \log q(\mathbf{h}_i) + \sum_{p=1}^{P} \log q(\mathbf{y}_p). \tag{5.12}$$

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

When the lower bound $\mathcal{L}(q)$ in Eq. 5.11 is maximized, we have [1, 30]

$$\begin{aligned}\log q(\underline{\mathbf{h}}) &= \mathbb{E}_{q(\underline{\mathbf{y}})}\{\log p(\underline{\mathbf{h}}, \underline{\mathbf{y}}, \mathcal{X})\} + \text{const} \\ \log q(\underline{\mathbf{y}}) &= \mathbb{E}_{q(\underline{\mathbf{h}})}\{\log p(\underline{\mathbf{h}}, \underline{\mathbf{y}}, \mathcal{X})\} + \text{const},\end{aligned} \tag{5.13}$$

where $\mathbb{E}_{q(\underline{\mathbf{y}})}\{Z\}$ is the expectation of Z using $q(\underline{\mathbf{y}})$ as the density.

Note that $\log q(\underline{\mathbf{h}})$ in Eq. 5.13 can be written as

$$\begin{aligned}&\log q(\underline{\mathbf{h}}) \\ &= \sum\nolimits_i \log q(\mathbf{h}_i) \\ &= \langle \log p(\underline{\mathbf{h}}, \underline{\mathbf{y}}, \mathcal{X}) \rangle_{\underline{\mathbf{y}}} + \text{const} \\ &= \langle \log p(\mathcal{X}|\underline{\mathbf{h}}, \underline{\mathbf{y}}) \rangle_{\underline{\mathbf{y}}} + \langle \log p(\underline{\mathbf{h}}, \underline{\mathbf{y}}) \rangle_{\underline{\mathbf{y}}} + \text{const} \\ &= \sum\nolimits_{ijp} \left\langle \log \mathcal{N}(\mathbf{x}_{ij}^p | \mathbf{m} + \mathbf{V}\mathbf{h}_i + \mathbf{R}\mathbf{y}_p, \boldsymbol{\Sigma}) \right\rangle_{\mathbf{y}_p} + \sum\nolimits_i \langle \log \mathcal{N}(\mathbf{h}_i | \mathbf{0}, \mathbf{I}) \rangle_{\underline{\mathbf{y}}} \\ &\quad + \sum\nolimits_p \langle \log \mathcal{N}(\mathbf{y}_p | \mathbf{0}, \mathbf{I}) \rangle_{\mathbf{y}_p} + \text{const} \\ &= -\frac{1}{2} \sum\nolimits_{ijp} (\mathbf{x}_{ij}^p - \mathbf{m} - \mathbf{V}\mathbf{h}_i - \mathbf{R}\mathbf{y}_p^*)^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}_{ij}^p - \mathbf{m} - \mathbf{V}\mathbf{h}_i - \mathbf{R}\mathbf{y}_p^*) \\ &\quad - \frac{1}{2} \sum\nolimits_i \mathbf{h}_i^{\mathsf{T}} \mathbf{h}_i + \text{const} \\ &= \sum\nolimits_i \left[\mathbf{h}_i^{\mathsf{T}} \mathbf{V}^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \sum\nolimits_{jp} (\mathbf{x}_{ij}^p - \mathbf{m} - \mathbf{R}\mathbf{y}_p^*) - \frac{1}{2} \mathbf{h}_i^{\mathsf{T}} \left(\mathbf{I} + \sum\nolimits_p H_i(p) \mathbf{V}^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \mathbf{V} \right) \mathbf{h}_i^{\mathsf{T}} \right] \\ &\quad + \text{const},\end{aligned} \tag{5.14}$$

where $\mathbf{y}_p^* \equiv \langle \mathbf{y}_p | \mathcal{X} \rangle_{\mathbf{y}_p}$ is the posterior mean of $\mathbf{y}_p$ in the previous iteration and $\langle . \rangle_{\mathbf{y}_p}$ denotes the expectation with respect to $\mathbf{y}_p$. Note that $\langle \log \mathcal{N}(\mathbf{y}_p | \mathbf{0}, \mathbf{I}) \rangle_{\mathbf{y}_p}$ is the differential entropy of a normal distribution, which does not depend on $\mathbf{h}_i$ [169, Chapter 8].

By extracting $\mathbf{h}_i$ in Eq. 5.14 and comparing with $\sum_i \log q(\mathbf{h}_i)$, we can see that $q(\mathbf{h}_i)$ is a Gaussian with mean vector and precision matrix as follows:

$$\mathbb{E}_{q(\mathbf{h}_i)}\{\mathbf{h}_i | \mathcal{X}\} = \langle \mathbf{h}_i | \mathcal{X} \rangle = \left(\mathbf{L}_i^{(1)} \right)^{-1} \mathbf{V}^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \sum_{p=1}^{P} \sum_{j=1}^{H_i(p)} (\mathbf{x}_{ij}^p - \mathbf{m} - \mathbf{R}\mathbf{y}_p^*)$$

and

$$\mathbf{L}_i^{(1)} \equiv \mathbf{I} + \sum_{p=1}^{P} H_i(p) \mathbf{V}^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \mathbf{V}.$$

Using this precision matrix, we can compute the second-order moment (which will be needed in the M-step) as follows:

$$\langle \mathbf{h}_i \mathbf{h}_i^{\mathsf{T}} | \mathcal{X} \rangle = \left(\mathbf{L}_i^{(1)} \right)^{-1} + \langle \mathbf{h}_i | \mathcal{X} \rangle \langle \mathbf{h}_i | \mathcal{X} \rangle^{\mathsf{T}}.$$

In the same manner, we obtain the posterior mean and second-order moment of $\mathbf{y}_p$ by comparing the terms in $\log q(\mathbf{y}_p)$ with a Gaussian distribution.

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

The posterior moment $\langle \mathbf{h}_i \mathbf{y}_p^\mathsf{T} | \mathcal{X} \rangle$ is also needed in the M-step. We can use variational Bayes to approximate it as follows:

$$p(\mathbf{h}_i, \mathbf{y}_p | \mathcal{X}) \approx q(\mathbf{h}_i) q(\mathbf{y}_p), \tag{5.15}$$

where both $q(\mathbf{h}_i)$ and $q(\mathbf{y}_p)$ are Gaussians. According to the law of total expectation [170], we factorize Eq. 5.15 to get

$$\begin{aligned} \langle \mathbf{y}_p \mathbf{h}_i^\mathsf{T} | \mathcal{X} \rangle &\approx \langle \mathbf{y}_p | \mathcal{X} \rangle \langle \mathbf{h}_i | \mathcal{X} \rangle^\mathsf{T} \\ \langle \mathbf{h}_i \mathbf{y}_p^\mathsf{T} | \mathcal{X} \rangle &\approx \langle \mathbf{h}_i | \mathcal{X} \rangle \langle \mathbf{y}_p | \mathcal{X} \rangle^\mathsf{T}. \end{aligned}$$

Therefore, the equations for the variational E-step are as follows:

$$\mathbf{L}_i^{(1)} = \mathbf{I} + N_i \mathbf{V}^\mathsf{T} \boldsymbol{\Sigma}^{-1} \mathbf{V} \quad i = 1, \ldots, S \tag{5.16a}$$

$$\mathbf{L}_p^{(2)} = \mathbf{I} + B_p \mathbf{R}^\mathsf{T} \boldsymbol{\Sigma}^{-1} \mathbf{R} \quad p = 1, \ldots, P \tag{5.16b}$$

$$\langle \mathbf{h}_i | \mathcal{X} \rangle = (\mathbf{L}_i^{(1)})^{-1} \mathbf{V}^\mathsf{T} \boldsymbol{\Sigma}^{-1} \sum_{p=1}^{P} \sum_{j=1}^{H_i(p)} (\mathbf{x}_{ij}^p - \mathbf{m} - \mathbf{R}\mathbf{y}_p^*) \tag{5.16c}$$

$$\langle \mathbf{y}_p | \mathcal{X} \rangle = (\mathbf{L}_p^{(2)})^{-1} \mathbf{R}^\mathsf{T} \boldsymbol{\Sigma}^{-1} \sum_{i=1}^{S} \sum_{j=1}^{H_i(p)} (\mathbf{x}_{ij}^p - \mathbf{m} - \mathbf{V}\mathbf{h}_i^*) \tag{5.16d}$$

$$\langle \mathbf{h}_i \mathbf{h}_i^\mathsf{T} | \mathcal{X} \rangle = (\mathbf{L}_i^{(1)})^{-1} + \langle \mathbf{h}_i | \mathcal{X} \rangle \langle \mathbf{h}_i | \mathcal{X} \rangle^\mathsf{T} \tag{5.16e}$$

$$\langle \mathbf{y}_p \mathbf{y}_p^\mathsf{T} | \mathcal{X} \rangle = (\mathbf{L}_p^{(2)})^{-1} + \langle \mathbf{y}_p | \mathcal{X} \rangle \langle \mathbf{y}_p | \mathcal{X} \rangle^\mathsf{T} \tag{5.16f}$$

$$\langle \mathbf{y}_p \mathbf{h}_i^\mathsf{T} | \mathcal{X} \rangle \approx \langle \mathbf{y}_p | \mathcal{X} \rangle \langle \mathbf{h}_i | \mathcal{X} \rangle^\mathsf{T} \tag{5.16g}$$

$$\langle \mathbf{h}_i \mathbf{y}_p^\mathsf{T} | \mathcal{X} \rangle \approx \langle \mathbf{h}_i | \mathcal{X} \rangle \langle \mathbf{y}_p | \mathcal{X} \rangle^\mathsf{T} \tag{5.16h}$$

Algorithm 4 shows the procedure of training a duration-invariant PLDA model.

Derivation of VB Posteriors

Our goal is to approximate the true joint posteriors $p(\underline{\mathbf{h}}, \underline{\mathbf{w}} | \mathcal{X})$ by a distribution $q(\underline{\mathbf{h}}, \underline{\mathbf{w}})$ such that

$$q(\underline{\mathbf{h}}, \underline{\mathbf{w}}) = q(\underline{\mathbf{h}}) q(\underline{\mathbf{w}}).$$

The derivation of the VB posteriors begins with minimizing the KL-divergence

$$\begin{aligned} \mathcal{D}_{\mathrm{KL}}(q(\underline{\mathbf{h}}, \underline{\mathbf{w}}) || p(\underline{\mathbf{h}}, \underline{\mathbf{w}} | \mathcal{X})) &= \int \int q(\underline{\mathbf{h}}, \underline{\mathbf{w}}) \log \left[\frac{q(\underline{\mathbf{h}}, \underline{\mathbf{w}})}{p(\underline{\mathbf{h}}, \underline{\mathbf{w}} | \mathcal{X})} \right] \mathrm{d}\underline{\mathbf{h}} \mathrm{d}\underline{\mathbf{w}} \\ &= -\int \int q(\underline{\mathbf{h}}, \underline{\mathbf{w}}) \log \left[\frac{p(\underline{\mathbf{h}}, \underline{\mathbf{w}} | \mathcal{X})}{q(\underline{\mathbf{h}}, \underline{\mathbf{w}})} \right] \mathrm{d}\underline{\mathbf{h}} \mathrm{d}\underline{\mathbf{w}} \\ &= -\int \int q(\underline{\mathbf{h}}, \underline{\mathbf{w}}) \log \left[\frac{p(\underline{\mathbf{h}}, \underline{\mathbf{w}}, \mathcal{X})}{q(\underline{\mathbf{h}}, \underline{\mathbf{w}}) p(\mathcal{X})} \right] \mathrm{d}\underline{\mathbf{h}} \mathrm{d}\underline{\mathbf{w}} \\ &= -\mathcal{L}(q) + \log p(\mathcal{X}), \end{aligned} \tag{5.17}$$

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

Algorithm 4 Variational Bayes EM algorithm for duration-invariant PLDA

Input:

Development data set consisting of i-vectors $\mathcal{X} = \{\mathbf{x}_{ij}^p | i = 1, \ldots, S; j = 1, \ldots, H_i(p); p = 1, \ldots, P\}$, with identity labels and duration group labels.

Initialization:

$\mathbf{y}_p^* \leftarrow \mathbf{0}$;
$\boldsymbol{\Sigma} \leftarrow 0.01\mathbf{I}$;
$\mathbf{V}, \mathbf{R} \leftarrow$ eigenvectors of PCA projection matrix learned using data set $\mathcal{X}$;

Parameter Estimation:

Compute $\mathbf{m}$ via Eq. 5.9(a);
Compute $\mathbf{L}_i^{(1)}$ and $\mathbf{L}_p^{(2)}$ according to Eqs. 5.16(a) and (b), respectively;
Set $\mathbf{y}_p^*$ to the posterior mean of $\mathbf{y}_p$. Compute the posterior mean of $\mathbf{h}_i$ using Eq. 5.16(c);
Use the posterior mean of $\mathbf{h}_i$ computed in Step 3 to update the posterior mean of $\mathbf{y}_p$ according to Eq. 5.16(d);
Compute the other terms in the E-step, i.e., Eq. 5.16(e)–(h);
Update the model parameters using Eq. 5.9(a)–(c);
Go to Step 2 until convergence;

Return: the parameters of the duration-invariant PLDA model $\boldsymbol{\theta} = \{\mathbf{m}, \mathbf{V}, \mathbf{R}, \boldsymbol{\Sigma}\}$.

where $\mathcal{L}(q)$ is the variational Bayes lower bound (VBLB). The lower bound is given by:

$$\begin{aligned}
\mathcal{L}(q) &= \int\int q(\underline{\mathbf{h}}, \underline{\mathbf{w}}) \log \left[\frac{p(\underline{\mathbf{h}}, \underline{\mathbf{w}}, \mathcal{X})}{q(\underline{\mathbf{h}}, \underline{\mathbf{w}})} \right] \mathrm{d}\underline{\mathbf{h}}\mathrm{d}\underline{\mathbf{w}} \\
&= \int\int q(\underline{\mathbf{h}}, \underline{\mathbf{w}}) \log p(\underline{\mathbf{h}}, \underline{\mathbf{w}}, \mathcal{X}) \mathrm{d}\underline{\mathbf{h}}\mathrm{d}\underline{\mathbf{w}} - \int\int q(\underline{\mathbf{h}}, \underline{\mathbf{w}}) \log q(\underline{\mathbf{h}}, \underline{\mathbf{w}}) \mathrm{d}\underline{\mathbf{h}}\mathrm{d}\underline{\mathbf{w}} \\
&= \int\int q(\underline{\mathbf{h}}) q(\underline{\mathbf{w}}) \log p(\underline{\mathbf{h}}, \underline{\mathbf{w}}, \mathcal{X}) \mathrm{d}\underline{\mathbf{h}}\mathrm{d}\underline{\mathbf{w}} - \int\int q(\underline{\mathbf{h}}) q(\underline{\mathbf{w}}) \log q(\underline{\mathbf{h}}) \mathrm{d}\underline{\mathbf{h}}\mathrm{d}\underline{\mathbf{w}} \\
&\quad - \int\int q(\underline{\mathbf{h}}) q(\underline{\mathbf{w}}) \log q(\underline{\mathbf{w}}) \mathrm{d}\underline{\mathbf{h}}\mathrm{d}\underline{\mathbf{w}} \\
&= \int\int q(\underline{\mathbf{h}}) q(\underline{\mathbf{w}}) \log p(\underline{\mathbf{h}}, \underline{\mathbf{w}}, \mathcal{X}) \mathrm{d}\underline{\mathbf{h}}\mathrm{d}\underline{\mathbf{w}} - \int q(\underline{\mathbf{h}}) \log q(\underline{\mathbf{h}}) \mathrm{d}\underline{\mathbf{h}} \\
&\quad - \int q(\underline{\mathbf{w}}) \log q(\underline{\mathbf{w}}) \mathrm{d}\underline{\mathbf{w}}.
\end{aligned} \tag{5.18}$$

Note that the first term in Eq. 6.39 can be written as

$$\begin{aligned}
\int\int q(\underline{\mathbf{h}}) q(\underline{\mathbf{w}}) \log p(\underline{\mathbf{h}}, \underline{\mathbf{w}}, \mathcal{X}) \mathrm{d}\underline{\mathbf{h}}\mathrm{d}\underline{\mathbf{w}} &= \int q(\underline{\mathbf{h}}) \left[\int q(\underline{\mathbf{w}}) \log p(\underline{\mathbf{h}}, \underline{\mathbf{w}}, \mathcal{X}) \mathrm{d}\underline{\mathbf{w}} \right] \mathrm{d}\underline{\mathbf{h}} \\
&= \int q(\underline{\mathbf{h}}) \mathbb{E}_{q(\underline{\mathbf{w}})} \{\log p(\underline{\mathbf{h}}, \underline{\mathbf{w}}, \mathcal{X})\} \mathrm{d}\underline{\mathbf{h}}.
\end{aligned} \tag{5.19}$$

Define a distribution of $\underline{\mathbf{h}}$ as

$$q^*(\underline{\mathbf{h}}) \equiv \frac{1}{Z} \exp\left\{ \mathbb{E}_{q(\underline{\mathbf{w}})} \{\log p(\underline{\mathbf{h}}, \underline{\mathbf{w}}, \mathcal{X})\} \right\}, \tag{5.20}$$

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

where Z is to normalize the distribution. Using Eqs. 5.19 and 5.20, Eq. 6.39 can be written as:

$$\begin{aligned}\mathcal{L}(q) &= \int q(\underline{\mathbf{h}}) \log q^*(\underline{\mathbf{h}}) \mathrm{d}\underline{\mathbf{h}} - \int q(\underline{\mathbf{h}}) \log q(\underline{\mathbf{h}}) \mathrm{d}\underline{\mathbf{h}} - \int q(\underline{\mathbf{w}}) \log q(\underline{\mathbf{w}}) \mathrm{d}\underline{\mathbf{w}} + \log Z \\ &= -\int q(\underline{\mathbf{h}}) \log \left[\frac{q(\underline{\mathbf{h}})}{q^*(\underline{\mathbf{h}})} \right] \mathrm{d}\underline{\mathbf{h}} + \mathcal{H}(q(\underline{\mathbf{w}})) + \log Z \\ &= -\mathcal{D}_{\mathrm{KL}}(q(\underline{\mathbf{h}}) || q^*(\underline{\mathbf{h}})) + \mathcal{H}(q(\underline{\mathbf{w}})) + \log Z.\end{aligned}$$

$\mathcal{L}(q)$ will attain its maximum when the KL-divergence vanishes, i.e.,

$$\log q(\underline{\mathbf{h}}) = \log q^*(\underline{\mathbf{h}}) = \mathbb{E}_{q(\underline{\mathbf{w}})}\{\log p(\underline{\mathbf{h}}, \underline{\mathbf{w}}, \mathcal{X})\} + \text{const.} \tag{5.21}$$

Now, we write Eq. 6.41 as

$$\log p(\mathcal{X}) = \mathcal{L}(q) + \mathcal{D}_{\mathrm{KL}}(q(\underline{\mathbf{h}}, \underline{\mathbf{w}}) || p(\underline{\mathbf{h}}, \underline{\mathbf{w}} | \mathcal{X})).$$

Because KL-divergence is nonnegative, we may maximize the data likelihood $p(\mathcal{X})$ by maximizing the VB lower bound $\mathcal{L}(q)$, which can be achieved by Eq. 5.21. A similar treatment can be applied to $q(\underline{\mathbf{w}})$.

Likelihood Ratio Scores

Suppose we do not know (or not use) the duration of the target and test utterances. Then, the scoring function of SI-PLDA [55] can be used for computing the likelihood ratio score. However, because we know the duration of the the target and test utterances (ℓ_s and ℓ_t), the likelihood-ratio score can be computed as follows:

$$S_{\mathrm{LR}}(\mathbf{x}_s, \mathbf{x}_t | \ell_s, \ell_t) = \log \frac{p(\mathbf{x}_s, \mathbf{x}_t | \text{same-speaker}, \ell_s, \ell_t)}{p(\mathbf{x}_s, \mathbf{x}_t | \text{different-speakers}, \ell_s, \ell_t)}, \tag{5.22}$$

where $\mathbf{x}_s$ and $\mathbf{x}_t$ are the i-vectors of the target speaker and test speaker, respectively.

There are two approaches to calculating the score in Eq. 5.22, depending on the sharpness of the posterior density of $\mathbf{y}_p$. Consider the case where the posterior density of $\mathbf{y}_p$ is sharp at its mean $\mathbf{y}_p^*$.[1] Assuming that ℓ belongs to duration group p, the marginal-likelihood of i-vector $\mathbf{x}$ can then be expressed as:

$$\begin{aligned}p(\mathbf{x} | \ell \in p\text{th duration group}) &= \int_{\mathbf{h}} p(\mathbf{x} | \mathbf{h}, \mathbf{y}_p^*) p(\mathbf{h}) d\mathbf{h} \\ &= \int_{\mathbf{h}} \mathcal{N}(\mathbf{x} | \mathbf{m} + \mathbf{V}\mathbf{h} + \mathbf{R}\mathbf{y}_p^*, \boldsymbol{\Sigma}) \mathcal{N}(\mathbf{h} | \mathbf{0}, \mathbf{I}) d\mathbf{h} \\ &= \mathcal{N}(\mathbf{x} | \mathbf{m} + \mathbf{R}\mathbf{y}_p^*, \mathbf{V}\mathbf{V}^{\mathsf{T}} + \boldsymbol{\Sigma}),\end{aligned} \tag{5.23}$$

where $\mathbf{y}_p^* \equiv \langle \mathbf{y}_p | \mathcal{X} \rangle$ is the posterior mean of $\mathbf{y}_p$. Given a test i-vector $\mathbf{x}_t$ and a target i-vector $\mathbf{x}_s$, we can use Eq. 5.23 to compute the likelihood ratio score:

[1] As suggested by Eq. 5.16(b), this occurs when the number of training i-vectors (B_p) in duration group p is large.

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

$$
\begin{aligned}
S_{\text{LR}}(\mathbf{x}_s, \mathbf{x}_t|\ell_s, \ell_t) &= \log \frac{p(\mathbf{x}_s, \mathbf{x}_t|\text{same-speaker}, \ell_s, \ell_t)}{p(\mathbf{x}_s, \mathbf{x}_t|\text{different-speakers}, \ell_s, \ell_t)} \\
&= \log \frac{\mathcal{N}\left(\begin{bmatrix}\mathbf{x}_s \\ \mathbf{x}_t\end{bmatrix} \middle| \begin{bmatrix}\mathbf{m} + \mathbf{R}\mathbf{y}^*_{p_s} \\ \mathbf{m} + \mathbf{R}\mathbf{y}^*_{p_t}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Psi} & \boldsymbol{\Sigma}_{ac} \\ \boldsymbol{\Sigma}_{ac} & \boldsymbol{\Psi}\end{bmatrix}\right)}{\mathcal{N}\left(\begin{bmatrix}\mathbf{x}_s \\ \mathbf{x}_t\end{bmatrix} \middle| \begin{bmatrix}\mathbf{m} + \mathbf{R}\mathbf{y}^*_{p_s} \\ \mathbf{m} + \mathbf{R}\mathbf{y}^*_{p_t}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Psi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}\end{bmatrix}\right)} \\
&= \frac{1}{2}[\bar{\mathbf{x}}_s^{\mathsf{T}}\mathbf{Q}\bar{\mathbf{x}}_s + 2\bar{\mathbf{x}}_s^{\mathsf{T}}\mathbf{P}\bar{\mathbf{x}}_t + \bar{\mathbf{x}}_t^{\mathsf{T}}\mathbf{Q}\bar{\mathbf{x}}_t] + \text{const}
\end{aligned} \tag{5.24}
$$

where

$$
\begin{aligned}
\bar{\mathbf{x}}_s &= \mathbf{x}_s - \mathbf{m} - \mathbf{R}\mathbf{y}^*_{p_s} \\
\bar{\mathbf{x}}_t &= \mathbf{x}_t - \mathbf{m} - \mathbf{R}\mathbf{y}^*_{p_t} \\
\mathbf{Q} &= \boldsymbol{\Psi}^{-1} - (\boldsymbol{\Psi} - \boldsymbol{\Sigma}_{ac}\boldsymbol{\Psi}^{-1}\boldsymbol{\Sigma}_{ac})^{-1} \\
\mathbf{P} &= \boldsymbol{\Psi}^{-1}\boldsymbol{\Sigma}_{ac}(\boldsymbol{\Psi} - \boldsymbol{\Sigma}_{ac}\boldsymbol{\Psi}^{-1}\boldsymbol{\Sigma}_{ac})^{-1} \\
\boldsymbol{\Psi} &= \mathbf{V}\mathbf{V}^{\mathsf{T}} + \boldsymbol{\Sigma}; \quad \boldsymbol{\Sigma}_{ac} = \mathbf{V}\mathbf{V}^{\mathsf{T}}.
\end{aligned}
$$

Now, let's consider the case where the posterior density of $\mathbf{y}_p$ is *moderately* sharp and is a Gaussian $\mathcal{N}(\mathbf{y}_p|\boldsymbol{\mu}^*_p, \boldsymbol{\Sigma}^*_p)$.[2] If ℓ falls on duration group p, the marginal-likelihood of i-vector $\mathbf{x}$ is:

$$
\begin{aligned}
p(\mathbf{x}|\ell \in p\text{th duration group}) &= \int_{\mathbf{h}}\int_{\mathbf{y}_p} p(\mathbf{x}|\mathbf{h}, \mathbf{y}_p)p(\mathbf{h})p(\mathbf{y}_p)\text{d}\mathbf{h}\text{d}\mathbf{y}_p \\
&= \int_{\mathbf{h}}\int_{\mathbf{y}_p} \mathcal{N}(\mathbf{x}|\mathbf{m} + \mathbf{V}\mathbf{h} + \mathbf{R}\mathbf{y}_p, \boldsymbol{\Sigma})\mathcal{N}(\mathbf{h}|\mathbf{0}, \mathbf{I})\mathcal{N}(\mathbf{y}_p|\mathbf{0}, \mathbf{I})\text{d}\mathbf{h}\text{d}\mathbf{y}_p \\
&= \int_{\mathbf{y}_p} \mathcal{N}(\mathbf{x}|\mathbf{m} + \mathbf{R}\mathbf{y}_p, \mathbf{V}\mathbf{V}^{\mathsf{T}} + \boldsymbol{\Sigma})\mathcal{N}(\mathbf{y}_p|\mathbf{0}, \mathbf{I})\text{d}\mathbf{y}_p \\
&= \mathcal{N}(\mathbf{x}|\mathbf{m} + \mathbf{R}\boldsymbol{\mu}^*_p, \mathbf{V}\mathbf{V}^{\mathsf{T}} + \mathbf{R}\boldsymbol{\Sigma}^*_p\mathbf{R}^{\mathsf{T}} + \boldsymbol{\Sigma}),
\end{aligned} \tag{5.25}
$$

where we may use Eq. 5.16(d) to compute $\boldsymbol{\mu}^*_p$ and Eq. 5.16(b) to estimate $\boldsymbol{\Sigma}^*_p$. Given a test i-vector $\mathbf{x}_t$ and a target i-vector $\mathbf{x}_s$, the likelihood ratio score can be computed as:

$$
S_{\text{LR}}(\mathbf{x}_s, \mathbf{x}_t|\ell_s, \ell_t) = \log \frac{p(\mathbf{x}_s, \mathbf{x}_t|\text{same-speaker}, \ell_s, \ell_t)}{p(\mathbf{x}_s, \mathbf{x}_t|\text{different-speakers}, \ell_s, \ell_t)} \tag{5.26}
$$

$$
= \log \frac{\mathcal{N}\left(\begin{bmatrix}\mathbf{x}_s \\ \mathbf{x}_t\end{bmatrix} \middle| \begin{bmatrix}\mathbf{m} + \mathbf{R}\boldsymbol{\mu}^*_{p_s} \\ \mathbf{m} + \mathbf{R}\boldsymbol{\mu}^*_{p_t}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_s & \boldsymbol{\Sigma}_{ac} \\ \boldsymbol{\Sigma}_{ac} & \boldsymbol{\Sigma}_t\end{bmatrix}\right)}{\mathcal{N}\left(\begin{bmatrix}\mathbf{x}_s \\ \mathbf{x}_t\end{bmatrix} \middle| \begin{bmatrix}\mathbf{m} + \mathbf{R}\boldsymbol{\mu}^*_{p_s} \\ \mathbf{m} + \mathbf{R}\boldsymbol{\mu}^*_{p_t}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_s & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_t\end{bmatrix}\right)} \tag{5.27}
$$

$$
= \frac{1}{2}[\bar{\mathbf{x}}_s^{\mathsf{T}}\mathbf{A}_{s,t}\bar{\mathbf{x}}_s + 2\bar{\mathbf{x}}_s^{\mathsf{T}}\mathbf{B}_{s,t}\bar{\mathbf{x}}_t + \bar{\mathbf{x}}_t^{\mathsf{T}}\mathbf{C}_{s,t}\bar{\mathbf{x}}_t] + \text{const} \tag{5.28}
$$

[2] This happens when there are a moderate number of training i-vectors (B_p) in duration group p, as evident by Eq. 5.16(b).

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

where

$$
\begin{aligned}
\bar{\mathbf{x}}_s &= \mathbf{x}_s - \mathbf{m} - \mathbf{R}\boldsymbol{\mu}^*_{p_s} \\
\bar{\mathbf{x}}_t &= \mathbf{x}_t - \mathbf{m} - \mathbf{R}\boldsymbol{\mu}^*_{p_t} \\
\mathbf{A}_{s,t} &= \boldsymbol{\Sigma}_s^{-1} - (\boldsymbol{\Sigma}_s - \boldsymbol{\Sigma}_{ac}\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\Sigma}_{ac})^{-1} \\
\mathbf{B}_{s,t} &= \boldsymbol{\Sigma}_s^{-1}\boldsymbol{\Sigma}_{ac}(\boldsymbol{\Sigma}_t - \boldsymbol{\Sigma}_{ac}\boldsymbol{\Sigma}_s^{-1}\boldsymbol{\Sigma}_{ac})^{-1} \\
\mathbf{C}_{s,t} &= \boldsymbol{\Sigma}_t^{-1} - (\boldsymbol{\Sigma}_t - \boldsymbol{\Sigma}_{ac}\boldsymbol{\Sigma}_s^{-1}\boldsymbol{\Sigma}_{ac})^{-1} \\
\boldsymbol{\Sigma}_s &= \mathbf{V}\mathbf{V}^{\mathsf{T}} + \mathbf{R}\boldsymbol{\Sigma}^*_{p_s}\mathbf{R}^{\mathsf{T}} + \boldsymbol{\Sigma} \\
\boldsymbol{\Sigma}_t &= \mathbf{V}\mathbf{V}^{\mathsf{T}} + \mathbf{R}\boldsymbol{\Sigma}^*_{p_t}\mathbf{R}^{\mathsf{T}} + \boldsymbol{\Sigma} \\
\boldsymbol{\Sigma}_{ac} &= \mathbf{V}\mathbf{V}^{\mathsf{T}}.
\end{aligned}
$$

5.3.3 SNR- and Duration-Invariant PLDA

The SNR-invariant PLDA in Section 5.3.1 and the duration-invariant PLDA in Section 5.3.2 can be combined to address both SNR and duration variabilities in utterances. It is called SNR- and duration-invariant PLDA (SDI-PLDA) in [163]. The model has three latent factors, and they represent speaker, SNR, and duration information, respectively.

Generative Model

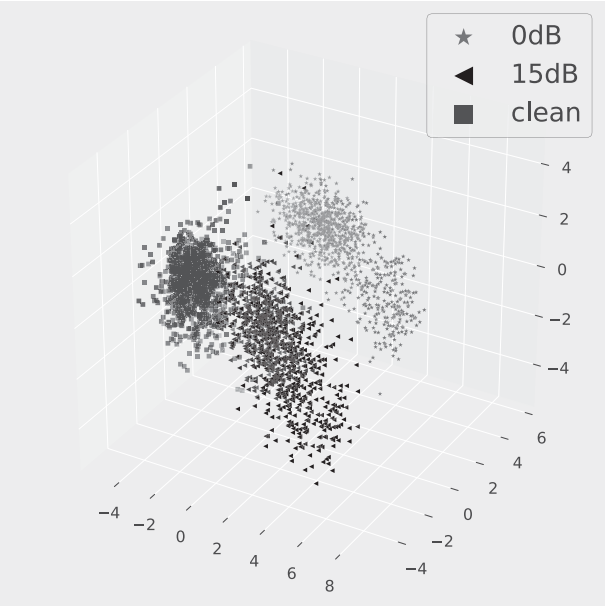
In classical PLDA, i-vectors from the same speaker share the same speaker factor. Inspired by this notion, SDI-PLDA assumes (1) that utterances with similar SNR will lead to i-vectors that possess similar SNR information and (2) that utterances of approximately the same length should lead to i-vectors that own similar duration information.

Figure 5.9 demonstrates that the assumptions above are reasonable. In Figure 5.9(a), we project the i-vectors to the first three principal components and denote three groups of i-vectors by three different colors. Each group corresponds to the same set of speakers whose utterances' SNR belong to the same SNR group. Grouping the i-vectors in this way is to ensure that any displacement in the clusters is not caused by speaker variability. Figure 5.9(a) clearly shows three clusters, and each cluster corresponds to one SNR group. To show that the second assumption is valid, we divided the clean i-vectors according to their utterance duration and plotted them on the first three principal components. Here, the word "clean" means that the i-vectors were obtained from clean telephone utterances. Again, each duration group in the legend of Figure 5.9(b) corresponds to the same set of speakers. Results clearly show that i-vectors are duration dependent and that the cluster locations depend on the utterance duration.

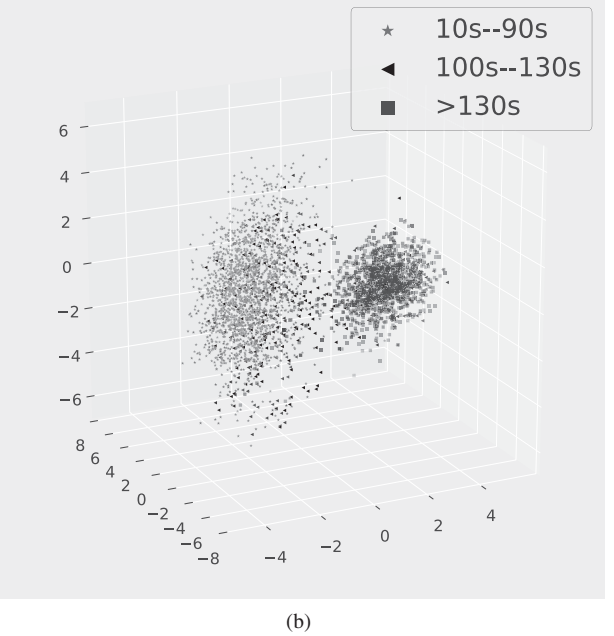
With the above assumptions, i-vectors can be generated from a linear model that comprises five components, namely speaker, SNR, duration, channel, and residue. Suppose we have a set of D-dimensional i-vectors

$$
\mathcal{X} = \{\hat{\mathbf{x}}_{ij}^{kp} | i = 1, \ldots, S; k = 1, \ldots, K; p = 1, \ldots, P; j = 1, \ldots, H_{ik}(p)\}
$$

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press



(a)



(b)

Figure 5.9 Projection of i-vectors on the first three principal components. (a) Utterances of different SNRs. (b) Utterances of different durations. *It may be better to view the color version of this figure, which is available at https://github.com/enmwmak/ML-for-Spkrec.*

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

Table 5.2 The definitions of the variables in Eq. 5.29.

Symbol	Dimension	Definition
$\mathbf{m}$	D	Global mean vector
$\mathbf{h}_i$	Q_1	Speaker factor with Gaussian prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{w}_k$	Q_2	SNR factor with Gaussian prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{y}_p$	Q_3	Duration factor with Gaussian prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$
$\boldsymbol{\epsilon}_{ij}^{kp}$	D	Residual vector with Gaussian prior $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$
$\mathbf{V}$	$D \times Q_1$	Speaker loading matrix
$\mathbf{U}$	$D \times Q_2$	SNR loading matrix
$\mathbf{R}$	$D \times Q_3$	Duration loading matrix

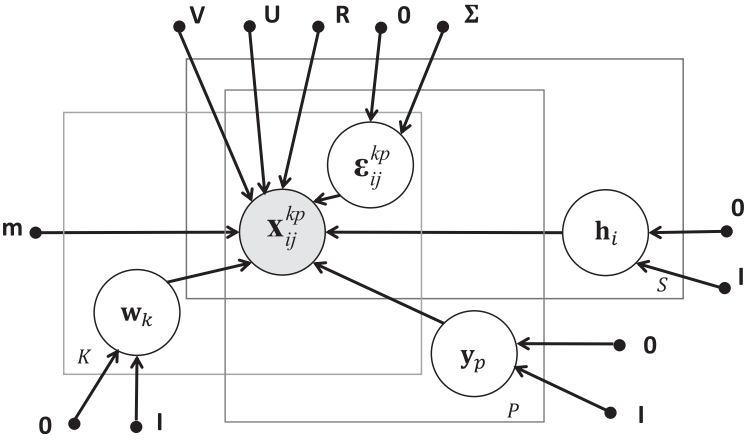


Figure 5.10 Probabilistic graphical model of SDI-PLDA.

obtained from S speakers, where $\hat{\mathbf{x}}_{ij}^{kp}$ is the jth i-vector from speaker i and k and p denote the SNR and duration groups, respectively. In SDI-PLDA, $\hat{\mathbf{x}}_{ij}^{kp}$ can be considered generated from the model:

$$\hat{\mathbf{x}}_{ij}^{kp} = \mathbf{m} + \mathbf{V}\mathbf{h}_i + \mathbf{U}\mathbf{w}_k + \mathbf{R}\mathbf{y}_p + \boldsymbol{\epsilon}_{ij}^{kp}, \tag{5.29}$$

where the variables in Eq. 5.29 are defined in Table 5.2. Note that $\mathbf{h}_i$, $\mathbf{w}_k$, and $\mathbf{y}_p$ are assumed to be independent in their prior. Figure 5.10 shows the graphical model of SDI-PLDA.

The SNR- and duration-invariant PLDA extends the conventional PLDA in that the former takes the advantage of having multiple labels (speaker IDs, SNR levels, and duration ranges) in the training set to train the loading matrices, whereas the latter only uses the speaker IDs. SDI-PLDA uses the additional duration subspace to leverage the duration information in the utterances. Different from the session- and speaker-dependent term $\mathbf{G}\mathbf{r}_{ij}$ in Eq. 5.4, the duration term $\mathbf{R}\mathbf{y}_p$ and the SNR term $\mathbf{U}\mathbf{w}_k$ in Eq. 5.29 depend on the duration and SNR groups, respectively.

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

Variational Bayes EM Algorithm

The parameters $\boldsymbol{\theta} = \{\mathbf{m}, \mathbf{V}, \mathbf{U}, \mathbf{R}, \boldsymbol{\Sigma}\}$ of the SNR- and duration-invariant PLDA model can be obtained by using the maximum likelihood principle. Specifically, denote $\boldsymbol{\theta}$ as the old parameters. We estimate the new parameters $\boldsymbol{\theta}'$ by maximizing the auxiliary function:

$$\begin{aligned} Q(\boldsymbol{\theta}'|\boldsymbol{\theta}) &= \mathbb{E}_{q(\underline{\mathbf{h}},\underline{\mathbf{w}},\underline{\mathbf{y}})}\Big[\log p(\mathcal{X},\underline{\mathbf{h}},\underline{\mathbf{w}},\underline{\mathbf{y}}|\boldsymbol{\theta}')\Big|\mathcal{X},\boldsymbol{\theta}\Big] \\ &= \mathbb{E}_{q(\underline{\mathbf{h}},\underline{\mathbf{w}},\underline{\mathbf{y}})}\Big[\sum\nolimits_{ikpj}\log\Big(p(\hat{\mathbf{x}}_{ij}^{kp}|\mathbf{h}_i,\mathbf{w}_k,\mathbf{y}_p,\boldsymbol{\theta}')p(\mathbf{h}_i,\mathbf{w}_k,\mathbf{y}_p)\Big)\Big|\mathcal{X},\boldsymbol{\theta}\Big]. \end{aligned} \tag{5.30}$$

Taking expectation with respect to $q(\underline{\mathbf{h}},\underline{\mathbf{w}},\underline{\mathbf{y}})$ suggests that it is necessary to compute the posterior distributions of the latent variables given the model parameters $\boldsymbol{\theta}$. Let $N_i = \sum_{kp} H_{ik}(p)$ be the number of training i-vectors from speaker i, $M_k = \sum_{ip} H_{ik}(p)$ be the number of training i-vectors from SNR group k, and $B_p = \sum_{ik} H_{ik}(p)$ be the number of training i-vectors from duration group p. Similar to duration-invariant PLDA, we may use variational Bayes to compute these posteriors:

$$\mathbf{L}_i^{(1)} = \mathbf{I} + N_i\mathbf{V}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{V} \quad i = 1,\ldots,S \tag{5.31a}$$

$$\mathbf{L}_k^{(2)} = \mathbf{I} + M_k\mathbf{U}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{U} \quad k = 1,\ldots,K \tag{5.31b}$$

$$\mathbf{L}_p^{(3)} = \mathbf{I} + B_p\mathbf{R}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R} \quad p = 1,\ldots,P \tag{5.31c}$$

$$\langle\mathbf{h}_i|\mathcal{X}\rangle = (\mathbf{L}_i^{(1)})^{-1}\mathbf{V}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\sum\nolimits_{kpj}(\hat{\mathbf{x}}_{ij}^{kp} - \mathbf{m} - \mathbf{U}\mathbf{w}_k^* - \mathbf{R}\mathbf{y}_p^*) \tag{5.31d}$$

$$\langle\mathbf{w}_k|\mathcal{X}\rangle = (\mathbf{L}_k^{(2)})^{-1}\mathbf{U}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\sum\nolimits_{ipj}(\hat{\mathbf{x}}_{ij}^{kp} - \mathbf{m} - \mathbf{V}\mathbf{h}_i^* - \mathbf{R}\mathbf{y}_p^*) \tag{5.31e}$$

$$\langle\mathbf{y}_p|\mathcal{X}\rangle = (\mathbf{L}_p^{(3)})^{-1}\mathbf{R}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\sum\nolimits_{ikj}(\hat{\mathbf{x}}_{ij}^{kp} - \mathbf{m} - \mathbf{V}\mathbf{h}_i^* - \mathbf{U}\mathbf{w}_k^*) \tag{5.31f}$$

$$\langle\mathbf{h}_i\mathbf{h}_i^{\mathsf{T}}|\mathcal{X}\rangle = (\mathbf{L}_i^{(1)})^{-1} + \langle\mathbf{h}_i|\mathcal{X}\rangle\langle\mathbf{h}_i|\mathcal{X}\rangle^{\mathsf{T}} \tag{5.31g}$$

$$\langle\mathbf{w}_k\mathbf{w}_k^{\mathsf{T}}|\mathcal{X}\rangle = (\mathbf{L}_k^{(2)})^{-1} + \langle\mathbf{w}_k|\mathcal{X}\rangle\langle\mathbf{w}_k|\mathcal{X}\rangle^{\mathsf{T}} \tag{5.31h}$$

$$\langle\mathbf{y}_p\mathbf{y}_p^{\mathsf{T}}|\mathcal{X}\rangle = (\mathbf{L}_p^{(3)})^{-1} + \langle\mathbf{y}_p|\mathcal{X}\rangle\langle\mathbf{y}_p|\mathcal{X}\rangle^{\mathsf{T}} \tag{5.31i}$$

$$\langle\mathbf{w}_k\mathbf{h}_i^{\mathsf{T}}|\mathcal{X}\rangle \approx \langle\mathbf{w}_k|\mathcal{X}\rangle\langle\mathbf{h}_i|\mathcal{X}\rangle^{\mathsf{T}} \tag{5.31j}$$

$$\langle\mathbf{h}_i\mathbf{w}_k^{\mathsf{T}}|\mathcal{X}\rangle \approx \langle\mathbf{h}_i|\mathcal{X}\rangle\langle\mathbf{w}_k|\mathcal{X}\rangle^{\mathsf{T}} \tag{5.31k}$$

$$\langle\mathbf{w}_k\mathbf{y}_p^{\mathsf{T}}|\mathcal{X}\rangle \approx \langle\mathbf{w}_k|\mathcal{X}\rangle\langle\mathbf{y}_p|\mathcal{X}\rangle^{\mathsf{T}} \tag{5.31l}$$

$$\langle\mathbf{y}_p\mathbf{w}_k^{\mathsf{T}}|\mathcal{X}\rangle \approx \langle\mathbf{y}_p|\mathcal{X}\rangle\langle\mathbf{w}_k|\mathcal{X}\rangle^{\mathsf{T}} \tag{5.31m}$$

$$\langle\mathbf{h}_i\mathbf{y}_p^{\mathsf{T}}|\mathcal{X}\rangle \approx \langle\mathbf{h}_i|\mathcal{X}\rangle\langle\mathbf{y}_p|\mathcal{X}\rangle^{\mathsf{T}} \tag{5.31n}$$

$$\langle\mathbf{y}_p\mathbf{h}_i^{\mathsf{T}}|\mathcal{X}\rangle \approx \langle\mathbf{y}_p|\mathcal{X}\rangle\langle\mathbf{h}_i|\mathcal{X}\rangle^{\mathsf{T}}, \tag{5.31o}$$

where $\mathbf{w}_k^*$, $\mathbf{y}_p^*$, and $\mathbf{h}_i^*$ denote the posterior mean of $\mathbf{w}_k$, $\mathbf{y}_p$, and $\mathbf{h}_i$ in the previous iteration, respectively.

Using Eq. 5.31(a)–(o), the model parameters $\boldsymbol{\theta}'$ can be estimated by maximizing the auxiliary function in Eq. 5.30, which gives

$$\mathbf{m} = \frac{1}{N}\sum\nolimits_{ikpj}\hat{\mathbf{x}}_{ij}^{kp} \tag{5.32a}$$

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

$$\mathbf{V}' = \left\{\sum_{ikpj}\left[(\hat{\mathbf{x}}_{ij}^{kp} - \mathbf{m})\langle\mathbf{h}_i|\mathcal{X}\rangle^{\mathsf{T}} - \mathbf{U}\langle\mathbf{w}_k\mathbf{h}_i^{\mathsf{T}}|\mathcal{X}\rangle - \mathbf{R}\langle\mathbf{y}_p\mathbf{h}_i^{\mathsf{T}}|\mathcal{X}\rangle\right]\right\} \left\{\sum_{ikpj}\langle\mathbf{h}_i\mathbf{h}_i^{\mathsf{T}}|\mathcal{X}\rangle\right\}^{-1} \tag{5.32b}$$

$$\mathbf{U}' = \left\{\sum_{ikpj}\left[(\hat{\mathbf{x}}_{ij}^{kp} - \mathbf{m})\langle\mathbf{w}_k|\mathcal{X}\rangle^{\mathsf{T}} - \mathbf{V}\langle\mathbf{h}_i\mathbf{w}_k^{\mathsf{T}}|\mathcal{X}\rangle - \mathbf{R}\langle\mathbf{y}_p\mathbf{w}_k^{\mathsf{T}}|\mathcal{X}\rangle\right]\right\} \left\{\sum_{ikpj}\langle\mathbf{w}_k\mathbf{w}_k^{\mathsf{T}}|\mathcal{X}\rangle\right\}^{-1} \tag{5.32c}$$

$$\mathbf{R}' = \left\{\sum_{ikpj}\left[(\hat{\mathbf{x}}_{ij}^{kp} - \mathbf{m})\langle\mathbf{y}_p|\mathcal{X}\rangle^{\mathsf{T}} - \mathbf{V}\langle\mathbf{h}_i\mathbf{y}_p^{\mathsf{T}}|\mathcal{X}\rangle - \mathbf{U}\langle\mathbf{w}_k\mathbf{y}_p^{\mathsf{T}}|\mathcal{X}\rangle\right]\right\} \left\{\sum_{ikpj}\langle\mathbf{y}_p\mathbf{y}_p^{\mathsf{T}}|\mathcal{X}\rangle\right\}^{-1} \tag{5.32d}$$

$$\begin{aligned}\boldsymbol{\Sigma}' = \frac{1}{N}\sum_{ikpj}\Big[&(\hat{\mathbf{x}}_{ij}^{kp} - \mathbf{m})(\hat{\mathbf{x}}_{ij}^{kp} - \mathbf{m})^{\mathsf{T}} - \mathbf{V}\langle\mathbf{h}_i|\mathcal{X}\rangle(\hat{\mathbf{x}}_{ij}^{kp} - \mathbf{m})^{\mathsf{T}} \\ &- \mathbf{U}\langle\mathbf{w}_k|\mathcal{X}\rangle(\hat{\mathbf{x}}_{ij}^{kp} - \mathbf{m})^{\mathsf{T}} - \mathbf{R}\langle\mathbf{y}_p|\mathcal{X}\rangle(\hat{\mathbf{x}}_{ij}^{kp} - \mathbf{m})^{\mathsf{T}}\Big],\end{aligned} \tag{5.32e}$$

where $N = \sum_{i=1}^{S} N_i = \sum_{k=1}^{K} M_k$. Algorithm 5 shows the procedure of applying the variational EM algorithm.

Algorithm 5 Variational Bayes EM algorithm for SNR- and duration-invariant PLDA

Input:

Development data set comprising i-vectors or LDA-projected i-vectors $\mathcal{X} = \{\hat{\mathbf{x}}_{ij}^{kp} | i = 1, \ldots, S; k = 1, \ldots, K; p = 1, \ldots, P; j = 1, \ldots, H_{ik}(p)\}$, with speaker labels, SNR group labels, and duration labels.

Initialization:

$\mathbf{y}_p^* \leftarrow \mathbf{0}$, $\mathbf{w}_k^* \leftarrow \mathbf{0}$;
$\boldsymbol{\Sigma} \leftarrow 0.01\mathbf{I}$;
$\mathbf{V}, \mathbf{U}, \mathbf{R} \leftarrow$ eigenvectors obtained from the PCA of $\mathcal{X}$;

Parameter Estimation:

Compute $\mathbf{m}$ via Eq. 5.32(a);
Compute $\mathbf{L}_i^{(1)}$, $\mathbf{L}_k^{(2)}$, and $\mathbf{L}_p^{(3)}$ according to Eq. 5.31(a) to Eq. 5.31(c), respectively;
Compute the posterior mean of $\mathbf{h}_i$ using Eq. 5.31(d);
Use the posterior mean of $\mathbf{h}_i$ computed in Step 3 to update the posterior means of $\mathbf{w}_k$ and $\mathbf{y}_p$ using Eq. 5.31(e)–(f);
Compute the other terms in the E-step, i.e., Eq. 5.31(g)–(o);
Update the model parameters using Eq. 5.32(b)–(e);
Set $\mathbf{y}_p^* = \langle\mathbf{y}_p|\mathcal{X}\rangle$, $\mathbf{w}_k^* = \langle\mathbf{w}_k|\mathcal{X}\rangle$, and $\mathbf{h}_i^* = \langle\mathbf{h}_i|\mathcal{X}\rangle$;
Go to Step 2 until convergence;

Return: the parameters of the SNR- and duration-invariant PLDA model $\boldsymbol{\theta} = \{\mathbf{m}, \mathbf{V}, \mathbf{U}, \mathbf{R}, \boldsymbol{\Sigma}\}$.

Likelihood Ratio Scores

Consider the general case where both SNR and duration of the target speaker's utterance and the test speaker's utterance are unknown. Let $\mathbf{x}_s$ and $\mathbf{x}_t$ be the i-vectors of the target speaker and test speaker respectively, the likelihood ratio score is

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

$$
\begin{aligned}
S_{\text{LR}}(\mathbf{x}_s, \mathbf{x}_t) &= \log \frac{P\left(\mathbf{x}_s, \mathbf{x}_t | \text{same-speaker}\right)}{P(\mathbf{x}_s, \mathbf{x}_t | \text{different-speakers})} \\
&= \text{const} + \frac{1}{2}\bar{\mathbf{x}}_s^{\mathsf{T}}\mathbf{Q}\bar{\mathbf{x}}_s + \frac{1}{2}\bar{\mathbf{x}}_t^{\mathsf{T}}\mathbf{Q}\bar{\mathbf{x}}_t + \bar{\mathbf{x}}_s^{\mathsf{T}}\mathbf{P}\bar{\mathbf{x}}_t,
\end{aligned} \tag{5.33}
$$

where

$$
\begin{aligned}
\bar{\mathbf{x}}_s &= \mathbf{x}_s - \mathbf{m}, \qquad \bar{\mathbf{x}}_t = \mathbf{x}_t - \mathbf{m}, \\
\mathbf{P} &= \boldsymbol{\Sigma}_{tot}^{-1}\boldsymbol{\Sigma}_{ac}(\boldsymbol{\Sigma}_{tot} - \boldsymbol{\Sigma}_{ac}\boldsymbol{\Sigma}_{tot}^{-1}\boldsymbol{\Sigma}_{ac})^{-1}, \\
\mathbf{Q} &= \boldsymbol{\Sigma}_{tot}^{-1} - (\boldsymbol{\Sigma}_{tot} - \boldsymbol{\Sigma}_{ac}\boldsymbol{\Sigma}_{tot}^{-1}\boldsymbol{\Sigma}_{ac})^{-1}, \\
\boldsymbol{\Sigma}_{ac} &= \mathbf{V}\mathbf{V}^{\mathsf{T}}, \text{ and } \boldsymbol{\Sigma}_{tot} = \mathbf{V}\mathbf{V}^{\mathsf{T}} + \mathbf{U}\mathbf{U}^{\mathsf{T}} + \mathbf{R}\mathbf{R}^{\mathsf{T}} + \boldsymbol{\Sigma}.
\end{aligned}
$$

See Section 3.4.3 for the derivation of Eq. 5.33. If we know the utterance duration and SNR, we may use the principle in Eq. 5.26 to derive the scoring function. SDI-PLDA scoring is computationally light because both $\mathbf{P}$ and $\mathbf{Q}$ can be computed during training.

5.4 Mixture of PLDA

In duration- and SNR-invariant PLDA, the duration and SNR variabilities are modeled by the duration and SNR subspace in the PLDA model. In case the variabilities are very large, e.g., some utterances are very clean but some are very noisy, then the variabilities may be better modeled by a mixture of PLDA models in which each model is responsible for a small range of variability. In this section, we will discuss the mixture models that are good at modeling utterances with a wide range of SNR. Much of the materials in this section are based on the authors' previous work on PLDA mixture models [47, 95, 171–173].

5.4.1 SNR-Independent Mixture of PLDA

In Eq. 3.70, the PLDA model assumes that the length-normalized i-vectors are Gaussian distributed. The modeling capability of a single Gaussian, however, is rather limited, causing deficiency of the Gaussian PLDA model in tackling noise and reverberation with varying levels. In such scenarios, it is more appropriate to model the i-vectors by a mixture of K factor analysis (FA) models. Specifically, the parameters of the mixture FA model are given by $\underline{\boldsymbol{\omega}} = \{\varphi_k, \mathbf{m}_k, \mathbf{V}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$, where φ_k, $\mathbf{m}_k$, $\mathbf{V}_k$, and $\boldsymbol{\Sigma}_k$ are the mixture, coefficient, mean vector, speaker loading matrix and covariance matrix of the k-th mixture, respectively. It is assumed that the full covariance matrices $\boldsymbol{\Sigma}_k$'s are capable of modeling the channel effect. Because there is no direct relationship between the mixture components and the SNR of utterances, we refer to this type of mixture models as SNR-independent mixture of PLDA (SI-mPLDA).

The following subsection will provide the derivation of the EM formulations for the scenario in which an i-vector is connected with K latent factors. After that, the formulations will be extended to the scenario in which each speaker has only one latent factor across all mixtures.

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

EM Formulation

Denote $\mathcal{Y} = \{y_{ijk}\}_{k=1}^{K}$ as the latent indicator variables identifying which of the K factor analysis models $\{\varphi_k, \mathbf{m}_k, \mathbf{V}_k, \mathbf{\Sigma}_k\}_{k=1}^{K}$ produces $\mathbf{x}_{ij}$. Specifically, $y_{ijk} = 1$ if the FA model k produces $\mathbf{x}_{ij}$, and $y_{ijk} = 0$ otherwise. Also denote $\mathcal{Z} = \{\mathbf{z}_{ik}\}_{k=1}^{K}$ as the latent factors for the K mixtures. Then, the auxiliary function in Eq. 3.71 can be written as

$$\begin{aligned}
Q(\underline{\boldsymbol{\omega}'}|\underline{\boldsymbol{\omega}}) &= \mathbb{E}_{\mathcal{Y},\mathcal{Z}}\{\log p(\mathcal{X},\mathcal{Y},\mathcal{Z}|\underline{\boldsymbol{\omega}}')|\mathcal{X},\underline{\boldsymbol{\omega}}\} \\
&= \mathbb{E}_{\mathcal{Y},\mathcal{Z}}\left\{\sum_{ijk} y_{ijk}\log\left[p(y_{ijk}|\underline{\boldsymbol{\omega}}')p(\mathbf{x}_{ij}|\mathbf{z}_{ik},y_{ijk}=1,\boldsymbol{\omega}_k')p(\mathbf{z}_{ik}|\underline{\boldsymbol{\omega}}')\right]\middle|\mathcal{X},\underline{\boldsymbol{\omega}}\right\} \\
&= \sum_{i=1}^{N}\sum_{j=1}^{H_i}\sum_{k=1}^{K}\mathbb{E}_{\mathcal{Y},\mathcal{Z}}\left\{y_{ijk}\log\left[\varphi_k'\mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}_k'+\mathbf{V}_k'\mathbf{z}_{ik},\mathbf{\Sigma}_k')\mathcal{N}(\mathbf{z}_{ik}|\mathbf{0},\mathbf{I})\right]\middle|\mathcal{X},\underline{\boldsymbol{\omega}}\right\}.
\end{aligned} \tag{5.34}$$

Although the ith training speaker have multiple (H_i) sessions, these sessions share the same set of latent variables $\{\mathbf{z}_{ik}\}_{k=1}^{K}$. To simplify notations, we will drop the $'$ symbol in Eq. 5.34:

$$\begin{aligned}
Q(\underline{\boldsymbol{\omega}}) &= \sum_{ijk}\mathbb{E}_{\mathcal{Y},\mathcal{Z}}\left\{y_{ijk}\log\left[\varphi_k\mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}_k+\mathbf{V}_k\mathbf{z}_{ik},\mathbf{\Sigma}_k)\mathcal{N}(\mathbf{z}_{ik}|\mathbf{0},\mathbf{I})\right]\middle|\mathcal{X}\right\} \\
&= \sum_{ijk}\left\langle y_{ijk}\left\{\log\varphi_k - \frac{1}{2}\log|\mathbf{\Sigma}_k|\right\}\middle|\mathcal{X}\right\rangle \\
&\quad - \frac{1}{2}\sum_{ijk}\left\langle y_{ijk}\left\{(\mathbf{x}_{ij}-\mathbf{m}_k-\mathbf{V}_k\mathbf{z}_{ik})^{\mathsf{T}}\mathbf{\Sigma}_k^{-1}(\mathbf{x}_{ij}-\mathbf{m}_k-\mathbf{V}_k\mathbf{z}_{ik})\right\}\middle|\mathcal{X}\right\rangle \\
&\quad - \frac{1}{2}\sum_{ijk}\left\langle y_{ijk}\mathbf{z}_{ik}{}^{\mathsf{T}}\mathbf{z}_{ik}\middle|\mathcal{X}\right\rangle \\
&= \sum_{ijk}\langle y_{ijk}|\mathcal{X}\rangle\left[\log\varphi_k - \frac{1}{2}\log|\mathbf{\Sigma}_k| - \frac{1}{2}(\mathbf{x}_{ij}-\mathbf{m}_k)^{\mathsf{T}}\mathbf{\Sigma}_k^{-1}(\mathbf{x}_{ij}-\mathbf{m}_k)\right] \\
&\quad + \sum_{ijk}(\mathbf{x}_{ij}-\mathbf{m}_k)^{\mathsf{T}}\mathbf{\Sigma}_k^{-1}\mathbf{V}_k\langle y_{ijk}\mathbf{z}_{ik}|\mathcal{X}\rangle \\
&\quad - \frac{1}{2}\left[\sum_{ijk}\mathrm{tr}\left\{\left(\mathbf{V}_k^{\mathsf{T}}\mathbf{\Sigma}_k^{-1}\mathbf{V}_k+1\right)\left\langle y_{ijk}\mathbf{z}_{ik}\mathbf{z}_{ik}^{\mathsf{T}}|\mathcal{X}\right\rangle\right\}\right].
\end{aligned} \tag{5.35}$$

To compute $\mathbf{m}_k$, we follow the two-step EM as suggested in [174]. Specifically, we drop $\mathbf{z}_{ik}$ at this stage of EM and setting[3]

$$\frac{\partial Q}{\partial \mathbf{m}_k} = -\sum_{ij}\langle y_{ijk}|\mathcal{X}\rangle\left(\mathbf{\Sigma}_k^{-1}\mathbf{m}_k - \mathbf{\Sigma}_k^{-1}\mathbf{x}_{ij}\right) = 0,$$

[3] In [174], the first step in the two-step EM does not consider $\mathbf{z}_{ik}$. As a result, the 6th row of Eq. 5.35 is not included in the derivative.

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

which results in

$$\mathbf{m}_k = \frac{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\langle y_{ijk}|\mathcal{X}\rangle \mathbf{x}_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\langle y_{ijk}|\mathcal{X}\rangle}.$$

To find $\mathbf{V}_k$, we compute

$$\frac{\partial Q}{\partial \mathbf{V}_k} = \sum_{ij}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_{ij}-\mathbf{m}_k)\langle y_{ijk}\mathbf{z}_{ik}^{\mathsf{T}}|\mathcal{X}\rangle - \sum_{ij}\boldsymbol{\Sigma}_k^{-1}\mathbf{V}_k\left\langle y_{ijk}\mathbf{z}_{ik}\mathbf{z}_{ik}^{\mathsf{T}}|\mathcal{X}\right\rangle. \tag{5.36}$$

Setting $\frac{\partial Q}{\partial \mathbf{V}_k} = 0$, we obtain

$$\mathbf{V}_k = \left[\sum_{ij}(\mathbf{x}_{ij}-\mathbf{m}_k)\langle y_{ijk}\mathbf{z}_{ik}|\mathcal{X}\rangle^{\mathsf{T}}\right]\left[\sum_{ij}\left\langle y_{ijk}\mathbf{z}_{ik}\mathbf{z}_{ik}^{\mathsf{T}}|\mathcal{X}\right\rangle\right]^{-1}. \tag{5.37}$$

To find $\boldsymbol{\Sigma}_k$, we evaluate

$$\begin{aligned}\frac{\partial Q}{\partial \boldsymbol{\Sigma}_k^{-1}} &= \frac{1}{2}\sum_{ij}\langle y_{ij}|\mathcal{X}\rangle\left[\boldsymbol{\Sigma}_k - (\mathbf{x}_{ij}-\mathbf{m}_k)(\mathbf{x}_{ij}-\mathbf{m}_k)^{\mathsf{T}}\right]\\ &\quad + \sum_{ij}(\mathbf{x}_{ij}-\mathbf{m}_k)\langle y_{ijk}\mathbf{z}_{ik}^{\mathsf{T}}|\mathcal{X}\rangle\mathbf{V}_k^{\mathsf{T}} - \frac{1}{2}\mathbf{V}_k\left[\sum_{ij}\langle y_{ijk}\mathbf{z}_{ik}\mathbf{z}_{ik}^{\mathsf{T}}|\mathcal{X}\rangle\right]\mathbf{V}_k^{\mathsf{T}}.\end{aligned} \tag{5.38}$$

Substituting Eq. 5.37 into Eq. 5.38 and setting $\frac{\partial Q}{\partial \boldsymbol{\Sigma}_k^{-1}} = 0$, we have

$$\begin{aligned}&\sum_{ij}\langle y_{ijk}|\mathcal{X}\rangle\boldsymbol{\Sigma}_k\\ &= \sum_{ij}\left[\langle y_{ijk}|\mathcal{X}\rangle(\mathbf{x}_{ij}-\mathbf{m}_k)(\mathbf{x}_{ij}-\mathbf{m}_k)^{\mathsf{T}} - (\mathbf{x}_{ij}-\mathbf{m}_k)\langle y_{ijk}\mathbf{z}_{ik}^{\mathsf{T}}|\mathcal{X}\rangle\mathbf{V}_k^{\mathsf{T}}\right].\end{aligned}$$

Rearranging, we have

$$\boldsymbol{\Sigma}_k = \frac{\sum_{ij}\left[\langle y_{ijk}|\mathcal{X}\rangle(\mathbf{x}_{ij}-\mathbf{m}_k)(\mathbf{x}_{ij}-\mathbf{m}_k)^{\mathsf{T}} - \mathbf{V}_k\langle y_{ijk}\mathbf{z}_{ik}|\mathcal{X}\rangle(\mathbf{x}_{ij}-\mathbf{m}_k)^{\mathsf{T}}\right]}{\sum_{ij}\langle y_{ijk}|\mathcal{X}\rangle}.$$

To compute φ_k, we optimize $Q(\underline{\boldsymbol{\omega}})$ subject to the constraint $\sum_k \varphi_k = 1$. This can be achieved by introducing a Lagrange multiplier λ such that $Q'(\underline{\boldsymbol{\omega}}) = Q(\underline{\boldsymbol{\omega}}) + \lambda(\sum_k \varphi_k - 1)$. λ can be found by setting $\frac{\partial Q'}{\partial \varphi_k} = 0$, which results in

$$-\varphi_k\lambda = \sum_{ij}\langle y_{ijk}|\mathcal{X}\rangle. \tag{5.39}$$

Summing both side of Eq. 5.39 from 1 to K, we obtain

$$\lambda = -\sum_{ijk}\langle y_{ijk}|\mathcal{X}\rangle. \tag{5.40}$$

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

Substituting Eq. 5.40 into Eq. 5.39 and rearranging, we obtain

$$\varphi_k = \frac{\sum_{ij}\langle y_{ijk}|\mathcal{X}\rangle}{\sum_{ijl}\langle y_{ijl}|\mathcal{X}\rangle}.$$

E-Step: Computing Posterior Expectations

Because the mixture model comprises more than one latent variable, formally these latent variables should be inferred through the variational Bayes (VB) method in which the variational distributions over the latent variables $\mathbf{z}_{ik}$ and y_{ijk} are assumed to be factorizable. Specifically, we have $q(\mathbf{z}_{ik}, y_{ijk}) = q(\mathbf{z}_{ik})q(y_{ijk})$. The VB method is able to find the closest variational distribution to the true joint posterior distribution of two dependent latent variables $p(\mathbf{z}_{ik}, y_{ijk}|\mathcal{X})$. In the VB-E step, the optimal variational distribution or variational parameters with the largest lower-bound of the likelihood are estimated. Then, in the VB-M step, given the updated variational distribution, we update the model parameters $\{\varphi_k, \mathbf{m}_k, \mathbf{V}_k, \mathbf{\Sigma}_k\}_{k=1}^K$ by maximizing the variational lower bound.

Rather than using the more complex VB method, we assume that the latent variable $\mathbf{z}_{ik}$ is posteriorly independent of y_{ijk}, that is,

$$p(\mathbf{z}_{ik}, y_{ijk}|\mathcal{X}_i) = p(\mathbf{z}_{ik}|\mathcal{X}_i)p(y_{ijk}|\mathbf{x}_{ij}).$$

This assumption is similar to that of traditional continuous density HMM in which the HMM states and Gaussian mixtures are assumed independent. Therefore, in the E-step, we compute the posterior means $\langle y_{ijk}\mathbf{z}_{ik}|\mathcal{X}\rangle$ and posterior moment $\langle y_{ijk}\mathbf{z}_{ik}\mathbf{z}_{ik}^{\mathsf{T}}|\mathcal{X}\rangle$, where $j = 1, \ldots, H_i$. The joint posterior expectations are given by:

$$\begin{aligned}\langle y_{ijk}\mathbf{z}_{ik}|\mathcal{X}_i\rangle &= \langle y_{ijk}|\mathbf{x}_{ij}\rangle\langle \mathbf{z}_{ik}|\mathcal{X}_i\rangle \\ \langle y_{ijk}\mathbf{z}_{ik}\mathbf{z}_{ik}^{\mathsf{T}}|\mathcal{X}_i\rangle &= \langle y_{ijk}|\mathbf{x}_{ij}\rangle\langle \mathbf{z}_{ik}\mathbf{z}_{ik}^{\mathsf{T}}|\mathcal{X}_i\rangle.\end{aligned} \tag{5.41}$$

Figure 5.11 shows the relation among $\mathbf{z}_{ik}$, $\mathbf{x}_{ij}$ and y_{ijk} [171]. In the figure, $\underline{\boldsymbol{\omega}} = \{\varphi_k, \mathbf{m}_k, \mathbf{\Sigma}_k, \mathbf{V}_k\}_{k=1}^K$. In the diagram, $\underline{\mathbf{m}} = \{\mathbf{m}_k\}_{k=1}^K$, $\underline{\mathbf{V}} = \{\mathbf{V}_k\}_{k=1}^K$, $\underline{\mathbf{\Sigma}} = \{\mathbf{\Sigma}_k\}_{k=1}^K$, and $\underline{\varphi} = \{\varphi_k\}_{k=1}^K$.

Let $y_{i\cdot k}$ be the indicator variables of the H_i i-vectors from the ith speaker for the kth mixture. We use the Bayes rule to estimate the joint posterior expectations as follows:

$$\begin{aligned}p(\mathbf{z}_{ik}, y_{i\cdot k}|\mathcal{X}_i) &\propto p(\mathcal{X}_i|\mathbf{z}_{ik}, y_{i\cdot k} = 1)p(\mathbf{z}_{ik}, y_{i\cdot k}) \\ &= p(\mathcal{X}_i|\mathbf{z}_{ik}, y_{i\cdot k} = 1)p(y_{i\cdot k})p(\mathbf{z}_{ik}) \qquad \because \mathbf{z}_{ik} \text{ and } y_{i\cdot k} \text{ are independent} \\ &= \prod_{j=1}^{H_i}\left[\varphi_k p(\mathbf{x}_{ij}|y_{ijk} = 1, \mathbf{z}_{ik})\right]^{y_{ijk}} p(\mathbf{z}_{ik}) \qquad \text{[1, Eq. 9.38]} \\ &= \underbrace{p(\mathbf{z}_{ik})\prod_{j=1}^{H_i}\left[\mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}_k + \mathbf{V}_k\mathbf{z}_{ik}, \mathbf{\Sigma}_k)\right]^{y_{ijk}}}_{\propto\, p(\mathbf{z}_{ik}|\mathcal{X}_i)} \varphi_k^{y_{ijk}}.\end{aligned} \tag{5.42}$$

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

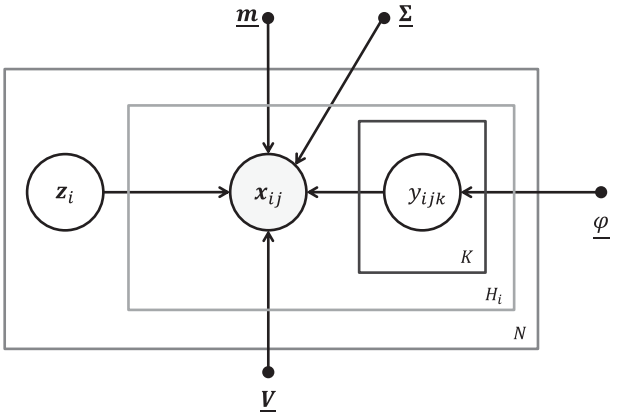


Figure 5.11 Probabilistic graphical model representing SNR-independent mixture of PLDA. [Reprinted from *Mixture of PLDA for Noise Robust I-Vector Speaker Verification (Figure 2), M.W. Mak, X.M Pang and J.T. Chien., IEEE/ACM Trans. on Audio Speech and Language Processing, vol. 24, No. 1, pp. 130–142, Jan. 2016, with permission of IEEE*]

To find the posterior of $\mathbf{z}_{ik}$, we extract terms dependent on $\mathbf{z}_{ik}$ from Eq. 5.42 as follows:

$$
\begin{aligned}
&p(\mathbf{z}_{ik}|\mathcal{X}_i) \\
&\quad \propto \exp\left\{-\frac{1}{2}\sum_{j=1}^{H_i} y_{ijk}(\mathbf{x}_{ij}-\mathbf{m}_k-\mathbf{V}_k\mathbf{z}_{ik})^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_{ij}-\mathbf{m}_k-\mathbf{V}_k\mathbf{z}_{ik})-\frac{1}{2}\mathbf{z}_{ik}^{\mathsf{T}}\mathbf{z}_{ik}\right\} \\
&\quad = \exp\left\{\mathbf{z}_{ik}^{\mathsf{T}}\mathbf{V}_k^{\mathsf{T}}\sum_{j\in\mathcal{H}_{ik}}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_{ij}-\mathbf{m}_k)-\frac{1}{2}\mathbf{z}_{ik}^{\mathsf{T}}\left(\mathbf{I}+\sum_{j\in\mathcal{H}_{ik}}\mathbf{V}_k^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}\mathbf{V}_k\right)\mathbf{z}_{ik}\right\}, \quad (5.43)
\end{aligned}
$$

where $\mathcal{H}_{ik}$ comprises the indexes of speaker i's i-vectors that aligned to mixture k. Comparing Eq. 3.77 with Eq. 5.43, we have

$$
\begin{aligned}
\langle \mathbf{z}_{ik}|\mathcal{X}_i\rangle &= \mathbf{L}_{ik}^{-1}\mathbf{V}_k^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}\sum_{j\in\mathcal{H}_{ik}}(\mathbf{x}_{ij}-\mathbf{m}_k) \\
\langle \mathbf{z}_{ik}\mathbf{z}_{ik}^{\mathsf{T}}|\mathcal{X}_i\rangle &= \mathbf{L}_{ik}^{-1}+\langle \mathbf{z}_{ik}|\mathcal{X}_i\rangle\langle \mathbf{z}_{ik}^{\mathsf{T}}|\mathcal{X}_i\rangle,
\end{aligned} \quad (5.44)
$$

where

$$
\mathbf{L}_{ik} = \mathbf{I} + H_{ik}\mathbf{V}_k^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}\mathbf{V}_k, \quad (5.45)
$$

where H_{ik} is the number (occupancy count) of i-vectors from speaker i aligned to mixture j.

Eq. 5.44 suggests that only H_{ik} out of H_i i-vectors from speaker i are generated by mixture k. This characteristic is crucial for proper modeling of i-vectors when they are obtained from utterances with large difference in SNR. This is because these i-vectors tend to fall on different areas of the i-vector space (see Hypothesis I in Section II of [171]). For example, the training i-vectors in [171] were obtained from utterances having three noise levels: 6dB, 15dB, and clean. Therefore, when $K = 3$, Mixtures 1,

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

2, and 3 will be responsible for generating i-vectors whose utterances have noise level of 6dB, 15dB, and clean, respectively. This characteristic also causes the PLDA mixture model different from that of [175]. In particular, in [175], a mixture component is first chosen for the i-vector of a particular speaker; then the selected component generates all of the i-vectors from that speaker. Clearly, this strategy is constrained to the case in which the i-vectors from a speaker are obtained from a similar acoustic environment.

To estimate the posterior expectation of y_{ijk}, we may use the Bayes rule:

$$\begin{aligned}\langle y_{ijk}|\mathcal{X}_i\rangle &= P(y_{ijk}=1|\mathbf{x}_{ij},\underline{\boldsymbol{\omega}})\\ &= \frac{P(y_{ijk}=1)p(\mathbf{x}_{ij}|y_{ijk}=1,\underline{\boldsymbol{\omega}})}{\sum_{r=1}^{K}P(y_{ijr}=1)p(\mathbf{x}_{ij}|y_{ijr}=1,\underline{\boldsymbol{\omega}})}\\ &= \frac{\varphi_k\mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}_k,\mathbf{V}_k\mathbf{V}_k^{\mathsf{T}}+\boldsymbol{\Sigma}_k)}{\sum_{r=1}^{K}\varphi_r\mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}_r,\mathbf{V}_r\mathbf{V}_r^{\mathsf{T}}+\boldsymbol{\Sigma}_r)}.\end{aligned} \tag{5.46}$$

Note that in Eq. 5.46,

$$\begin{aligned}p(\mathbf{x}_{ij}|y_{ijk}=1,\underline{\boldsymbol{\omega}}) &= \int p(\mathbf{x}_{ij}|\mathbf{z},y_{ijk}=1,\underline{\boldsymbol{\omega}})p(\mathbf{z})\mathrm{d}\mathbf{z}\\ &= \int \mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}_k+\mathbf{V}_k\mathbf{z},\boldsymbol{\Sigma}_k)\mathcal{N}(\mathbf{z}|\mathbf{0},\mathbf{I})\mathrm{d}\mathbf{z}\\ &= \mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}_k,\mathbf{V}_k\mathbf{V}_k^{\mathsf{T}}+\boldsymbol{\Sigma}_k).\end{aligned}$$

Sharing Latent Factors

Note that in Eq. 5.44, each speaker needs K latent factors $\mathbf{z}_{ik}$. Another way is to let the speaker to share the same latent factor, i.e., $\mathbf{z}_{ik}=\mathbf{z}_i\ \forall i$. Therefore, Eq. 5.34 changes into

$$Q(\underline{\boldsymbol{\omega}}'|\underline{\boldsymbol{\omega}}) = \sum\nolimits_{ijk}\mathbb{E}_{\boldsymbol{y},\mathcal{Z}}\left\{y_{ijk}\log\left[\varphi_k'\mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}_k'+\mathbf{V}_k'\mathbf{z}_i,\boldsymbol{\Sigma}_k')\mathcal{N}(\mathbf{z}_i|\mathbf{0},\mathbf{I})\right]\middle|\mathcal{X},\underline{\boldsymbol{\omega}}\right\}. \tag{5.47}$$

Let $y_{i\cdot\cdot}$ be the indicator variables for all possible sessions and mixture components for speaker i. The joint posterior density in Eq. 5.42 turns into

$$\begin{aligned}p(\mathbf{z}_i,y_{i\cdot\cdot}|\mathcal{X}_i) &\propto p(\mathcal{X}_i|\mathbf{z}_i,y_{i\cdot\cdot}=1)p(\mathbf{z}_i,y_{i\cdot\cdot})\\ &= p(\mathcal{X}_i|\mathbf{z}_i,y_{i\cdot\cdot}=1)p(y_{i\cdot\cdot})p(\mathbf{z}_i) \qquad \because \mathbf{z}_i \text{ and } y_{i\cdot\cdot} \text{ are independent}\\ &= \prod_{j=1}^{H_i}\prod_{k=1}^{K}\left[\varphi_k p(\mathbf{x}_{ij}|y_{ijk}=1,\mathbf{z}_i)\right]^{y_{ijk}}p(\mathbf{z}_i) \qquad \text{[1, Eq. 9.38]}\\ &= \underbrace{p(\mathbf{z}_i)\prod_{j=1}^{H_i}\prod_{k=1}^{K}\left[\mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}_k+\mathbf{V}_k\mathbf{z}_i,\boldsymbol{\Sigma}_k)\right]^{y_{ijk}}}_{\propto\, p(\mathbf{z}_i|\mathcal{X}_i)}\varphi_k^{y_{ijk}},\end{aligned} \tag{5.48}$$

where we have used the property that for each i and j, only one of y_{ijk}'s equals to 1, the rest are 0. Taking out terms depending on $\mathbf{z}_i$ from Eq. 5.48 and contrasting with Eq. 3.77, we have the posterior expectations as follows:

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

$$\begin{aligned}\langle \mathbf{z}_i|\mathcal{X}_i\rangle &= \mathbf{L}_i^{-1}\sum_{k=1}^{K}\sum_{j\in\mathcal{H}_{ik}}\mathbf{V}_k^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_{ij}-\mathbf{m}_k)\\ \langle \mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X}_i\rangle &= \mathbf{L}_i^{-1}+\langle \mathbf{z}_i|\mathcal{X}_i\rangle\langle \mathbf{z}_i^{\mathsf{T}}|\mathcal{X}_i\rangle.\end{aligned} \tag{5.49}$$

We also have the posterior precision as follows:

$$\mathbf{L}_i = \mathbf{I}+\sum_{k=1}^{K}\sum_{j\in\mathcal{H}_{ik}}\mathbf{V}_k^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}\mathbf{V}_k, \tag{5.50}$$

where $\mathcal{H}_{ik}$ contains the indexes of the i-vectors from speaker i that aligned to mixture k.

In summary, we have the following EM formulations:

E-Step:

$$\langle y_{ijk}|\mathcal{X}\rangle = \frac{\varphi_k\mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}_k,\mathbf{V}_k\mathbf{V}_k^{\mathsf{T}}+\boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K}\varphi_l\mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}_l,\mathbf{V}_l\mathbf{V}_l^{\mathsf{T}}+\boldsymbol{\Sigma}_l)} \tag{5.51a}$$

$$\mathbf{L}_i = \mathbf{I}+\sum_{k=1}^{K}H_{ik}\mathbf{V}_k^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}\mathbf{V}_k \tag{5.51b}$$

$$\langle y_{ijk}\mathbf{z}_i|\mathcal{X}\rangle = \langle y_{ijk}|\mathcal{X}\rangle\langle \mathbf{z}_i|\mathcal{X}\rangle \tag{5.51c}$$

$$\langle \mathbf{z}_i|\mathcal{X}\rangle = \mathbf{L}_i^{-1}\sum_{k=1}^{K}\mathbf{V}_k^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}\sum_{j\in\mathcal{H}_{ik}}(\mathbf{x}_{ij}-\mathbf{m}_k) \tag{5.51d}$$

$$\langle y_{ijk}\mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X}\rangle = \langle y_{ijk}|\mathcal{X}\rangle\langle \mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X}\rangle \tag{5.51e}$$

$$\langle \mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X}\rangle = \mathbf{L}_i^{-1}+\langle \mathbf{z}_i|\mathcal{X}\rangle\langle \mathbf{z}_i|\mathcal{X}\rangle^{\mathsf{T}} \tag{5.51f}$$

M-Step:

$$\mathbf{m}_k' = \frac{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\langle y_{ijk}|\mathcal{X}\rangle\mathbf{x}_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\langle y_{ijk}|\mathcal{X}\rangle} \tag{5.52a}$$

$$\varphi_k' = \frac{\sum_{ij}\langle y_{ijk}|\mathcal{X}\rangle}{\sum_{ijl}\langle y_{ijl}|\mathcal{X}\rangle} \tag{5.52b}$$

$$\mathbf{V}_k' = \left[\sum_{ij}(\mathbf{x}_{ij}-\mathbf{m}_k')\langle y_{ijk}\mathbf{z}_i|\mathcal{X}\rangle^{\mathsf{T}}\right]\left[\sum_{ij}\left\langle y_{ijk}\mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X}\right\rangle\right]^{-1} \tag{5.52c}$$

$$\boldsymbol{\Sigma}_k' = \frac{\sum_{ij}\left[\langle y_{ijk}|\mathcal{X}\rangle(\mathbf{x}_{ij}-\mathbf{m}_k')(\mathbf{x}_{ij}-\mathbf{m}_k')^{\mathsf{T}}-\mathbf{V}'\langle y_{ijk}\mathbf{z}_i|\mathcal{X}\rangle(\mathbf{x}_{ij}-\mathbf{m}_k')^{\mathsf{T}}\right]}{\sum_{ij}\langle y_{ijk}|\mathcal{X}\rangle} \tag{5.52d}$$

where $\langle y_{ijk}|\mathcal{X}\rangle \equiv \mathbb{E}_{\mathcal{Y}}\left\{y_{ijk}|\mathcal{X},\underline{\boldsymbol{\omega}}\right\}$, $\langle y_{ijk}\mathbf{z}_i|\mathcal{X}\rangle \equiv \mathbb{E}_{\mathcal{Y},\mathcal{Z}}\left\{y_{ijk}\mathbf{z}_i|\mathcal{X},\underline{\boldsymbol{\omega}}\right\}$, and $\langle y_{ijk}\mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X}\rangle \equiv \mathbb{E}_{\mathcal{Y},\mathcal{Z}}\left\{y_{ijk}\mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X},\underline{\boldsymbol{\omega}}\right\}$.

Mixture PLDA Scoring

Given the target speaker's i-vector $\mathbf{x}_s$ and a test i-vector $\mathbf{x}_t$, the same-speaker likelihood (numerator) can be written as:

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

$$
\begin{aligned}
&p(\mathbf{x}_s, \mathbf{x}_t|\text{same-speaker}) \\
&= \sum_{k_s=1}^{K} \sum_{k_t=1}^{K} \int p(\mathbf{x}_s, \mathbf{x}_t, y_{k_s} = 1, y_{k_t} = 1, \mathbf{z}|\underline{\boldsymbol{\omega}})\mathrm{d}\mathbf{z} \\
&= \sum_{k_s=1}^{K} \sum_{k_t=1}^{K} P(y_{k_s} = 1, y_{k_t} = 1|\underline{\boldsymbol{\omega}}) \int p(\mathbf{x}_s, \mathbf{x}_t|y_{k_s} = 1, y_{k_t} = 1, \mathbf{z}, \underline{\boldsymbol{\omega}})p(\mathbf{z})\mathrm{d}\mathbf{z} \\
&= \sum_{k_s=1}^{K} \sum_{k_t=1}^{K} \varphi_{k_s}\varphi_{k_t} \int p(\mathbf{x}_s, \mathbf{x}_t|y_{k_s} = 1, y_{k_t} = 1, \mathbf{z}, \underline{\boldsymbol{\omega}})p(\mathbf{z})\mathrm{d}\mathbf{z} \\
&= \sum_{k_s=1}^{K} \sum_{k_t=1}^{K} \varphi_{k_s}\varphi_{k_t} \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_s^{\mathsf{T}} & \mathbf{x}_t^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}} \middle| \begin{bmatrix}\mathbf{m}_{k_s}^{\mathsf{T}} & \mathbf{m}_{k_t}^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}, \hat{\mathbf{V}}_{k_sk_t}\hat{\mathbf{V}}_{k_sk_t}^{\mathsf{T}} + \hat{\boldsymbol{\Sigma}}_{k_sk_t}\right).
\end{aligned} \tag{5.53}
$$

A similar procedure is applied to compute the different-speaker likelihood. Therefore, the likelihood ratio score for the mixture of PLDA becomes is

$$
\begin{aligned}
&S_{\text{mLR}}(\mathbf{x}_s, \mathbf{x}_t) \\
&= \frac{\sum_{k_s=1}^{K} \sum_{k_t=1}^{K} \varphi_{k_s}\varphi_{k_t} \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_s^{\mathsf{T}} & \mathbf{x}_t^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}} \middle| \begin{bmatrix}\mathbf{m}_{k_s}^{\mathsf{T}} & \mathbf{m}_{k_t}^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}, \hat{\mathbf{V}}_{k_sk_t}\hat{\mathbf{V}}_{k_sk_t}^{\mathsf{T}} + \hat{\boldsymbol{\Sigma}}_{k_sk_t}\right)}{\left[\sum_{k_s=1}^{K} \varphi_{k_s} \mathcal{N}\left(\mathbf{x}_s|\mathbf{m}_{k_s}, \mathbf{V}_{k_s}\mathbf{V}_{k_s}^{\mathsf{T}} + \boldsymbol{\Sigma}_{k_s}\right)\right]\left[\sum_{k_t=1}^{K} \varphi_{k_t} \mathcal{N}\left(\mathbf{x}_t|\mathbf{m}_{k_t}, \mathbf{V}_{k_t}\mathbf{V}_{k_t}^{\mathsf{T}} + \boldsymbol{\Sigma}_{k_t}\right)\right]}
\end{aligned} \tag{5.54}
$$

where $\hat{\boldsymbol{\Sigma}}_{k_sk_t} = \text{diag}\{\boldsymbol{\Sigma}_{k_s}, \boldsymbol{\Sigma}_{k_t}\}$ and $\hat{\mathbf{V}}_{k_sk_t} = [\mathbf{V}_{k_s}^{\mathsf{T}} \ \ \mathbf{V}_{k_t}^{\mathsf{T}}]^{\mathsf{T}}$.

5.4.2 SNR-Dependent Mixture of PLDA

The SNR-dependent mixture of PLDA (SD-mPLDA) uses the SNR of utterances to guide the clustering of i-vectors. As a result, the alignment of i-vectors is based on the posterior probabilities of SNR rather than the posterior probabilities of i-vectors, as in the SI-PLDA in Section 5.4.1.

In SD-mPLDA, i-vectors are modeled by a mixture of SNR-dependent factor analyzers with parameters:

$$
\underline{\boldsymbol{\theta}} = \{\underline{\boldsymbol{\lambda}}, \underline{\boldsymbol{\omega}}\} = \{\boldsymbol{\lambda}_k, \boldsymbol{\omega}_k\}_{k=1}^{K} = \{\pi_k, \mu_k, \sigma_k, \mathbf{m}_k, \mathbf{V}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}, \tag{5.55}
$$

where $\boldsymbol{\lambda}_k = \{\pi_k, \mu_k, \sigma_k\}$ contains the mixture coefficient, mean and standard deviation of the kth SNR group, and $\boldsymbol{\omega}_k = \{\mathbf{m}_k, \mathbf{V}_k, \boldsymbol{\Sigma}_k\}$ comprises the mean i-vector, factor loading matrix, and residual covariance matrix of the FA model associated with SNR group k.

M-Step: Maximizing Expectation of Complete Likelihood

Let $\mathcal{Y} = \{y_{ijk}\}_{k=1}^{K}$ be the latent indicator variables identifying which of the K FA models should be selected based on the SNR of the training utterances. More precisely, if the kth FA model produces $\mathbf{x}_{ij}$, then $y_{ijk} = 1$; otherwise $y_{ijk} = 0$. We also let $\mathcal{L} = \{\ell_{ij}; i = 1, \ldots, N; j = 1, \ldots, H_i\}$ be the SNR of the training utterances. Then, the auxiliary function for deriving the EM formulation can be written as

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

$$
\begin{aligned}
Q(\underline{\boldsymbol{\theta}}'|\underline{\boldsymbol{\theta}}) &= \mathbb{E}_{\mathcal{Y},\mathcal{Z}}\{\log p(\mathcal{X},\mathcal{L},\mathcal{Y},\mathcal{Z}|\underline{\boldsymbol{\theta}}')|\mathcal{X},\mathcal{L},\underline{\boldsymbol{\theta}}\} \\
&= \mathbb{E}_{\mathcal{Y},\mathcal{Z}}\left\{\sum_{ijk} y_{ijk}\log\left[p(\ell_{ij}|y_{ijk}=1)p(y_{ijk})p(\mathbf{x}_{ij}|\mathbf{z}_i,y_{ijk}=1,\boldsymbol{\omega}'_k)p(\mathbf{z}_i)\right]\middle|\mathcal{X},\mathcal{L},\underline{\boldsymbol{\theta}}\right\} \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{H_i}\mathbb{E}_{\mathcal{Y},\mathcal{Z}}\Big\{y_{ijk}\log\left[\mathcal{N}(\ell_{ij}|\mu'_k,\sigma'_k)\pi'_k\mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}'_k+\mathbf{V}'_k\mathbf{z}_i,\boldsymbol{\Sigma}'_k)\right. \\
&\qquad \left.\mathcal{N}(\mathbf{z}_i|\mathbf{0},\mathbf{I})\right]\Big|\mathcal{X},\mathcal{L},\underline{\boldsymbol{\theta}}\Big\}. \qquad (5.56)
\end{aligned}
$$

For notation simplicity, we will remove the symbol ($'$) in Eq. 5.56 and ignore the constants that do not depend on the model parameters. Then, Eq. 5.56 is written as

$$
\begin{aligned}
Q(\underline{\boldsymbol{\theta}}) &= \sum_{ijk}\langle y_{ijk}|\mathcal{L}\rangle\left[-\log\sigma_k-\frac{1}{2}\sigma_k^{-2}(\ell_{ij}-\mu_k)^2+\log\pi_k\right] \\
&\quad+\sum_{ijk}\Big\langle y_{ijk}\Big[-\frac{1}{2}\log|\boldsymbol{\Sigma}_k| \\
&\quad-\frac{1}{2}(\mathbf{x}_{ij}-\mathbf{m}_k-\mathbf{V}_k\langle\mathbf{z}_i|\mathcal{X}\rangle)^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_{ij}-\mathbf{m}_k-\mathbf{V}_k\langle\mathbf{z}_i|\mathcal{X}\rangle)\Big]\Big\rangle \\
&\quad-\frac{1}{2}\sum_{ijk}\left\langle y_{ijk}\mathbf{z}_i{}^{\mathsf{T}}\mathbf{z}_i|\mathcal{X},\mathcal{L}\right\rangle \\
&= \sum_{ijk}\langle y_{ijk}|\mathcal{L}\rangle\left[-\log\sigma_k-\frac{1}{2}\sigma_k^{-2}(\ell_{ij}-\mu_k)^2+\log\pi_k\right] \\
&\quad+\sum_{ijk}\langle y_{ijk}|\mathcal{L}\rangle\left[-\frac{1}{2}\log|\boldsymbol{\Sigma}_k|-\frac{1}{2}(\mathbf{x}_{ij}-\mathbf{m}_k)^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_{ij}-\mathbf{m}_k)\right] \\
&\quad+\sum_{ijk}(\mathbf{x}_{ij}-\mathbf{m}_k)^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}\mathbf{V}_k\left\langle y_{ijk}\mathbf{z}_i|\mathcal{X},\mathcal{L}\right\rangle \\
&\quad-\frac{1}{2}\left[\sum_{ijk}\left\langle y_{ijk}\mathbf{z}_i^{\mathsf{T}}\mathbf{V}_k^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}\mathbf{V}_k\mathbf{z}_i|\mathcal{X},\mathcal{L}\right\rangle+\left\langle y_{ijk}\mathbf{z}_i^{\mathsf{T}}\mathbf{z}_i|\mathcal{X},\mathcal{L}\right\rangle\right] \\
&= \sum_{ijk}\langle y_{ijk}|\mathcal{L}\rangle\left[-\log\sigma_k-\frac{1}{2}\sigma_k^{-2}(\ell_{ij}-\mu_k)^2+\log\pi_k\right] \\
&\quad+\sum_{ijk}\langle y_{ijk}|\mathcal{L}\rangle\left[-\frac{1}{2}\log|\boldsymbol{\Sigma}_k|-\frac{1}{2}(\mathbf{x}_{ij}-\mathbf{m}_k)^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_{ij}-\mathbf{m}_k)\right] \\
&\quad+\sum_{ijk}(\mathbf{x}_{ij}-\mathbf{m}_k)^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}\mathbf{V}_k\left\langle y_{ijk}\mathbf{z}_i|\mathcal{X},\mathcal{L}\right\rangle \\
&\quad-\frac{1}{2}\left[\sum_{ijk}\mathrm{tr}\left\{\left(\mathbf{V}_k^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}\mathbf{V}_k+\mathbf{I}\right)\left\langle y_{ijk}\mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X},\mathcal{L}\right\rangle\right\}\right], \qquad (5.57)
\end{aligned}
$$

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

where we have used the identity

$$\left\langle \mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x}\right\rangle = \mathrm{tr}\left\{\mathbf{A}\left[\left\langle \mathbf{x}\mathbf{x}^{T}\right\rangle - \langle\mathbf{x}\rangle\langle\mathbf{x}^{\mathsf{T}}\rangle\right]\right\} + \langle\mathbf{x}^{\mathsf{T}}\rangle\mathbf{A}\langle\mathbf{x}\rangle.$$

The derivation of SD-mPLDA parameters is almost the same as that of SI-mPLDA in Section 5.4.1, except for the parameters of the SNR model. Specifically, to compute μ_k and σ_k, we set $\frac{\partial Q}{\partial \mu_k} = 0$ and $\frac{\partial Q}{\partial \sigma_k} = 0$, which result in

$$\mu_k = \frac{\sum_{ij}\langle y_{ijk}|\mathcal{L}\rangle \ell_{ij}}{\sum_{ij}\langle y_{ij}|\mathcal{L}\rangle} \quad \text{and} \quad \sigma_k^2 = \frac{\sum_{ij}\langle y_{ijk}|\mathcal{L}\rangle(\ell_{ij}-\mu_k)^2}{\sum_{ij}\langle y_{ijk}|\mathcal{L}\rangle}.$$

E-Step: Computing Posterior Expectations

The E-step is almost the same as the "Sharing Latent Factors" in Section 5.4.1, except for the additional observations $\mathcal{L}$ (the SNR of utterances). Let $\mathcal{X}_i$ and $\mathcal{L}_i$ be the i-vectors and SNR of utterances from the ith speaker, respectively. We begin with the joint posterior density:

$$\begin{aligned}
p(\mathbf{z}_i, y_{i\cdot\cdot}|\mathcal{X}_i,\mathcal{L}_i) &\propto p(\mathcal{X}_i,\mathcal{L}_i|\mathbf{z}_i, y_{i\cdot\cdot}=1)p(\mathbf{z}_i, y_{i\cdot\cdot}) \\
&= p(\mathcal{X}_i|\mathbf{z}_i, y_{i\cdot\cdot}=1)p(\mathcal{L}_i|y_{i\cdot\cdot}=1)p(y_{i\cdot\cdot})p(\mathbf{z}_i) \quad \because \mathbf{z}_i \text{ and } y_{i\cdot\cdot} \text{ are independent} \\
&= \prod_{j=1}^{H_i}\prod_{k=1}^{K}\left[\pi_k p(\mathbf{x}_{ij}|y_{ijk}=1,\mathbf{z}_i)p(\ell_{ij}|y_{ijk}=1)\right]^{y_{ijk}} p(\mathbf{z}_i) \qquad \text{[1, Eq. 9.38]} \qquad (5.58) \\
&= \underbrace{p(\mathbf{z}_i)\left\{\prod_{j=1}^{H_i}\prod_{k=1}^{K}\left[\mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}_k+\mathbf{V}_k\mathbf{z}_i,\boldsymbol{\Sigma}_k)\right]^{y_{ijk}}\right\}}_{\propto\, p(\mathbf{z}_i|\mathcal{X}_i)}\left\{\prod_{j=1}^{H_i}\prod_{k=1}^{K}\left[\pi_k\mathcal{N}(\ell_{ij}|\mu_k,\sigma_k^2)\right]^{y_{ijk}}\right\}, \qquad (5.59)
\end{aligned}$$

where we have used the property that y_{ijk} is governed by ℓ_{ij}. Notice that the posterior density of $\mathbf{z}_i$ has the identical form as Eq. 5.48. The only dissimilarity is the posterior of y_{ijk}, which can be estimated by the Bayes rule:

$$\begin{aligned}
\langle y_{ijk}|\mathcal{L}\rangle &= \langle y_{ijk}|\ell_{ij}\rangle \\
&= P(y_{ijk}=1|\ell_{ij},\underline{\boldsymbol{\lambda}}) \\
&= \frac{P(y_{ijk}=1)p(\ell_{ij}|y_{ijk}=1,\underline{\boldsymbol{\lambda}})}{\sum_{r=1}^{K}P(y_{ijr}=1)p(\ell_{ij}|y_{ijr}=1,\underline{\boldsymbol{\lambda}})} \\
&= \frac{\pi_k\mathcal{N}(\ell_{ij}|\mu_k,\sigma_k^2)}{\sum_{r=1}^{K}\pi_r\mathcal{N}(\ell_{ij}|\mu_r,\sigma_r^2)}.
\end{aligned} \qquad (5.60)$$

Then, the joint posterior expectations are given by

$$\begin{aligned}
\langle y_{ijk}\mathbf{z}_i|\mathcal{X}_i,\mathcal{L}_i\rangle &= \langle y_{ijk}|\ell_{ij}\rangle\langle\mathbf{z}_i|\mathcal{X}_i\rangle \\
\langle y_{ijk}\mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X}_i,\mathcal{L}_i\rangle &= \langle y_{ijk}|\ell_{ij}\rangle\langle\mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X}_i\rangle.
\end{aligned} \qquad (5.61)$$

To make sure that the clustering process is guided by the SNR of utterances instead of their i-vectors, we assume that y_{ijk} is posteriorly independent of $\mathbf{x}_{ij}$, i.e., $\langle y_{ijk}|\ell_{ij},\mathbf{x}_{ij}\rangle =$

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

$\langle y_{ijk}|\ell_{ij}\rangle$. This assumption causes the alignments of i-vectors depend on the SNR of utterances only.

In summary, the EM steps of SI-mPLDA are given below:

E-Step:

$$\mathbb{E}_{\mathcal{Y}}\left\{y_{ijk}=1|\mathcal{L},\underline{\lambda}\right\}\equiv\langle y_{ijk}|\mathcal{L}\rangle=\frac{\pi_k\mathcal{N}(\ell_{ij}|\mu_k,\sigma_k^2)}{\sum_{r=1}^{K}\pi_r\mathcal{N}(\ell_{ij}|\mu_r,\sigma_r^2)} \tag{5.62a}$$

$$\mathbf{L}_i=\mathbf{I}+\sum_{k=1}^{K}H_{ik}\mathbf{V}_k^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}\mathbf{V}_k \tag{5.62b}$$

$$\langle y_{ijk}\mathbf{z}_i|\mathcal{X},\mathcal{L}\rangle=\langle y_{ijk}|\mathcal{L}\rangle\langle\mathbf{z}_i|\mathcal{X}\rangle \tag{5.62c}$$

$$\langle\mathbf{z}_i|\mathcal{X}\rangle=\mathbf{L}_i^{-1}\sum_{k=1}^{K}\sum_{j\in\mathcal{H}_{ik}}\mathbf{V}_k^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_{ij}-\mathbf{m}_k) \tag{5.62d}$$

$$\langle y_{ijk}\mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X},\mathcal{L}\rangle=\langle y_{ijk}|\mathcal{L}\rangle\langle\mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X}\rangle \tag{5.62e}$$

$$\langle\mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X}\rangle=\mathbf{L}_i^{-1}+\langle\mathbf{z}_i|\mathcal{X}\rangle\langle\mathbf{z}_i|\mathcal{X}\rangle^{\mathsf{T}} \tag{5.62f}$$

M-Step:

$$\pi_k'=\frac{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\langle y_{ijk}|\mathcal{L}\rangle}{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\sum_{l=1}^{K}\langle y_{ijl}|\mathcal{L}\rangle} \tag{5.63a}$$

$$\mu_k'=\frac{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\langle y_{ijk}|\mathcal{L}\rangle\ell_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\langle y_{ijk}|\mathcal{L}\rangle} \tag{5.63b}$$

$$\sigma_k^{2\prime}=\frac{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\langle y_{ijk}|\mathcal{L}\rangle(\ell_{ij}-\mu_k')^2}{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\langle y_{ijk}|\mathcal{L}\rangle} \tag{5.63c}$$

$$\mathbf{m}_k'=\frac{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\langle y_{ijk}|\mathcal{L}\rangle\mathbf{x}_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\langle y_{ijk}|\mathcal{L}\rangle} \tag{5.63d}$$

$$\mathbf{V}_k'=\left[\sum_{i=1}^{N}\sum_{j=1}^{H_i}\mathbf{f}_{ijk}'\langle y_{ijk}\mathbf{z}_i|\mathcal{X},\mathcal{L}\rangle^{\mathsf{T}}\right]\left[\sum_{i=1}^{N}\sum_{j=1}^{H_i}\left\langle y_{ijk}\mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X},\mathcal{L}\right\rangle\right]^{-1}$$

$$\boldsymbol{\Sigma}_k'=\frac{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\left[\langle y_{ijk}|\mathcal{L}\rangle\mathbf{f}_{ijk}'\mathbf{f}_{ijk}'^{\mathsf{T}}-\mathbf{V}_k'\langle y_{ijk}\mathbf{z}_i|\mathcal{X},\mathcal{L}\rangle\mathbf{f}_{ijk}'^{\mathsf{T}}\right]}{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\langle y_{ijk}|\mathcal{L}\rangle} \tag{5.63e}$$

$$\mathbf{f}_{ijk}'=\mathbf{x}_{ij}-\mathbf{m}_k' \tag{5.63f}$$

Likelihood Ratio Scores

Denote $\mathbf{x}_s$ and $\mathbf{x}_t$ as the i-vectors of the target speaker and the test speaker, respectively. Also denote ℓ_s and ℓ_t as the SNR (in dB) of the corresponding utterances. Then, the same-speaker marginal likelihood is

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

$$
\begin{aligned}
&p(\mathbf{x}_s, \mathbf{x}_t, \ell_s, \ell_t | \text{same-speaker}) \\
&= p(\ell_s) p(\ell_t) p(\mathbf{x}_s, \mathbf{x}_t | \ell_s, \ell_t, \text{same-speaker}) \\
&= p_{st} \sum_{k_s=1}^{K} \sum_{k_t=1}^{K} \int p(\mathbf{x}_s, \mathbf{x}_t, y_{k_s} = 1, y_{k_t} = 1, \mathbf{z} | \underline{\boldsymbol{\theta}}, \ell_s, \ell_t) \mathrm{d}\mathbf{z} \\
&= p_{st} \sum_{k_s=1}^{K} \sum_{k_t=1}^{K} \gamma_{\ell_s, \ell_t}(y_{k_s}, y_{k_t}) \int p(\mathbf{x}_s, \mathbf{x}_t | y_{k_s} = 1, y_{k_t} = 1, \mathbf{z}, \underline{\boldsymbol{\omega}}) p(\mathbf{z}) \mathrm{d}\mathbf{z} \\
&= p_{st} \sum_{k_s=1}^{K} \sum_{k_t=1}^{K} \gamma_{\ell_s, \ell_t}(y_{k_s}, y_{k_t}) \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_s^{\mathsf{T}} & \mathbf{x}_t^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \middle| \begin{bmatrix} \mathbf{m}_{k_s}^{\mathsf{T}} & \mathbf{m}_{k_t}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}, \hat{\mathbf{V}}_{k_s k_t} \hat{\mathbf{V}}_{k_s k_t}^{\mathsf{T}} + \hat{\boldsymbol{\Sigma}}_k \right)
\end{aligned}
$$

where $p_{st} = p(\ell_s)p(\ell_t)$, $\hat{\mathbf{V}}_{k_s k_t} = [\mathbf{V}_{k_s}^{\mathsf{T}} \;\; \mathbf{V}_{k_t}^{\mathsf{T}}]^{\mathsf{T}}$, $\hat{\boldsymbol{\Sigma}}_k = \text{diag}\{\boldsymbol{\Sigma}_{k_s}, \boldsymbol{\Sigma}_{k_t}\}$ and

$$
\begin{aligned}
&\gamma_{\ell_s, \ell_t}(y_{k_s}, y_{k_t}) \equiv P(y_{k_s} = 1, y_{k_t} = 1 | \ell_s, \ell_t, \underline{\boldsymbol{\lambda}}) \\
&= \frac{\pi_{k_s} \pi_{k_t} \mathcal{N}([\ell_s \;\; \ell_t]^{\mathsf{T}} | [\mu_{k_s} \;\; \mu_{k_t}]^{\mathsf{T}}, \text{diag}\{\sigma_{k_s}^2, \sigma_{k_t}^2\})}{\sum_{k_s'=1}^{K} \sum_{k_t'=1}^{K} \pi_{k_s'} \pi_{k_t'} \mathcal{N}([\ell_s \;\; \ell_t]^{\mathsf{T}} | [\mu_{k_s'} \;\; \mu_{k_t'}]^{\mathsf{T}}, \text{diag}\{\sigma_{k_s'}^2, \sigma_{k_t'}^2\})}.
\end{aligned}
$$

Likewise, the different-speaker marginal likelihood is

$$
p(\mathbf{x}_s, \mathbf{x}_t, \ell_s, \ell_t | \text{different-speaker}) = p(\mathbf{x}_s, \ell_s | \text{Spk } s) p(\mathbf{x}_t, \ell_t | \text{Spk } t), \qquad \text{Spk } s \neq \text{Spk } t,
$$

where

$$
\begin{aligned}
p(\mathbf{x}_s, \ell_s | \text{Spk } s) &= p(\ell_s) \sum\nolimits_{k_s=1}^{K} \int p(\mathbf{x}_s, y_{k_s} = 1, \mathbf{z} | \underline{\boldsymbol{\theta}}, \ell_s) \mathrm{d}\mathbf{z} \\
&= p(\ell_s) \sum\nolimits_{k_s=1}^{K} \gamma_{\ell_s}(y_{k_s}) \mathcal{N}\left(\mathbf{x}_s \middle| \mathbf{m}_{k_s}, \mathbf{V}_{k_s} \mathbf{V}_{k_s}^{\mathsf{T}} + \boldsymbol{\Sigma}_{k_s} \right),
\end{aligned}
$$

and similarly for $p(\mathbf{x}_t, \ell_t | \text{Spk } t)$. Therefore, the likelihood ratio $S_{\text{mLR-SNR}}$ is given by:

$$
\begin{aligned}
&S_{\text{mLR-SNR}}(\mathbf{x}_s, \mathbf{x}_t) = \\
&\frac{\sum_{k_s=1}^{K} \sum_{k_t=1}^{K} \gamma_{\ell_s, \ell_t}(y_{k_s}, y_{k_t}) \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_s^{\mathsf{T}} & \mathbf{x}_t^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \middle| \begin{bmatrix} \mathbf{m}_{k_s}^{\mathsf{T}} & \mathbf{m}_{k_t}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}, \hat{\mathbf{V}}_{k_s k_t} \hat{\mathbf{V}}_{k_s k_t}^{\mathsf{T}} + \hat{\boldsymbol{\Sigma}}_{k_s k_t} \right)}{\left[\sum_{k_s=1}^{K} \gamma_{\ell_s}(y_{k_s}) \mathcal{N}\left(\mathbf{x}_s \middle| \mathbf{m}_{k_s}, \mathbf{V}_{k_s} \mathbf{V}_{k_s}^{\mathsf{T}} + \boldsymbol{\Sigma}_{k_s} \right) \right] \left[\sum_{k_t=1}^{K} \gamma_{\ell_t}(y_{k_t}) \mathcal{N}\left(\mathbf{x}_t \middle| \mathbf{m}_{k_t}, \mathbf{V}_{k_t} \mathbf{V}_{k_t}^{\mathsf{T}} + \boldsymbol{\Sigma}_{k_t} \right) \right]}
\end{aligned}
\tag{5.64}
$$

Because the determinant of $\hat{\mathbf{V}}_{k_s} \hat{\mathbf{V}}_{k_s}^{\mathsf{T}} + \hat{\boldsymbol{\Sigma}}_{k_s}$ could exceed the double-precision representation, direct evaluations of Eq. 5.64 could cause numerical problems. However, the problems can be overcome by noting the identity: $|\alpha \mathbf{A}| = \alpha^D |\mathbf{A}|$ where α is a scalar and $\mathbf{A}$ is a $D \times D$ matrix. Thus, we can rewrite Eq. 5.64 as

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

$$S_{\text{mLR-SNR}}(\mathbf{x}_s, \mathbf{x}_t) = \frac{\sum_{k_s=1}^{K}\sum_{k_t=1}^{K}\gamma_{\ell_s,\ell_t}(y_{k_s}, y_{k_t})e^{\left\{-\frac{1}{2}\log|\alpha\hat{\mathbf{\Lambda}}_{k_sk_t}|-\frac{1}{2}\mathcal{D}\left(\left[\mathbf{x}_s^\mathsf{T}\ \mathbf{x}_t^\mathsf{T}\right]^\mathsf{T}\|\left[\mathbf{m}_{k_s}^\mathsf{T}\ \mathbf{m}_{k_t}^\mathsf{T}\right]^\mathsf{T}\right)\right\}}}{\left[\sum_{k_s=1}^{K}\gamma_{\ell_s}(y_{k_s})e^{\left\{-\frac{1}{2}\log|\alpha\mathbf{\Lambda}_{k_s}|-\frac{1}{2}\mathcal{D}(\mathbf{x}_s\|\mathbf{m}_{k_s})\right\}}\right]\left[\sum_{k_t=1}^{K}\gamma_{\ell_t}(y_{k_t})e^{\left\{-\frac{1}{2}\log|\alpha\mathbf{\Lambda}_{k_t}|-\frac{1}{2}\mathcal{D}(\mathbf{x}_t\|\mathbf{m}_{k_t})\right\}}\right]}, \tag{5.65}$$

where $\hat{\mathbf{\Lambda}}_{k_sk_t} = \hat{\mathbf{V}}_{k_s}\hat{\mathbf{V}}_{k_t}^\mathsf{T} + \hat{\mathbf{\Sigma}}_{k_sk_t}$, $\mathbf{\Lambda}_{k_s} = \mathbf{V}_{k_s}\mathbf{V}_{k_s}^\mathsf{T} + \mathbf{\Sigma}_{k_s}$, $\hat{\mathbf{\Sigma}}_{k_sk_t} = \text{diag}\{\mathbf{\Sigma}_{k_s}, \mathbf{\Sigma}_{k_t}\}$, and $\mathcal{D}(\mathbf{x}\|\mathbf{y})$ is the Mahalanobis distance between $\mathbf{x}$ and $\mathbf{y}$, $\mathcal{D}(\mathbf{x}\|\mathbf{y}) = (\mathbf{x}-\mathbf{y})^\mathsf{T}\mathbf{S}^{-1}(\mathbf{x}-\mathbf{y})$, where $\mathbf{S} = \text{cov}(\mathbf{x}, \mathbf{x})$. In [171], α was set to 5.

5.4.3 DNN-Driven Mixture of PLDA

The DNN-driven mixture of PLDA [95, 176] is an extension of SNR-dependent mixture of PLDA in Section 5.4.2. The main idea is to compute the posteriors of mixture components by using an SNR-aware DNN instead of using a one-D SNR-GMM model. Figure 5.12 shows the difference between the scoring process of (a) SNR-dependent mixture of PLDA and (b) DNN-driven mixture of PLDA. In the former, an SNR estimator is applied to estimate the SNR of the target utterance and the test utterance. The posterior probabilities $\gamma_\ell(y_k)$ of mixture components depend purely on the SNR ℓ:

$$\gamma_\ell(y_k) \equiv P(y_k = 1|\ell, \breve{}) = \frac{\pi_k\mathcal{N}(\ell|\mu_k, \sigma_k^2)}{\sum_{k'=1}^{K}\pi_{k'}\mathcal{N}(\ell|\mu_{k'}, \sigma_{k'}^2)}, \tag{5.66}$$

where $\{\pi_k, \mu_k, \sigma_k^2\}_{k=1}^K$ are the parameters of the one-D SNR-GMM model. For the latter, an SNR-aware DNN is trained to produce the posteriors of SNR $\gamma_\mathbf{x}(y_k)$ based on the input i-vector $\mathbf{x}$:

$$\gamma_\mathbf{x}(y_k) \equiv P(y_k = 1|\mathbf{x}, \mathbf{w}), \tag{5.67}$$

where $\mathbf{w}$ denotes the weights of the SNR-aware DNN. Plugging these mixture posteriors into the mixture of PLDA scoring function in Eq. 5.65, we have

$$S_{\text{DNN-mPLDA}}(\mathbf{x}_s, \mathbf{x}_t) = \frac{\sum_{k_s=1}^{K}\sum_{k_t=1}^{K}\gamma_{\mathbf{x}_s}(y_{k_s})\gamma_{\mathbf{x}_t}(y_{k_t})e^{\left\{-\frac{1}{2}\log|\alpha\hat{\mathbf{\Lambda}}_{k_sk_t}|-\frac{1}{2}\mathcal{D}\left(\left[\mathbf{x}_s^\mathsf{T}\ \mathbf{x}_t^\mathsf{T}\right]^\mathsf{T}\|\left[\mathbf{m}_{k_s}^\mathsf{T}\ \mathbf{m}_{k_t}^\mathsf{T}\right]^\mathsf{T}\right)\right\}}}{\left[\sum_{k_s=1}^{K}\gamma_{\mathbf{x}_s}(y_{k_s})e^{\left\{-\frac{1}{2}\log|\alpha\mathbf{\Lambda}_{k_s}|-\frac{1}{2}\mathcal{D}(\mathbf{x}_s\|\mathbf{m}_{k_s})\right\}}\right]\left[\sum_{k_t=1}^{K}\gamma_{\mathbf{x}_t}(y_{k_t})e^{\left\{-\frac{1}{2}\log|\alpha\mathbf{\Lambda}_{k_t}|-\frac{1}{2}\mathcal{D}(\mathbf{x}_t\|\mathbf{m}_{k_t})\right\}}\right]}, \tag{5.68}$$

Figure 5.13(a) shows the graphical model of SNR-dependent mixture of PLDA with parameters $\{\pi_k, \mu_k, \sigma_k, \mathbf{m}_k, \mathbf{V}_k, \mathbf{\Sigma}_k\}_{k=1}^K$. In the diagram, $\underline{\pi} = \{\pi_k\}_{k=1}^K$, $\underline{\mu} = \{\mu_k\}_{k=1}^K$, $\underline{\sigma} = \{\sigma_k\}_{k=1}^K$, $\underline{\mathbf{m}} = \{\mathbf{m}_k\}_{k=1}^K$, $\underline{\mathbf{V}} = \{\mathbf{V}_k\}_{k=1}^K$, $\underline{\mathbf{\Sigma}} = \{\mathbf{\Sigma}_k\}_{k=1}^K$. Figure 5.13(b) shows the graphical model of DNN-driven mixture of PLDA with parameters $\{\mathbf{m}_k, \mathbf{V}_k, \mathbf{\Sigma}_k\}_{k=1}^K$.

The key advantage of the DNN-driven mixture of PLDA is that it uses a classifier to guide the PLDA training, such that each i-vector cluster is modeled by one mixture component. During testing, we combine the PLDA scores using dynamic weights that depend on the classifier's output. This approach is flexible because there is no restriction

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

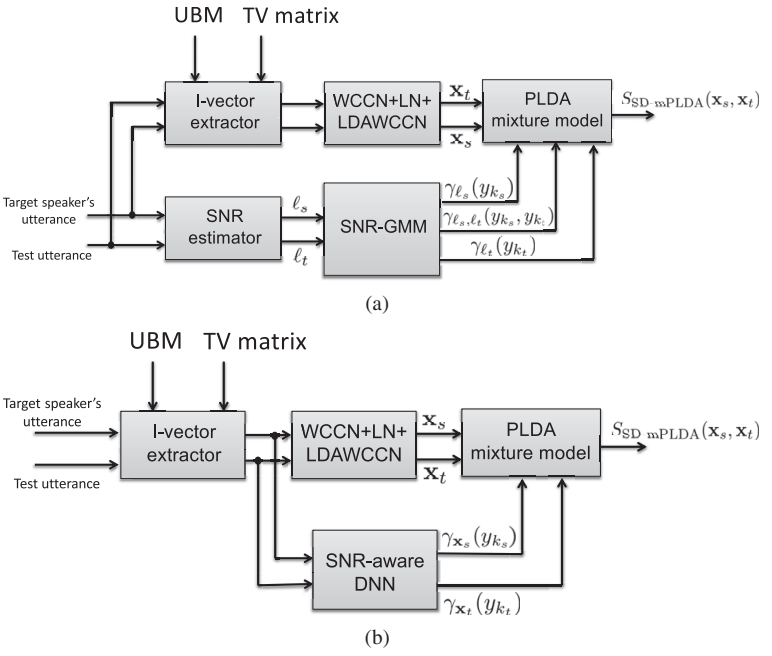


Figure 5.12 (a) Procedure of computing the score of two utterances with SNR ℓ_s and ℓ_t in SNR-dependent mixture of PLDA. (b) Same as (a) but replacing the SNR-dependent mixture of PLDA by DNN-driven mixture of PLDA.

on the type of classifiers to be used. In fact, any classifier, such as DNN, SVM, and logistic regression, can be used as long as they can leverage the cluster property in the training data. However, experimental results suggest that DNN classifiers give the best performance [95].

5.5 Multi-Task DNN for Score Calibration

Because adverse acoustic conditions and duration variability in utterances could change the distribution of PLDA scores, a number of back ends have been investigated to replace the PLDA models. These back ends include support vector machines (SVMs) [177] and end-to-end models [178].

Rather than enhancing the PLDA back end, score calibration methods have been proposed to compensate for the harmful effect on the PLDA scores due to background noise and duration variability. For example, the score shifts in in [167, 179, 180] are deterministic and the shifts are assumed to be linearly related to the utterances' SNR and/or to the logarithm of utterance duration. In [181], the authors assume that the shift follows a Gaussian distribution with quality-dependent mean and variance. A Bayesian network was used to infer the posterior distributions of the target and nontarget hypotheses; a calibrated likelihood-ratio is then computed from the posterior distributions. On

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

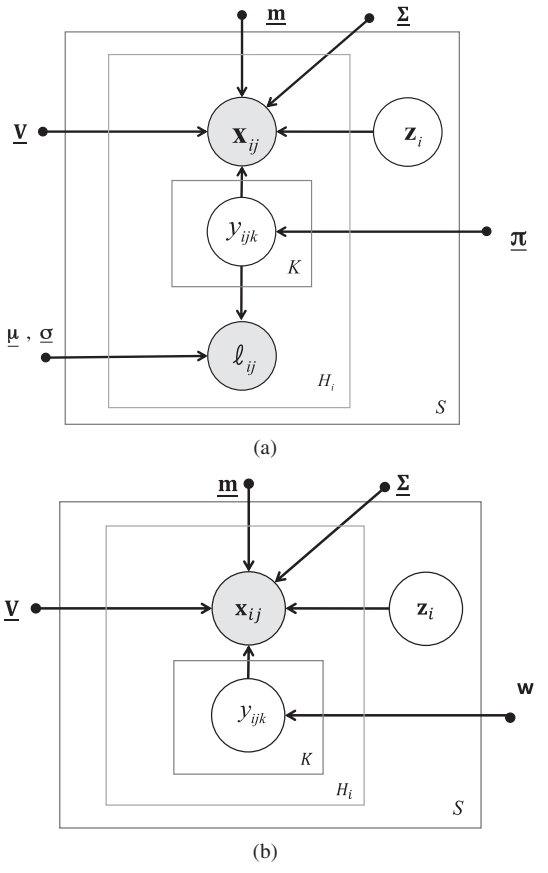


Figure 5.13 The graphical model of (a) SNR-dependent mixture of PLDA and (b) DNN-driven mixture of PLDA. [Reprinted with permission from *DNN-driven Mixture of PLDA for Robust Speaker Verification (Figure 2), N. Li, M.W. Mak, and J.T. Chien., IEEE/ACM Trans. on Audio Speech and Language Processing, vol. 25, no. 6, pp. 1371–1383, June 2017, with permission of IEEE*]

the other hand, the score shift in [182, 183] was assumed to be bilinear or considered as a cosine-distance function of the two quality vectors derived from the target-speaker and test i-vectors.

There are methods that compensate for the duration mismatch only [167, 184, 185] or both duration and SNR mismatch [179–182]. A common property of these methods is that all of them assume that the detrimental effect can be compensated by shifting the PLDA scores. These methods aim to estimate the shift based on some meta data, including duration and SNR [179, 180]), or from the i-vectors [182] to counteract the effect. Despite these methods use linear models, they have demonstrated improvement in performance in a number of systems. However, score shift may not be linearly related to SNR and log-duration. As a result, cosine-distance scores and bilinear transformation may not be able to capture the true relationship accurately.

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

5.5.1 Quality Measure Functions

A quality measure function (QMF) maps some measurable quantities, such as SNR and duration of utterances, to score shifts that represent the detrimental effect of background noise and duration variability on the PLDA scores [167, 179, 180]. Denote S as the *uncalibrated* PLDA score of a test utterance and a target speaker utterance. Also denote λ_{tgt} and λ_{tst} as the quality measures of the target speaker and test utterances, respectively. The *calibrated* score S' can then be evaluated according to:

$$S' = w_0 + w_1 S + Q(\lambda_{tgt}, \lambda_{tst}), \tag{5.69}$$

where $Q(\lambda_{tgt}, \lambda_{tst})$ is a QMF. In [179, 180], the QMFs were based on the duration (d_{tst}) and SNR (SNR_{tst}) of the test utterance:

$$\begin{aligned} Q_{\text{SNR}}(\text{SNR}_{tst}) &= w_2 \text{SNR}_{tst} \\ Q_{\text{Dur}}(d_{tst}) &= w_2 \log(d_{tst}) \\ Q_{\text{SNR+Dur}}(\text{SNR}_{tst}, d_{tst}) &= w_2 \text{SNR}_{tst} + w_3 \log(d_{tst}), \end{aligned} \tag{5.70}$$

where w_2 and w_3 are the weights of the respective meta information. In case the effect of noise in the target utterance is also considered, Q_{SNR} becomes:

$$Q_{\text{SNR2}}(\text{SNR}_{tgt}, \text{SNR}_{tst}) = w_2 \text{SNR}_{tgt} + w_3 \text{SNR}_{tst}, \tag{5.71}$$

where SNR_{tgt} is the SNR of the target speaker utterance. In Eqs. 5.69–5.71, the weights w_i, $i = 0, \ldots, 3$, can be estimated by logistic regression [186].

Ferrer et al. [183] and Nautsch et al. [182] derived a quality vector $\mathbf{q}$ based on the posterior probabilities of various acoustic conditions given an i-vector based on the assumption that i-vectors are acoustic-condition dependent. In the method, every i-vector has its corresponding quality vector. Given the i-vectors of a verification trial, the score shift is a function of the quality vectors in that trial.

Nautsch et al. [182] name the function the "function of quality estimate (FQE)." Specifically, i-vectors derived from utterances of 55 combinations of different durations and SNRs were used to train 55 Gaussian models $\Lambda_j = \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}\}_{j=1}^{55}$. These Gaussians have their own mean $\boldsymbol{\mu}_j$ estimated from the i-vectors of the respective conditions; however, they share the same global covariance matrix $\boldsymbol{\Sigma}$. For an i-vector $\mathbf{x}$, the jth component of $\mathbf{q}$ is the posterior of condition j:

$$q_j = \frac{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma})}{\sum_{j'} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{j'}, \boldsymbol{\Sigma})}, \; j = 1, \ldots, 55. \tag{5.72}$$

Given the i-vectors $\mathbf{x}_{tgt}$ and $\mathbf{x}_{tst}$ from a target speaker and a test speaker, respectively, the corresponding quality vectors $\mathbf{q}_{tgt}$ and $\mathbf{q}_{tst}$ are obtained from Eq. 5.72. Then, we can obtain the score shift as follows:

$$\begin{aligned} Q_{\text{UAC}}(\mathbf{q}_{tgt}, \mathbf{q}_{tst}) &= w_2 \mathbf{q}_{tgt}^{\mathsf{T}} \boldsymbol{W} \mathbf{q}_{tst} \\ Q_{\text{qvec}}(\mathbf{q}_{tgt}, \mathbf{q}_{tst}) &= w_2 \cos(\mathbf{q}_{tgt}, \mathbf{q}_{tst}), \end{aligned} \tag{5.73}$$

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

where $\mathbf{W}$ is a symmetric bilinear matrix and $\cos(\mathbf{a}, \mathbf{b})$ means cosine-distance score between vectors $\mathbf{a}$ and $\mathbf{b}$:

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^\intercal \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}.$$

In Eq. 5.73, the components of a quality vector are the posteriors of the respective SNR/duration groups. The cosine-distance and bilinear transformation of the two quality vectors reflect the similarity between the two vectors in terms of SNR and duration. They are similar to the QMFs in Eq. 5.70 and Eq. 5.71 in that the score shift is linearly related to the SNR of the utterances and/or to the logarithm of utterance duration. As will be discussed later, the relationship among score shift, SNR, and duration is rather complex. As a result, the information in SNR and duration is not enough to estimate the ideal score shift. It turns out that i-vectors can provide essential information for estimating the ideal score shift.

Theoretically, the FQE in Eq. 5.73 is better than the QMF in Eq. 5.69 because the former does not use SNR information directly but instead uses it implicitly through the i-vectors and the Gaussian models. Nevertheless, at low SNR, it is unclear whether the cosine distance and bilinear transformation can accurately estimate the score shift. As Figure 5.14 shows, the score shifts and the SNR of utterances have a nonlinear and complex relationship. Because of the noise-level dependence of i-vectors [171] (also see Figure 5.9), it is reasonable to explicitly predict the score shifts from i-vectors instead of implicitly predict them through the Gaussian models of the i-vectors such as the FQE method. In the next section, we explain how the score shifts can be accurately estimated through multi-task deep learning.

5.5.2 DNN-Based Score Calibration

In [3], a DNN-based score calibration algorithm was proposed to mitigate the limitations of the score calibration algorithms described in Section 5.5.1. The key idea is to apply a DNN to determine a suitable score shift for each pair of i-vectors or to estimate the clean PLDA score given a pair of noisy i-vector and a noisy PLDA score. The DNN is trained to carry out *score compensation*, and it plays the same role as function Q in Eq. 5.69. Nevertheless, when the DNN is trained to produce clean PLDA scores, it essentially carries out *score transformation*. For whatever purposes and roles, a further *calibration* process is necessary because there is no guarantee that the DNN can produce true log-likelihood ratios in its output. The authors in [3] collectively refer to the score compensation, transformation, and calibration processes as DNN-based score calibration. Figure 5.15 shows the full process.

In some reports [187, 188], the term *calibration* referred strictly to the process of score adjustment and the adjustment will not change the equal error rate (EER). Here, the terminology in [179, 180, 182] was adopted, i.e., we relax the definition of calibration so that it also includes the processes that could reduce the EER.

The most basic form of DNN-based score calibration is to use a DNN as shown in Figure 5.16 to estimate the appropriate score shift given the target and test i-vector pairs

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

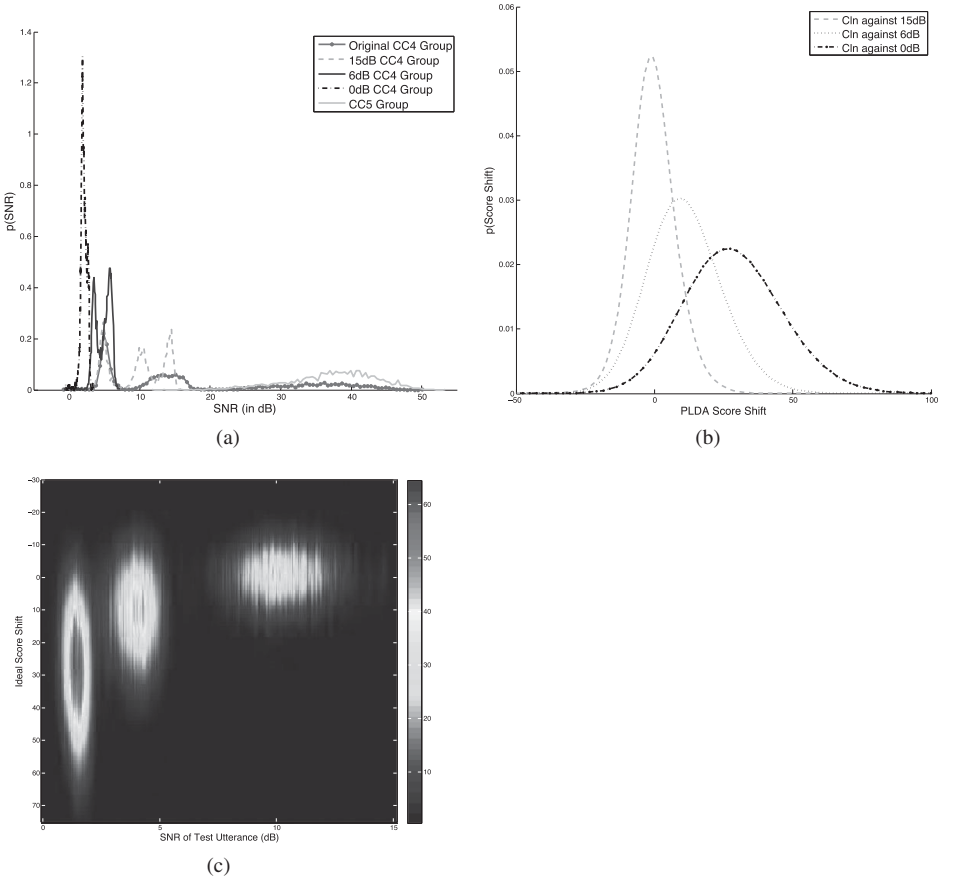


Figure 5.14 Nonlinear relationship between SNRs and socre shifts, and large score shift at low SNR. (a) Distribution of the SNR of the noise-contaminated test utterances and the original utterances in CC4 and CC5 (male) of NIST 2012 SRE. Babble noise was added to to original utterances at SNR of 15dB, 6DB, and 0dB. (b) Distributions of ideal score shifts ($S - S_{cln}$) under three SNR conditions in the test utterances and clean condition in the target speakers' utterances. S_{cln} (clean scores) were obtained by scoring clean test i-vectors against clean i-vectors from target speakers. (c) Distributions of score shifts with respect to test utterances' SNR under clean utterances from target speakers. [Reprinted from *Denoised Senone I-Vectors for Robust Speaker Verification (Figures 6, 8, and 9), Z.L. Tan, M.W. Mak, et al., IEEE/ACM Trans. on Audio Speech and Language Processing, vol. 26, no. 4, pp. 820–830, April 2018, with permission of IEEE*]

($\mathbf{x}_{tgt}$ and $\mathbf{x}_{tst}$) and the uncalibrated PLDA score S. Given an uncalibrated PLDA score S of a verification trial, the compensated score is given by:

$$S'_{\text{st}} = S + \text{DNN}_{\text{st}}(\mathbf{x}_s, \mathbf{x}_t, S), \tag{5.74}$$

where the subscript "st" denotes the output of a *single-task* DNN. With the i-vector pair and the uncalibrated score as input, the DNN outputs the shift of the PLDA score due to the deviation of the acoustic condition from the clean one:

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

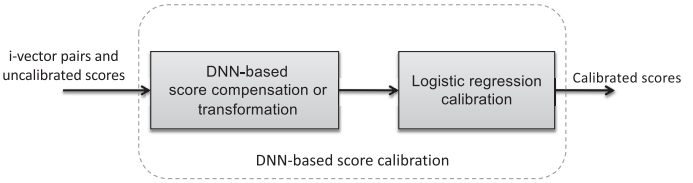


Figure 5.15 DNN-based score calibration.

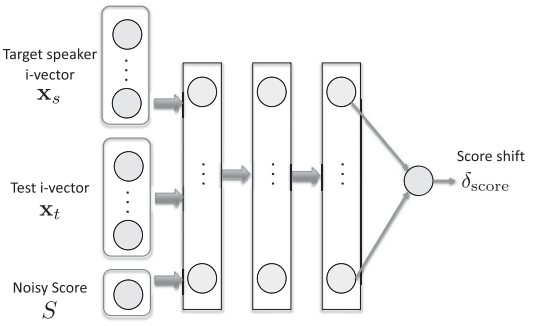


Figure 5.16 The most basic form of DNN-based score calibration in which a DNN predicts the score shift given the target speaker i-vector, the test i-vector, and noisy PLDA score as inputs.

$$\text{DNN}_{\text{st}}(\mathbf{x}_s, \mathbf{x}_t, S) \approx \delta_{\text{score}} = S_{\text{cln}} - S, \tag{5.75}$$

where S_{cln} is the PLDA score if both $\mathbf{x}_s$ and $\mathbf{x}_t$ were derived from clean utterances. Substituting Eq. 5.75 to Eq. 5.74, we have:

$$S'_{\text{st}} \approx S + (S_{\text{cln}} - S) = S_{\text{cln}}. \tag{5.76}$$

Eq. 5.76 suggests that the clean score can be recovered.

The PLDA score of i-vector pair $(\mathbf{x}_s, \mathbf{x}_t)$ is the log-likelihood ratio (Eq. 3.85):

$$\begin{aligned} S &= \text{LLR}(\mathbf{x}_s, \mathbf{x}_t) \\ &= \frac{1}{2}\mathbf{x}_s^{\mathsf{T}} \boldsymbol{Q} \mathbf{x}_t + \frac{1}{2}\mathbf{x}_t^{\mathsf{T}} \boldsymbol{Q} \mathbf{x}_t + \mathbf{x}_s^{\mathsf{T}} \boldsymbol{P} \mathbf{x}_t + \text{const}, \end{aligned} \tag{5.77}$$

where $\boldsymbol{P}$ and $\boldsymbol{Q}$ and are derived from the across-speaker covariances and total covariances of i-vectors. Using Eq. 5.77, the general form of score shift is:

$$\begin{aligned} \delta_{\text{score}} &= \text{LLR}(\mathbf{x}_{s\_\text{cln}}, \mathbf{x}_{t\_\text{cln}}) - \text{LLR}(\mathbf{x}_s, \mathbf{x}_t) \\ &= \frac{1}{2}\mathbf{x}_{s\_\text{cln}}^{\mathsf{T}} \boldsymbol{Q} \mathbf{x}_{s\_\text{cln}} - \frac{1}{2}\mathbf{x}_s^{\mathsf{T}} \boldsymbol{Q} \mathbf{x}_s + \frac{1}{2}\mathbf{x}_{t\_\text{cln}}^{\mathsf{T}} \boldsymbol{Q} \mathbf{x}_{t\_\text{cln}} \\ &\quad - \frac{1}{2}\mathbf{x}_t^{\mathsf{T}} \boldsymbol{Q} \mathbf{x}_t + \mathbf{x}_{s\_\text{cln}}^{\mathsf{T}} \boldsymbol{P} \mathbf{x}_{t\_\text{cln}} - \mathbf{x}_s^{\mathsf{T}} \boldsymbol{P} \mathbf{x}_t. \end{aligned} \tag{5.78}$$

Take notice of the differences between Eq. 5.78 and the bilinear transformation in Eq. 5.73. Specifically, the score shift in Eq. 5.78 involves both of the bilinear transformation of clean and noisy test i-vectors and the bilinear transformation between the target

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

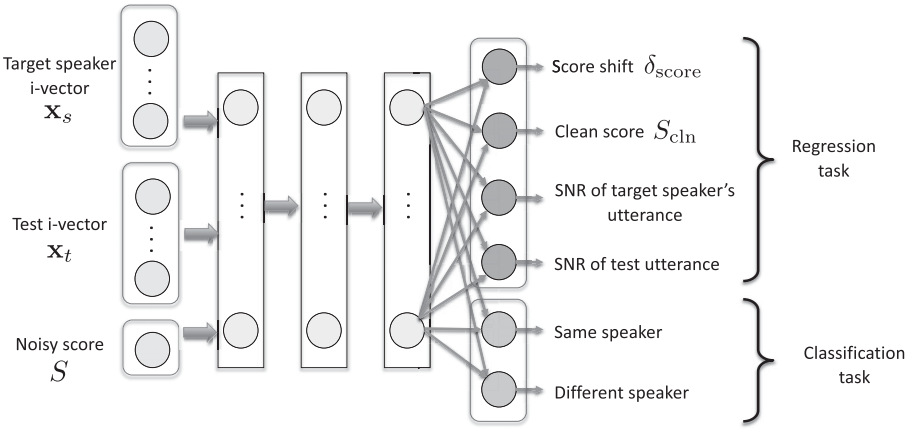


Figure 5.17 Multi-task DNN for score calibration.

speaker and test i-vectors. If the clean test i-vector ($\mathbf{x}_{t\_\text{cln}}$) for every noisy test i-vector ($\mathbf{x}_t$) were known, then Eq. 5.78 can be easily computed without a DNN. However, $\mathbf{x}_{t\_\text{cln}}$ is unknown; therefore, it is necessary for the DNN to learn the complex relationship between the score shifts and the i-vector pairs.

To train the DNN in Figure 5.16, it is necessary to propagate the output node's errors to hundreds of hidden nodes as well as to the input layers. Because there is only one output node, the error propagation is very inefficient. The problem could be solved by introducing some auxiliary tasks for the network to learn, i.e., multi-task learning [189, 190]. The idea is based on the notion that the auxiliary information in the output layer of a multi-task DNN may help to improve the learning efficiency.

In Figure 5.17, a DNN uses multi-task learning to learn two tasks: main and auxiliary. In the main task, the network is trained to produce score shift δ_{score} and clean score S_{cln}, whereas in the auxiliary tasks, the network produces the SNR of target speaker's utterance and the test utterance and outputs the same-speaker and different-speaker posteriors. Both the clean score S_{cln} and ideal score shift δ_{score} are the target outputs of the multi-task DNN.

The regression task uses four output nodes, two of which aim to predict the SNRs of the target speaker's utterance (SNR_s) and the test utterance (SNR_t). They are part of the auxiliary task that helps the network to estimate the score shift. The DNN's regression part uses minimum mean squared error as the optimization criterion and linear activation functions in its output nodes. Another auxiliary task is to enable the network to verify speakers. To this end, we add two classification nodes at the output to indicate whether the input i-vectors pair are from the same speaker or not. The classification part of the network uses softmax outputs and cross-entropy as the optimization criterion.

The multi-task DNN with two classification nodes and four regression nodes comprises two sets of vectors that are concatenated together:

$$\text{DNN}_{\text{mt}}(\mathbf{x}_s, \mathbf{x}_t, S) \approx \big[\underbrace{[\delta_{\text{score}}, S_{\text{cln}}, \text{SNR}_s, \text{SNR}_t]}_{\text{Regression}}, \underbrace{[p^+, p^-]}_{\text{Classification}}\big]^{\intercal}, \tag{5.79}$$

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

where DNN_{mt} denotes the *multi-task* DNN outputs, and p^+ and p^- are the posterior probabilities of same-speaker and different-speaker hypotheses, respectively. Similar to Eq. 5.75, Eq. 5.79 means that the DNN uses the i-vector pair $(\mathbf{x}_s, \mathbf{x}_t)$ and the original score S as input. The multi-task learning strategy enables the network to output the score shift δ_{score}, the clean score S_{cln}, the SNR of target-speaker speech, the SNR of test speech, and the posterior probabilities (p^+ and p^-).

After training, the network can be used for score transformation and calibration. The calibration process will only use the score shift and clean score produced by the multi-task DNN:

$$\text{DNN}_{\text{mt,shift}}(\mathbf{x}_s, \mathbf{x}_t, S) \approx \delta_{\text{score}}, \tag{5.80}$$

and

$$\text{DNN}_{\text{mt,cln}}(\mathbf{x}_s, \mathbf{x}_t, S) \approx S_{\text{cln}}, \tag{5.81}$$

where the subscripts denote the output nodes corresponding to the score shift and clean score in Figure 5.17, respectively. Therefore, we have

$$S'_{\text{mt}} = S + \text{DNN}_{\text{mt,shift}}(\mathbf{x}_s, \mathbf{x}_t, S) \approx S_{cln}, \tag{5.82}$$

and

$$S'_{\text{mt}} = \text{DNN}_{\text{mt,cln}}(\mathbf{x}_s, \mathbf{x}_t, S) \approx S_{cln}. \tag{5.83}$$

5.6 SNR-Invariant Multi-Task DNN

In [191], Yao and Mak analyze how noisy speech affects the distribution of i-vectors (Figures 5.18 and 5.19). The analysis shows that background noise has detrimental effects on intra-speaker variability and that the variability depends on the SNR levels of the utterances. Yao and Mak argue that this SNR-dependent noise effect on the i-vectors is caused by SNR variability. This effect causes difficulty in the back end to separate the speaker and channel variabilities. To address this problem, Yao and Mak proposed using DNNs to suppress both SNR and channel variabilities in the i-vector space directly. To this end, they proposed two models. The first model is called hierarchical regression DNN (H-RDNN). It contains two denoising regression DNNs hierarchically stacked. The second model is called multi-task DNN (MT-DNN). It is trained to carry out speaker classification as well as i-vector denoising (regression).

Yao and Mak observed that i-vectors derived from noise-contaminated speech with similar SNRs tend to form SNR-dependent clusters. This SNR-grouping phenomenon inspires the use of PLDA mixture models in Section 5.4 (see also [95, 171]) so that each SNR group can be handled by a PLDA model. Yao and Mak [191] argued that rather than tackling the i-vectors belonging to the SNR-dependent groups, as in [171], it is better to compensate for the variability directly in the i-vector space.

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

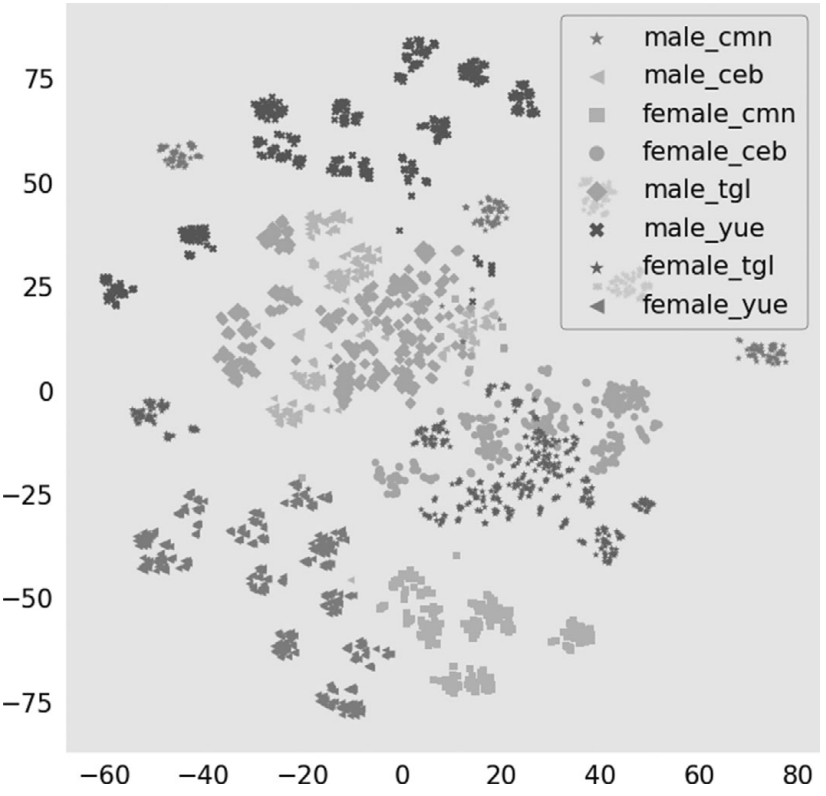


Figure 5.18 Distribution of gender- and language-dependent i-vectors in SRE16 on a two-dimensional t-SNE space. "cmn": Mandarin; "ceb": Cebuano; "yue": Cantonese; "tgl": Tagalog. *It may be better to view the color version of this figure, which is available at https://github.com/enmwmak/ML-for-Spkrec.*

5.6.1 Hierarchical Regression DNN

Figure 5.20 shows the structure of a hierarchical regression DNN (H-RDNN). Denote $\{\mathbf{x}_n\}$ as the i-vectors preprocessed by within-class covariance normalization (WCCN) and length normalization (LN), and $\mathbf{t}_n$ as target i-vectors obtained by averaging speaker-dependent i-vectors derived from clean utterances. Also denote $\mathcal{S} = \{\mathbf{x}_n, \mathbf{t}_n; n = 1, \ldots, N\}$ as a training set comprising N i-vector pairs. Training is divided into two stages. In the first stage, the regression network $f^{reg}_{\mathbf{\Theta}}(\cdot)$ works toward minimizing the MSE and the Frobenius norm of weight paramters:[4]

$$\min_{\mathbf{\Theta}} \frac{1}{N} \sum_{n=1}^{N} \frac{1}{2} ||f^{reg}_{\mathbf{\Theta}}(\mathbf{x}_n) - \mathbf{t}_n||_2^2 + \frac{\beta_{reg_1}}{2} ||\mathbf{\Theta}||_2^2, \tag{5.84}$$

[4] If $\mathbf{t}_n$ and $\mathbf{x}_n$ are derived from a clean utterance and its corresponding noise-contaminated version, Eq. 5.84 leads to the denoising autoencoder (DAE) [3].

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

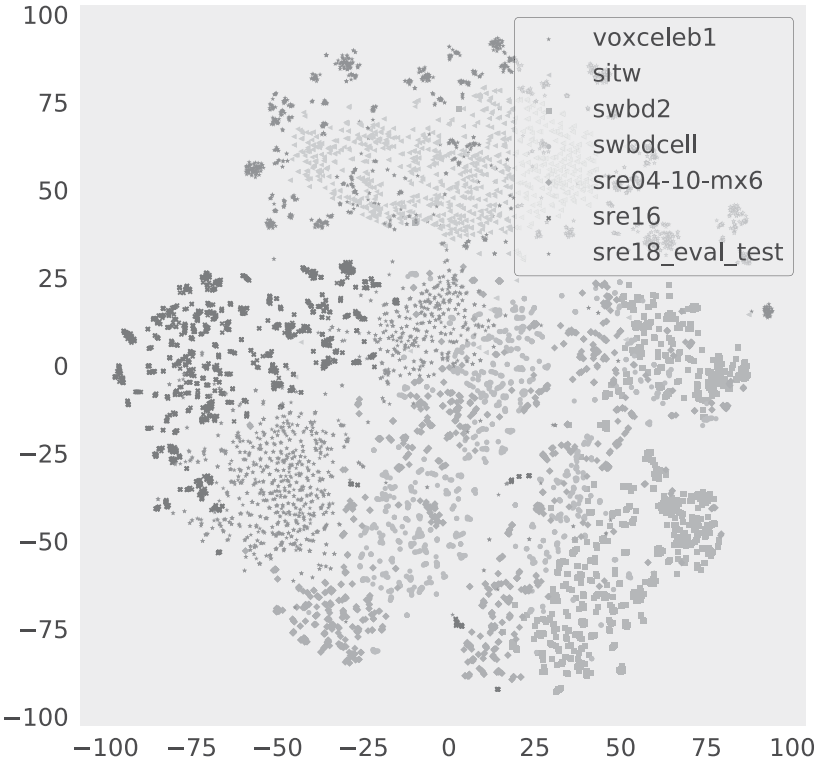


Figure 5.19 Distribution of x-vectors of various datasets on a two-dimensional t-SNE space. *It may be better to view the color version of this figure, which is available at https://github.com/enmwmak/ML-for-Spkrec.*

where $f_{\boldsymbol{\Theta}}^{reg}(\mathbf{x}_n)$ is the output of the top regression layer of the first (left) DNN in Figure 5.20; $\boldsymbol{\Theta}$ denotes the weights in the first regression network and β_{reg_1} controls the degree of regularization. The first regression DNN is trained to suppress channel and SNR variations simultaneously within each speaker cluster.

In the second stage, another regression DNN (the DNN on the right of Figure 5.20) is trained to constrain the outliers that the first DNN cannot properly denoise. Assume that all i-vectors have been processed by the first DNN followed by WCCN whitening and LN. Given a training set comprising N i-vector pairs: $\mathcal{S}' = \{\mathbf{x}'_n, \mathbf{t}'_n; n = 1, \ldots, N\}$, a regularization term and the MSE of the regression network $g_{\boldsymbol{\Phi}}^{reg}(\cdot)$ are minimized jointly:

$$\min_{\boldsymbol{\Phi}} \frac{1}{N} \sum_{n=1}^{N} \frac{1}{2} ||g_{\boldsymbol{\Phi}}^{reg}(\mathbf{x}'_n) - \mathbf{t}'_n||_2^2 + \frac{\beta_{reg_2}}{2} ||\boldsymbol{\Phi}||_2^2, \tag{5.85}$$

where $\mathbf{x}'_n$ is the nth i-vector denoised by the first DNN, i.e., $\mathbf{x}'_n = f_{\boldsymbol{\Theta}}^{reg}(\mathbf{x}_n)$; $\mathbf{t}'_n$ is the associated i-vector obtained from the original i-vector set (without noise contamination) and then denoised by the first DNN, i.e., $\mathbf{t}'_n = f_{\boldsymbol{\Theta}}^{reg}(\mathbf{x}_n^{org})$; $\mathbf{x}''_n = g_{\boldsymbol{\Phi}}^{reg}(\mathbf{x}'_n)$ is the output of

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

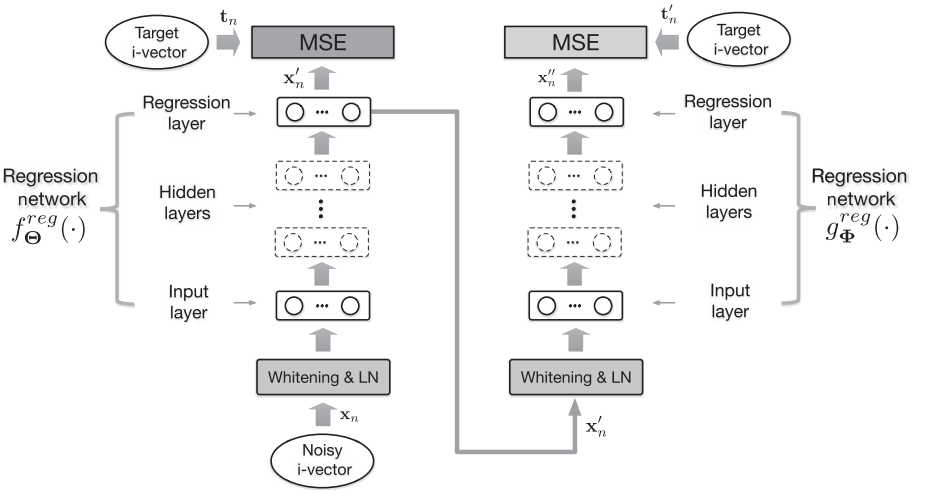


Figure 5.20 Structure of the hierarchical regression DNN (H-RDNN) with arrows denote the flow of data. The network receives noisy i-vector $\mathbf{x}_n$ as input. The regression DNN on the right receives $\mathbf{x}'_n$ as input and produces the final denoised i-vector $\mathbf{x}''_n$ as output. $f^{reg}_{\Theta}(\cdot)$ and $g^{reg}_{\Phi}(\cdot)$ are the mapping functions of the two regression networks, respectively. $\mathbf{t}_n$ and $\mathbf{t}'_n$ are the target i-vectors in Eq. 5.84 and Eq. 5.85, respectively. MSE: mean squared error. [Reprinted from *SNR-Invariant Multi-Task Deep Neural Networks for Robust Speaker Verification (Figure 1), Q. Yao and M.W. Mak, IEEE Signal Processing Letters, vol. 25, no. 11, pp. 1670–1674, Nov. 2018, with permission of IEEE*]

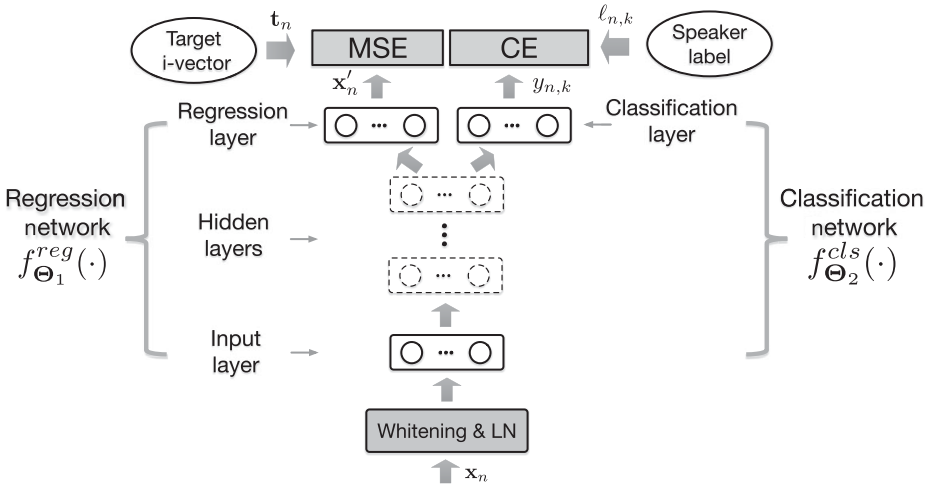


Figure 5.21 The structure of the multi-task DNN (MT-DNN). It receives noisy i-vector $\mathbf{x}_n$ as input and produces $\mathbf{x}'_n$ and $y_{n,k}$ as output. $\mathbf{t}_n$ in Eq. 5.84 and the target label for the classification task $\ell_{n,k}$ in Eq. 5.86 are the targets for the regression and classification tasks, respectively. MSE: mean squared error; CE: cross entropy. [Reprinted from *SNR-Invariant Multi-Task Deep Neural Networks for Robust Speaker Verification (Figure 2), Q. Yao and M.W. Mak, IEEE Signal Processing Letters, vol. 25, no. 11, pp. 1670–1674, Nov. 2018, with permission of IEEE*]

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

the second regression DNN; $\mathbf{\Phi}$ denotes the weights of the second regression DNN and β_{reg_2} controls the degree of regularization.

5.6.2 Multi-Task DNN

The regression task should avoid losing speaker information. To this end, we need the DNN-transformed i-vectors to form a small within-speaker scatter and a large between-speaker scatter. This is realized by a multi-task DNN (MT-DNN) shown in Figure 5.21.

Denote $\boldsymbol{\ell}_n$ as the speaker label in one-hot format of the nth utterance. Also denote $\mathbf{x}_n$ and $\mathbf{t}_n$ as the preprocessed i-vector and the target i-vector, respectively. Assume that we have a training set $\mathcal{S}' = \{\mathbf{x}_n, \mathbf{t}_n, \boldsymbol{\ell}_n; n = 1, \ldots, N\}$ derived from N utterances. To train the regression network $f^{reg}_{\mathbf{\Theta}_1}(\cdot)$ in Figure 5.21, the MSE is minimized in the identical manner as Eq. 5.84. The top regression layer's output is the denoised i-vector $\mathbf{x}'_n$, i.e., $\mathbf{x}'_n = f^{reg}_{\mathbf{\Theta}_1}(\mathbf{x}_n)$. To train the classification network $f^{cls}_{\mathbf{\Theta}_2}(\cdot)$, the cross-entropy (CE) cost together with the Frobenius norm of weights in the classification network are jointly minimized:

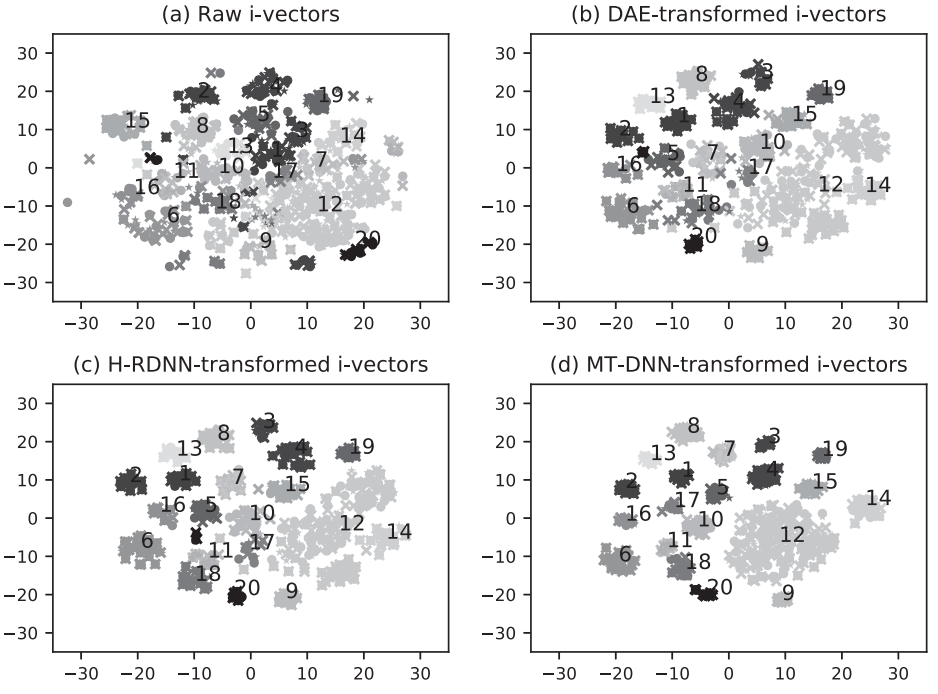


Figure 5.22 T-SNE visualization of 20 speakers from three SNR groups (org+15dB+6dB, telephone speech, babble noise). The colors of the markers represent speakers and the markers with different shapes (○, ×, and ∗) correspond to different SNR groups. *It may be better to view the color version of this figure, which is available at https://github.com/enmwmak/ML-for-Spkrec.* [Reprinted from *SNR-Invariant Multi-Task Deep Neural Networks for Robust Speaker Verification (Figure 3), Q. Yao and M.W. Mak, IEEE Signal Processing Letters, vol. 25, no. 11, pp. 1670–1674, Nov. 2018, with permission of IEEE*]

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

$$\min_{\mathbf{\Theta}_2} -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}\ell_{n,k}\log y_{n,k} + \frac{\beta_{cls}}{2}||\mathbf{\Theta}_2||_2^2. \tag{5.86}$$

In Eq. 5.86, $\ell_{n,k}$ is the kth element of $\boldsymbol{\ell}_n$; if the utterance of $\mathbf{x}_n$ is spoken by speaker k, then $\ell_{n,k} = 1$, otherwise it is equal to 0; $y_{n,k}$ is the posterior probability of the kth speaker (totally K speakers), which is a component of $\mathbf{y}_n$. Note that $\mathbf{y}_n$ is the classification network's output, i.e., $\mathbf{y}_n = f^{cls}_{\mathbf{\Theta}_2}(\mathbf{x}_n)$; $\mathbf{\Theta}_2$ denotes the weights in the classification network and β_{cls} controls the degree of regularization.

The classification and regression tasks are trained in an alternating manner. Specifically, at iteration t, weights are updated according to the gradient of regression loss, and at iteration $t+1$, weights are updated according to the gradient of classification loss. Then, the cycle repeats.

The distributions of 20 speaker clusters are shown in Figure 5.22. These clusters are formed by the raw i-vectors (original, 15 dB, and 6 dB) and the i-vectors transformed by different DNN models. Obviously, the original i-vectors (top-left) could not form

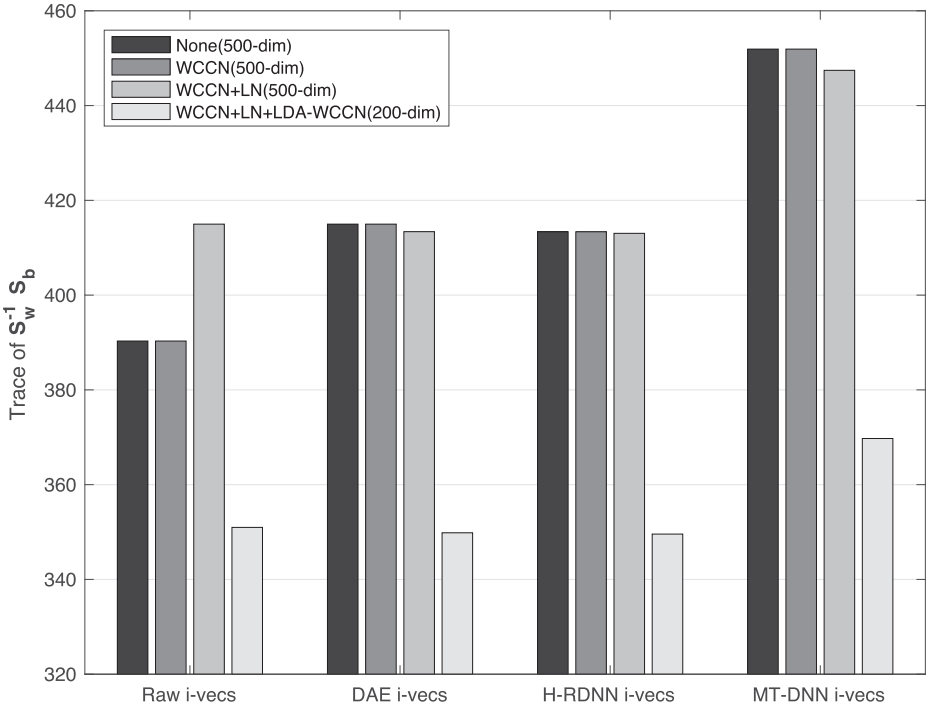


Figure 5.23 Degree of separation among the clusters of 20 speaker from three SNR groups: org+15dB+6dB, telephone speech, and babble noise. The x-axis labels correspond to different types of DNN transformation methods. The vertical axis shows the values of $\text{Tr}(\mathbf{S}_w^{-1}\mathbf{S}_b)$. The gray levels in the legend correspond to different i-vector post-processing methods applied to the DNN-transformed i-vectors. [Reprinted from *SNR-Invariant Multi-Task Deep Neural Networks for Robust Speaker Verification (Figure. 4), Q. Yao and M.W. Mak, IEEE Signal Processing Letters, vol. 25, no. 11, pp. 1670–1674, Nov. 2018, with permission of IEEE*]

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press

distinguishable clusters; on the other hand, the clusters formed by the DNN-transformed i-vectors are very apparent. Consider speaker cluster 6 (darkest shade) in the top-left figure, the left-most • of this speaker drifts from its center significantly. As this i-vector was extracted from an uncontaminated utterance, channel effect is the main reason for the deviation. The 15dB and 6dB i-vectors (marked with crosses and asterisks) have large speaker clusters. After denoising, i-vectors produced by the MT-DNN have the most compact clusters. This suggests that the i-vectors transformed by the MT-DNN are less dependent on the channel and background noise but more dependent on speakers, which is a favorable property for PLDA modeling.

Figure 5.23 shows the trace of $\mathbf{S}_w^{-1}\mathbf{S}_b$, where $\mathbf{S}_b$ and $\mathbf{S}_w$ are the between- and within-speaker scatter matrices obtained from different types of i-vectors (see the labels in the x-axis). These matrices were obtained from the same set of i-vectors used for producing Figure 5.22. The traces measure the dispersion of speaker clusters. The bars indicated by different gray level represent different i-vector post-processing methods. Because $\mathrm{Tr}\{\mathbf{S}_w^{-1}\mathbf{S}_b\}$ will not be affected by WCCN, the value of "None" and "WCCN" are the same. The figure suggest that MT-DNN can produce speaker clusters with the largest degree of separation.

https://doi.org/10.1017/9781108552332.006 Published online by Cambridge University Press