7 Dimension Reduction and Data Augmentation

In general, speaker recognition systems using the i-vectors [16] or x-vectors [156] as speaker features (addressed in Section 3.6) and the probabilistic linear discriminant analysis (PLDA) [17] (addressed in Section 3.5) as a scoring function can achieve good performance. PLDA is known as a linear latent variable model where the same speaker is assumed to be represented by a common factor and the i-vectors or x-vectors of different speakers are characterized by the corresponding latent variables or common factors. This chapter concerns several constraints that affect the performance of i-vectors or x-vectors in PLDA-based speaker recognition. The first constraint is about the insufficiency of speaker identity by using i-vectors, which is caused by the mixing of different factors, e.g., noise, channel, language, gender. In addition, the redundancy with high dimension in i-vectors likely happens. The constraint may be also caused by the insufficient number of training utterances or the imbalanced number of i-vectors among various speakers. The training performance is likely degraded due to the varying lengths of the enrolled sentences. The training of the speaker model may be downgraded by the sparse and heterogeneous utterances in data collection. It is beneficial to deal with the constraints of PLDA training by reducing the feature dimensions, especially in the case of insufficient training sentences [244].

This chapter deals with different constraints in the PLDA model and addresses a series of learning machines or neural networks that are specially designed for dimension reduction as well as data augmentation in speaker recognition systems based on the i-vector combined with PLDA. In particular, we describe how the expanding and emerging researches on a deep learning machine [95] based on the generative adversarial network (GAN) [118], as mentioned in Section 4.6, are developed to construct a subspace model with data augmentation.

Section 7.1 addresses how the objective of neighbor embedding is maximized to derive the low-dimensional representation for high-dimensional observation data based on i-vectors or x-vectors [245, 246]. Subspace mapping is performed in accordance with neighbor embedding. Supervised learning is implemented to assure a low-dimensional representation, which is close under the same speakers and separate between different speakers. To improve the performance of speaker embedding, in Section 7.2, the supervised or discriminative manifold learning [247, 248] is presented to carry out the adversarial learning with threefold advances in speaker recognition. The first idea is to conduct deep learning to characterize the unknown and complicated relation within and between speakers for PLDA speaker recognition. The second thought is

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

to build a latent variable representation [30] for an underlying model structure. The uncertainty of neural network is represented and compensated in a stochastic variant of error back-propagation algorithm by using minibatches of training utterances. The third advanced method is to carry out adversarial learning and manifold learning where an encoder is trained to generate random samples as the prior variables for discriminator. We present the way of merging the adversarial autoencoder [122] in PLDA model for speaker recognition. This solution is implemented as an *adversarial manifold learning* consisting of three components, which are encoder, decoder, and discriminator. The encoder is trained to identify latent representation corresponding to an i-vector, and the decoder is learned to transform the latent variable to reconstruct input i-vector. This reconstruction is used for PLDA scoring [249]. Also, the discriminator is trained to achieve the worst binary classification for judgment between real samples using the PLDA prior and fake data using the generator. We present the approach to this minimax learning problem according to the stochastic optimization using i-vectors in minibatches.

In Section 7.3, we present the approach to data augmentation that can increase the size of training utterances as well as balance the number of training utterances across various speakers. The augmentation of synthesized i-vectors in low-dimensional space is performed. In particular, an *adversarial augmentation learning* is developed to build a neural generative model for PLDA-based speaker recognition. As a result, the diversity of training utterances is increased so as to enhance the robustness in the trained speaker model. Minimax optimization is fulfilled to carry out a two-player game where the generator is optimized to estimate the artificial i-vectors by using random samples from standard Gaussian. These synthesized samples are hardly recognizable from true i-vectors based on the estimated discriminator. The neural networks of generator and discriminator are trained as a distribution model. We improve the system robustness in the presence of changing conditions of i-vectors due to various speakers. Furthermore, the auxiliary classifier GAN [250] is implemented as an extension of conditional GAN [251], which is merged with a class label so that the class conditional likelihood of true i-vectors and synthesized i-vectors are maximized. The speaker-embedded GAN is trained with a fusion of structural information in latent variables. For the sake of speaker recognition, we further minimize the cosine similarity between artificial i-vectors and true i-vectors in the observation level as well in the feature level. A dedicated GAN is formulated and trained in accordance with a multi-objective function where the Gaussian assumption in Gaussian PLDA is met and the reconstruction due to PLDA scoring is assured.

7.1 Variational Manifold PLDA

This section addresses a new variant of PLDA for speaker recognition where the manifold learning is incorporated into the construction of PLDA. A subspace PLDA that

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

carries out low-dimensional speaker recognition is presented. In what follows, we first introduce the subspace method based on the stochastic neighbor embedding.

7.1.1 Stochastic Neighbor Embedding

In [245], an unsupervised learning method called stochastic neighbor embedding (SNE) was proposed to learn the nonlinear manifold in high-dimensional feature space. Assume that $\mathcal{X} = \{\mathbf{x}_n\}$ contains samples in a high-dimensional space. SNE aims to estimate a low-dimensional representations $\mathcal{Z} = \{\mathbf{z}_n\}$ in the feature space, where $\mathbf{z}_n \in \mathbb{R}^d$ preserves the pairwise similarity to $\mathbf{x}_n \in \mathbb{R}^D$ such that $d < D$. The joint probability p_{nm} of two samples $\mathbf{x}_n$ and $\mathbf{x}_m$ can be obtained from a Gaussian distribution

$$p_{nm} = \frac{\exp\left(-\|\mathbf{x}_n - \mathbf{x}_m\|^2\right)}{\sum_s \sum_{t \neq s} \exp\left(-\|\mathbf{x}_s - \mathbf{x}_t\|^2\right)}. \tag{7.1}$$

The probability corresponding to p_{nm} in the low-dimensional representation can also be modeled by a Gaussian using the pairwise similarity among $\{\mathbf{z}_n\}$:

$$q_{nm} = \frac{\exp\left(-\|\mathbf{z}_n - \mathbf{z}_m\|^2\right)}{\sum_s \sum_{t \neq s} \exp\left(-\|\mathbf{z}_s - \mathbf{z}_t\|^2\right)}. \tag{7.2}$$

It is meaningful that p_{nn} and q_{nn} are set to zero in Eqs. 7.1 and 7.2. In [252], a symmetric SNE was realized by minimizing the KL-divergence between two sets of probability densities: $P_n = \{p_{nm}\}_{m=1}^N$ and $Q_n = \{q_{nm}\}_{m=1}^N$, i.e.,

$$\min_{\mathcal{Z}} \sum_n \mathcal{D}_{\mathrm{KL}}(P_n \| Q_n) = \min_{\mathcal{Z}} \sum_n \sum_m p_{nm} \log\left(\frac{p_{nm}}{q_{nm}}\right). \tag{7.3}$$

Nonlinear and nonparametric transformation can be used to preserve the neighbor embeddings of all samples in the original and embedded spaces.

Maaten and Hinton [4] proposed the t-distributed SNE (t-SNE) in which the joint distribution of two samples, $\mathbf{z}_n$ and $\mathbf{z}_m$, in low-dimensional space is modeled by a Student's t-distribution:

$$q_{nm} = \frac{\left(1 + \|\mathbf{z}_n - \mathbf{z}_m\|^2/\nu\right)^{-\frac{\nu+1}{2}}}{\sum_s \sum_{t \neq s} \left(1 + \|\mathbf{z}_s - \mathbf{z}_t\|^2/\nu\right)^{-\frac{\nu+1}{2}}}. \tag{7.4}$$

In Eq. 7.4, ν means the degree of freedom. t-SNE can alleviate the crowding problem in SNE. Figure 7.1 illustrates the concept of distribution construction based on neighbor embeddings in the original data space and a new data space. The front-end processing that uses t-SNE to reduce the dimensionality of i-vectors is performed for subspace speaker recognition using PLDA scoring function [247]. In [249], the PLDA subspace model was constructed by using the variational manifold learning. The stochastic gradient variational Bayesian [114] procedure was implemented to build the latent variable model with neighbor embeddings in PLDA subspace model. The performance of

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

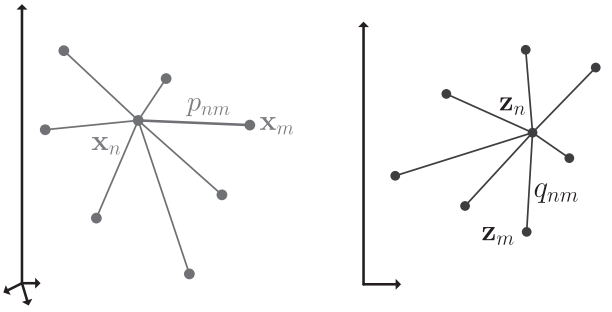

Figure 7.1 Illustration for joint distributions p_{nm} and q_{nm} calculated by neighboring samples centered at $\mathbf{x}_n$ and $\mathbf{z}_n$ in original data space $\mathbf{x}$ (left) and embedded data space $\mathbf{z}$ (right), respectively.

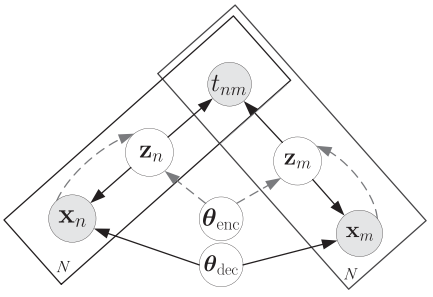

Figure 7.2 Graphical illustration for supervised manifold learning with an encoder (dash line) and a decoder (solid line) given by parameters $\boldsymbol{\theta}_{\text{enc}}$ and $\boldsymbol{\theta}_{\text{dec}}$, respectively. t_{nm} denotes the target value that indicates whether $\mathbf{x}_n$ and $\mathbf{x}_m$ belong to the same class. [Adapted from *Variational Manifold Learning for Speaker Recognition (Figure 2), J.T. Chien and C.W. Hsu, Proceedings International Conference on Acoustics, Speech, and Signal Processing, pp. 4935–4939, 2017, with permission of IEEE*]

speaker recognition based on PLDA model was elevated due to the uncertainty modeling of latent variables.

7.1.2 Variational Manifold Learning

In PLDA-based speaker recognition, the i-vector $\mathbf{x}_n$ in a speaker session is modeled by a latent variable representation based on a latent random vector $\mathbf{z}_n$. We present a neural network variant of a PLDA model where the variational inference is performed. In particular, this section addresses an approach to PLDA subspace representation, which is called the variational manifold PLDA (denoted by VM-PLDA) [249]. This approach aims to hold the property of neighbor embedding for low-dimensional latent representation $\mathbf{z}_n$ which is extracted from the original i-vector $\mathbf{x}_n$. Figure 7.2 illustrates a graphical representation of VM-PLDA. The underlying concept of this model is to implement a variational autoencoder [114] for manifold learning in supervised mode. Encoder

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

and decoder are allocated. As shown in this figure, dash lines show the encoder based on the variational posterior $q(\mathbf{z}_n|\mathbf{x}_n, \boldsymbol{\theta}_{\text{enc}})$ with parameter $\boldsymbol{\theta}_{\text{enc}}$ while solid lines show the decoder based on the generative distribution $p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\theta}_{\text{dec}})$ with parameter $\boldsymbol{\theta}_{\text{dec}}$. Similar to Section 2.3, we apply the approximate or variational inference to estimate the variational parameter $\boldsymbol{\theta}_{\text{enc}}$ in VB-E step and then calculate the model parameters $\boldsymbol{\theta}_{\text{dec}}$ in VB-M step. In the encoder, a Gaussian distribution is used to represent the variational distribution given by the mean vector $\boldsymbol{\mu}(\mathbf{x}_n)$ and the diagonal covariance matrix $\sigma^2(\mathbf{x}_n)\mathbf{I}$

$$q(\mathbf{z}_n|\mathbf{x}_n, \boldsymbol{\theta}_{\text{enc}}) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}(\mathbf{x}_n), \sigma^2(\mathbf{x}_n)\mathbf{I}), \tag{7.5}$$

where i-vector $\mathbf{x}_n$ is used as the input to a fully connected neural network with multiple layers of weights $\boldsymbol{\theta}_{\text{enc}} = \mathbf{W}_{\text{enc}}$ for Gaussian parameters. In the decoder, the likelihood of PLDA model using i-vector $\mathbf{x}_n$ is used for data reconstruction based on the PLDA parameters $\boldsymbol{\theta}_{\text{dec}} = \{\mathbf{m}, \mathbf{V}, \boldsymbol{\Sigma}\}$

$$p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\theta}_{\text{dec}}) = \mathcal{N}(\mathbf{x}_n|\mathbf{m} + \mathbf{V}\mathbf{z}_n, \boldsymbol{\Sigma}). \tag{7.6}$$

More specifically, the neighbor embedding is measured and adopted as a learning objective to conduct supervised manifold learning. This learning strategy focuses on estimating low-dimensional representation of two samples, $\mathbf{z}_n$ and $\mathbf{z}_m$, which sufficiently reflect the neighbor relation of the same samples, $\mathbf{x}_n$ and $\mathbf{x}_m$, in original i-vector space. The corresponding class targets t_n and t_m are merged in supervised manifold learning. For PLDA speaker recognition, the i-vectors within a specific speaker are represented by the shared latent variable. Under this assumption, the joint probability of two samples from the same speaker $t_n = t_m$ is defined by $p_{nn} = 0$ and $p_{nm} = 1$ while the probability is defined by $p_{nm} = 0$ for the case of different speaker $t_n \neq t_m$. We therefore use these joint probabilities $P = \{p_{nm}\}$ to describe the target probabilities for the associated low-dimensional representations $\mathbf{z}_n$ and $\mathbf{z}_m$. Correspondingly, the Bernoulli target value $t_{nm} \triangleq p_{nm}$ is seen as an observation sample in supervised training. In addition, t distribution shown in Eq. 7.4 is here adopted to characterize the neighbor embedding distribution q_{nm} in low-dimensional subspace. Using VM-PLDA model, the variational parameters $\boldsymbol{\theta}_{\text{enc}}$ in VB-E step and the model parameters $\boldsymbol{\theta}_{\text{dec}}$ in VB-M step are estimated by maximizing the variational lower bound to fulfill the hybrid learning objective for both manifold learning and PLDA reconstruction. Gaussian distribution $\mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}(\mathbf{x}_n), \mathbf{C}(\mathbf{x}_n))$ is used as a prior for latent variable $\mathbf{z}_n$ which is incorporated to estimate PLDA parameters $\boldsymbol{\theta}_{\text{dec}} = \{\mathbf{m}, \mathbf{V}, \boldsymbol{\Sigma}\}$.

The learning objective of VM-PLDA is constructed as a joint log-likelihood of observation data consisting of i-vectors and class targets $\log p(\mathcal{X}, \mathcal{T})$ where $\mathcal{X} = \{\mathbf{x}_n\}$ and $\mathcal{T} = \{t_{nm}\}$. Supervised learning is performed. Following the fundamental in variational inference, the variational lower bound is derived on the basis of a marginal likelihood with respect to the latent variables of different data pairs $\mathcal{Z} = \{\mathbf{z}_n, \mathbf{z}_m\}$, i.e.,

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

$$\begin{aligned}\log p(\mathcal{X},\mathcal{T}) &= \log \int \prod_m \prod_n p(t_{nm}|\mathbf{z}_n,\mathbf{z}_m)p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n)d\mathcal{Z} \\ &\geq \sum_m \Bigg[\sum_n \underbrace{\mathbb{E}_{q_n}\mathbb{E}_{q_m}[\log p(t_{nm}|\mathbf{z}_n,\mathbf{z}_m)]}_{\text{manifold likelihood}} \\ &\quad + \underbrace{\mathbb{E}_{q_n}[\log p(\mathbf{x}_n|\mathbf{z}_n,\boldsymbol{\theta}_{\text{dec}})]}_{\text{PLDA likelihood}} \\ &\quad - \underbrace{\mathcal{D}_{\text{KL}}(q(\mathbf{z}_n|\mathbf{x}_n,\boldsymbol{\theta}_{\text{enc}})\|p(\mathbf{z}_n))}_{\text{Gaussian regularization } (\mathcal{L}_{\text{gau}})} \Bigg] \\ &\triangleq \mathcal{L}(\boldsymbol{\theta}_{\text{enc}},\boldsymbol{\theta}_{\text{dec}}),\end{aligned} \tag{7.7}$$

where $\mathbb{E}_{q_n}$ means taking expectation with respect to $\mathbf{z}_n|\mathbf{x}_n \sim q(\mathbf{z}_n|\mathbf{x}_n,\boldsymbol{\theta}_{\text{enc}})$. $\log p(\mathcal{X},\mathcal{T})$ can be maximized by optimizing the variational lower bound or equivalently the evidence lower bound (ELBO) $\mathcal{L}(\boldsymbol{\theta}_{\text{enc}},\boldsymbol{\theta}_{\text{dec}})$. It is interesting that VM-PLDA fulfills a multi-objective programming for subspace learning and PLDA scoring. The optimal subspace with the most likely neighbor embedding as indicated in the first term of Eq. 7.7 and the smallest PLDA reconstruction error as indicated in the second term of Eq. 7.7 is constructed. Meaningfully, the remaining term $\mathcal{L}_{\text{gau}}$ is seen as a regularization in joint training that encourages the variational distribution $q(\mathbf{z}_n|\mathbf{x}_n,\boldsymbol{\theta}_{\text{enc}})$ to be close to a prior with standard Gaussian

$$p(\mathbf{z}_n) = \mathcal{N}(\mathbf{0},\mathbf{I}). \tag{7.8}$$

This regularization is imposed to reflect the Gaussian property in PLDA. The raining procedure of VM-PLDA is completed. In general, VM-PLDA is implemented to pursue neighbor embedding and PLDA reconstruction in a stochastic latent variable model where the single-player optimization is realized to estimate the variational neural network for speaker recognition. We argue that the latent variables $\{\mathbf{z}_n,\mathbf{z}_m\}$ may *not* be so discriminative or competitive. Single-player optimization may be improved to the two-player optimization that here means the adversarial learning based on the minimax optimization. Discriminative latent variables are focused. In what follows, we address the neural adversarial learning for speaker recognition.

7.2 Adversarial Manifold PLDA

This section addresses an adversarial manifold PLDA where the adversarial manifold learning is presented for the construction of PLDA-based speaker recognition [119]. The auxiliary classifier GAN is introduced for this implementation. The background of GAN has been addressed in Section 4.6.

7.2.1 Auxiliary Classifier GAN

For practical consideration, it is possible to improve the generation of the synthesized i-vectors $\widehat{\mathbf{x}}$ in GAN by incorporating auxiliary information such as class labels or speaker

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

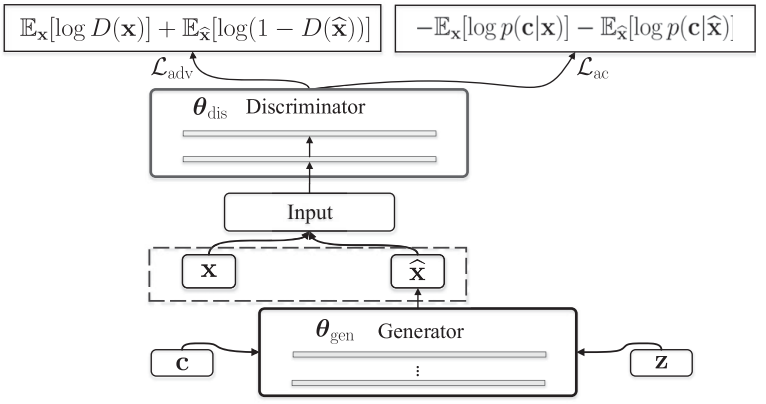

Figure 7.3 Procedure for calculation of adversarial loss $\mathcal{L}_{\text{adv}}$ and auxiliary classification loss $\mathcal{L}_{\text{ac}}$ in auxiliary classifier GAN. [Reprinted from *Adversarial Learning and Augmentation for Speaker Recognition (Figure 2), by J.T. Chien and K.T. Peng, Proceedings Odyssey 2018 The Speaker and Language Recognition Workshop, pp. 342–348, with permission of ISCA*]

identities $\mathbf{c}$ into the the generator and discriminator so that class conditional samples [251] or speaker conditional i-vectors can be produced. For example, the auxiliary classifier GAN (denoted by AC-GAN) [250] makes use of such an auxiliary information to train the discriminator and generator. Specifically, the discriminator was implemented and treated as an auxiliary neural network as decoder to output the class labels $\mathbf{c}$ given training data $\mathbf{x}$ or synthesize data $\widehat{\mathbf{x}}$. As shown in Figure 7.3, the generator receives a class label $\mathbf{c}$ and a noise sample $\mathbf{z}$ as input and synthesizes a vector $\widehat{\mathbf{x}}$ as output:

$$\widehat{\mathbf{x}} = G(\mathbf{z}, \mathbf{c}). \tag{7.9}$$

We therefore incorporate the auxiliary classification (AC) loss as an additional loss which consists of two terms. One is the class conditional distribution of class $\mathbf{c}$ measured by the real i-vector $\mathbf{x}$ while the other is the same distribution but measured by the synthesized i-vector $\widehat{\mathbf{x}}$, namely,

$$\mathcal{L}_{\text{ac}} = -\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\log p(\mathbf{c}|\mathbf{x})] - \mathbb{E}_{\widehat{\mathbf{x}}\sim p_{\text{gen}}(\mathbf{x})}[\log p(\mathbf{c}|\widehat{\mathbf{x}})]. \tag{7.10}$$

There are two posterior labels in the outputs of discriminator. One is the posterior for the label of true or fake and the other is the posterior for the label of classes. AC-GAN deals with a minimax optimization for estimation of discriminator and generator. Both discriminator and generator are estimated by minimizing the AC loss in Eq. 7.10. At the same time, the adversarial loss $\mathcal{L}_{\text{adv}}$ (as defined in Eq. 4.68) is included in minimax optimization for AC-GAN. The discriminator parameters $\boldsymbol{\theta}_{\text{dis}}$ are optimized by maximizing $\mathcal{L}_{\text{adv}}$, and the generator parameters $\boldsymbol{\theta}_{\text{gen}}$ are optimized by minimizing $\mathcal{L}_{\text{adv}}$. The objectives $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{ac}}$ are both included in multi-objective learning of discriminator $\boldsymbol{\theta}_{\text{dis}}$ as well as generator $\boldsymbol{\theta}_{\text{gen}}$. The AC loss $\mathcal{L}_{\text{ac}}$ is introduced to learn for the class information $\mathbf{c}$ while the adversarial loss $\mathcal{L}_{\text{adv}}$ is considered to implement an adversarial generative network for speaker recognition.

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

7.2.2 Adversarial Manifold Learning

As we know, speaker recognition system using a PLDA-based latent variable model is considerably affected by the learned latent space $\mathcal{Z}$. This space is constructed and controlled by the encoder and decoder with parameters $\boldsymbol{\theta}_{\text{enc}}$ and $\boldsymbol{\theta}_{\text{dec}}$, respectively. The adversarial generative network (GAN) is implemented to enhance the discrimination in latent variable space based on PLDA and develop the adversarial manifold PLDA (denoted by AM-PLDA) for speaker recognition. An additional discriminator is merged as displayed in Figure 7.4. The discriminator is characterized by a fully connected neural network given by the parameter $\boldsymbol{\theta}_{\text{dis}} = \mathbf{W}_{\text{dis}}$. The posterior output of discriminator is denoted by $D(\mathbf{z}_n, \boldsymbol{\theta}_{\text{dis}})$ with the input of the latent code $\mathbf{z}_n$.

In implementation of AM-PLDA, a multi-objective learning, driven by minimax optimization, is run to jointly estimate the discriminator parameters $\boldsymbol{\theta}_{\text{dis}} = \mathbf{W}_{\text{dis}}$ and the generator parameters $\{\boldsymbol{\theta}_{\text{enc}} = \mathbf{W}_{\text{enc}}, \boldsymbol{\theta}_{\text{dec}} = \{\mathbf{m}, \mathbf{V}, \boldsymbol{\Sigma}\}\}$ in a form of

$$\min_{\{\boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dec}}\}} \max_{\boldsymbol{\theta}_{\text{dis}}} \Bigg\{ \underbrace{\sum_n \Bigg[-\sum_m \mathbb{E}_{q_n} \mathbb{E}_{q_m} [\log p(t_{nm} | \mathbf{z}_n, \mathbf{z}_m)]}_{\mathcal{L}_{\text{m}}} \\ \underbrace{- \mathbb{E}_{q_n} [\log p(\mathbf{x}_n | \mathbf{z}_n, \boldsymbol{\theta}_{\text{dec}})]}_{\mathcal{L}_{\text{rec}}} \\ \underbrace{+ \mathbb{E}_{p_n} [\log D(\mathbf{z}_n, \boldsymbol{\theta}_{\text{dis}})] + \mathbb{E}_{q_n} [\log(1 - D(\mathbf{z}_n, \boldsymbol{\theta}_{\text{dis}}))] \Bigg]}_{\mathcal{L}_{\text{adv}}} \Bigg\}. \tag{7.11}$$

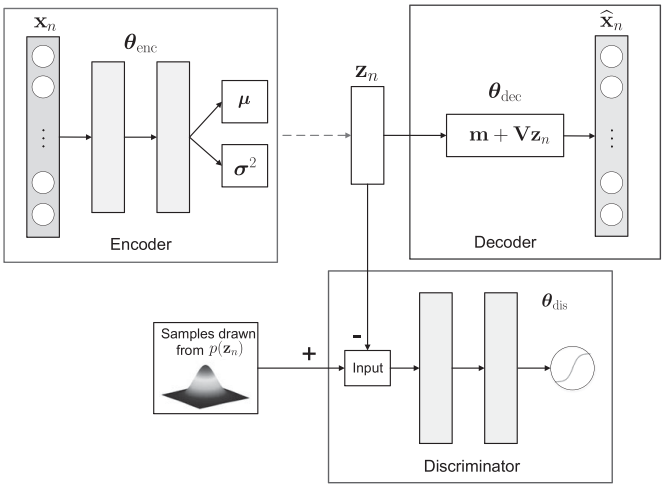

Figure 7.4 Adversarial manifold PLDA consisting of an encoder, a decoder, and a discriminator. A dashed line means sampling. Manifold loss is calculated from the neighbors of latent variables $\{\mathbf{z}_n, \mathbf{z}_m\}$. [Reprinted from *Adversarial Manifold Learning for Speaker Recognition (Figure 3), by J.T. Chien and K.T. Peng, Proceedings ASRU, 2017, with permission of IEEE*]

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

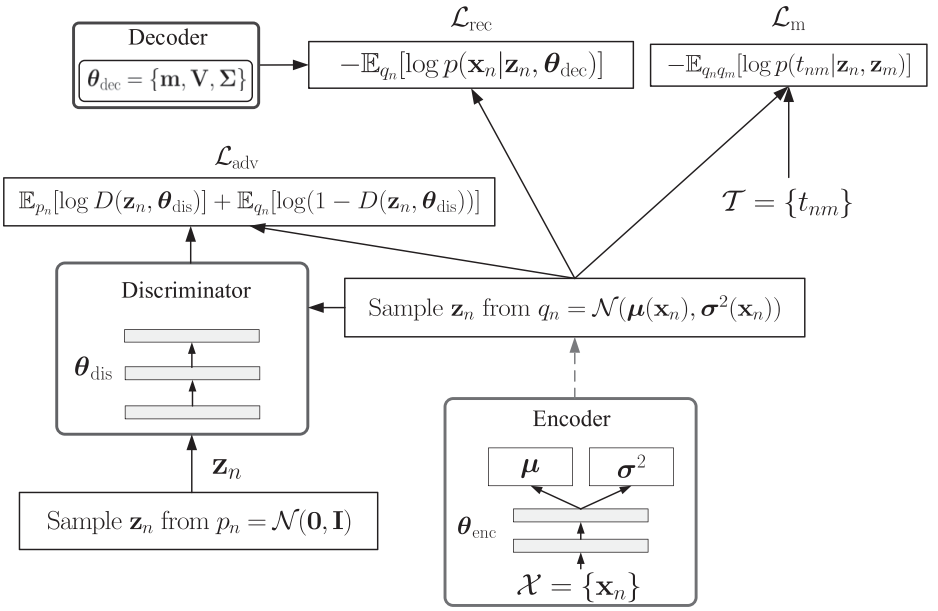

Figure 7.5 Procedure for calculation of manifold loss, PLDA reconstruction loss, and adversarial loss for adversarial manifold PLDA, which are displayed in different rectangles. [Adapted from *Adversarial Manifold Learning for Speaker Recognition (Figure 4), by J.T. Chien and K.T. Peng, Proceedings ASRU, 2017, with permission of IEEE*]

There are three loss functions in multi-objective learning that are the manifold loss $\mathcal{L}_{\text{m}}$, the PLDA reconstruction loss $\mathcal{L}_{\text{rec}}$, and the adversarial loss $\mathcal{L}_{\text{adv}}$. Adversarial loss is seen as the negative cross entropy function, which is calculated by using the class output posterior of discriminator $D(\mathbf{z}_n, \boldsymbol{\theta}_{\text{dis}})$ with the true distribution $p(\mathbf{z}_n) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and the class output posterior $1 - D(\mathbf{z}_n, \boldsymbol{\theta}_{\text{dis}})$ with the fake or variational distribution $q(\mathbf{z}_n|\mathbf{x}_n, \boldsymbol{\theta}_{\text{enc}})$. The system architecture of calculating different objectives $\{\mathcal{L}_{\text{m}}, \mathcal{L}_{\text{rec}}, \mathcal{L}_{\text{adv}}\}$ is depicted in Figure 7.5. The adversarial manifold PLDA is developed for speaker recognition. This AM-PLDA is seen as an extension of VM-PLDA by replacing the regularization term in Eq. 7.7 using the adversarial loss. The loss is calculated by a binary discriminator that judges if the latent variable $\mathbf{z}_n$ comes from a prior of standard Gaussian $p(\mathbf{z}_n) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ or from a variational posterior $q(\mathbf{z}_n|\mathbf{x}_n, \boldsymbol{\theta}_{\text{enc}})$ using an encoder. In minimax optimization using Eq. 7.11, we maximize the hybrid objective over the discriminator with parameters $\boldsymbol{\theta}_{\text{dis}}$ (included in the third and fourth terms) and minimize the hybrid objective over the encoder with parameters $\boldsymbol{\theta}_{\text{enc}}$ (included in the terms involving $\mathbb{E}_{q_n}$) and the decoder with parameters $\boldsymbol{\theta}_{\text{dec}}$ (included in the second term). Importantly, PLDA scoring is treated as a model regularization that is incorporated to mitigate the mode collapse problem in GAN model [253]. The parameters of encoder $\boldsymbol{\theta}_{\text{enc}}$, discriminator $\boldsymbol{\theta}_{\text{dis}}$, and decoder $\boldsymbol{\theta}_{\text{dec}}$ are jointly trained via a two-player game. The optimal parameters are obtained to achieve the *worst* performance in classification of latent code $\mathbf{z}_n$ either sampled from $q(\mathbf{z}_n|\mathbf{x}_n, \boldsymbol{\theta}_{\text{enc}})$ or

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

given by standard Gaussian prior $p(\mathbf{z}_n)$. The adversarial learning is performed with multi-objectives including manifold loss, cross-entropy loss, and reconstruction loss. We realize a two-player game theory for speaker recognition where the multi-objectives are maximized to estimate the discriminator and minimized to estimate the generator including an encoder and a decoder.

Therefore, the expectation of the negative log likelihood with Bernoulli distribution is calculated with respect to $\mathcal{Z}$ by using the samples $\mathbf{z}_{n,l}$ and $\mathbf{z}_{m,s}$ from a pair of neighbors, $\mathbf{z}_n$ and $\mathbf{z}_m$, which are obtained from the encoder distributions $q(\mathbf{z}_n|\mathbf{x}_n, \boldsymbol{\theta}_{\text{enc}})$ and $q(\mathbf{z}_m|\mathbf{x}_m, \boldsymbol{\theta}_{\text{enc}})$, respectively. The manifold loss $\mathcal{L}_m$ in Eq. 7.11 is implemented by

$$\begin{aligned}\mathcal{L}_m = \sum_n \sum_m \sum_l \sum_s \Bigg[& t_{nm} \frac{\nu+1}{2} \log\left(1 + \|\mathbf{z}_{n,l} - \mathbf{z}_{m,s}\|^2/\nu\right) \\ & - (1 - t_{nm}) \log\left(1 - \left(1 + \|\mathbf{z}_{n,l} - \mathbf{z}_{m,s}\|^2/\nu\right)^{-\frac{\nu+1}{2}}\right)\Bigg].\end{aligned} \tag{7.12}$$

This AM-PLDA is seen as a neural network solution to the supervised t-SNE [254] where the adversarial learning is merged. In addition, PLDA reconstruction error $\mathcal{L}_{\text{rec}}$ in the second term of Eq. 7.11 is seen as an expectation of negative log-likelihood, which is calculated by using the samples $\{\mathbf{z}_{n,l}\}_{l=1}^{L}$ of latent variable $\mathbf{z}_n$:

$$\begin{aligned}\mathcal{L}_{\text{rec}} = \frac{1}{2} \sum_{n=1}^{N} \sum_{l=1}^{L} \Big[& \log|2\pi\boldsymbol{\Sigma}| \\ & + (\mathbf{x}_n - \mathbf{m} - \mathbf{V}\mathbf{z}_{n,l})^{\top} \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \mathbf{m} - \mathbf{V}\mathbf{z}_{n,l})\Big].\end{aligned} \tag{7.13}$$

where the decoder parameters $\boldsymbol{\theta}_{\text{dec}} = \{\mathbf{m}, \mathbf{V}, \boldsymbol{\Sigma}\}$ are applied.

There are twofold differences in comparison between VM-PLDA and AM-PLDA. First, VM-PLDA carries out a single-player optimization with a regularization term based on the KL divergence that is minimized to regularize the variational posterior $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}_{\text{enc}})$ to be close to a standard Gaussian prior $p(\mathbf{z})$, as seen in Eq. 7.7. Differently, as explained in Eqs. 4.67 and 4.74, AM-PLDA implements a two-player optimization with a maximum adversarial loss that is achieved to find an optimized discriminator where the Jensen–Shannon (JS) divergence $p_{\text{gen}}(\mathbf{x}) \rightarrow p(\mathbf{x})$ is minimum. The aspect of discriminator was ignored in latent variable representation in the implementation of VM-PLDA. In addition, different from VM-PLDA, AM-PLDA is viewed as an extension of adversarial autoencoder [122] for PLDA speaker recognition where the latent variable of i-vector is encoded and then used to decode for PLDA scoring. Learning representation is boosted by an adversarial training procedure.

This chapter not only addresses the adversarial learning for PLDA subspace modeling based on the adversarial manifold learning, but also presents the adversarial learning for data augmentation. The adversarial augmentation learning is developed for PLDA

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

speaker verification in the condition that the collected i-vectors are imbalanced among various speakers.

7.3 Adversarial Augmentation PLDA

This section describes a specialized generative adversarial network (GAN) for data augmentation [120]. The i-vectors generated by the network can be augmented to the original training set to compensate the issues of imbalanced and insufficient data in PLDA-based speaker verification. The neural network is trained to ensure that the generated i-vectors reflect speaker behaviors or properties that are matching with those of the real i-vectors. We will address how practical constraints are imposed to perform the constrained optimization, so as to improve model capability for PLDA speaker recognition. An adversarial augmentation PLDA (denoted by AA-PLDA) is developed and implemented to generate new i-vectors for data augmentation. Similar to AM-PLDA, this AA-PLDA also considers speaker identity to build a kind of *speaker-dependent* i-vector generation model where the auxiliary classifier GAN (AC-GAN), as mentioned in Section 7.2.1, is incorporated in i-vector augmentation. Namely, the noise sample $\mathbf{z}$ and speaker label $\mathbf{c}$ are both merged in generator to find synthesized i-vector, i.e.,

$$\widehat{\mathbf{x}} = G(\mathbf{z}, \mathbf{c}). \tag{7.14}$$

Hereafter, the sample index n or m is neglected for notation simplicity.

In implementation of AA-PLDA for speaker recognition, the adversarial augmentation learning is performed by jointly optimizing an adversarial loss $\mathcal{L}_{\text{adv}}$ as given in Eq. 4.68 as well as an auxiliary classification (AC) loss $\mathcal{L}_{\text{ac}}$ as given in Eq. 7.10. AC loss generally expresses the expectation of the negative logarithm of class conditional likelihoods $-\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\log p(\mathbf{c}|\mathbf{x})]$ and $-\mathbb{E}_{\widehat{\mathbf{x}}\sim p_{\text{gen}}(\mathbf{x})}[\log p(\mathbf{c}|\widehat{\mathbf{x}})]$ measured by the original i-vector and the synthesized i-vector, respectively. Nevertheless, it is insufficient to directly realize AC-GAN for data augmentation in PLDA speaker recognition. Some physical properties can be considered as constraints to compensate such an insufficiency. We therefore incorporate a couple of constraints to characterize practical meanings to generate new and informative i-vectors for PLDA speaker recognition. The cosine generative adversarial network and the PLDA generative adversarial network are accordingly developed as the specialized generative models for synthesis of i-vectors in PLDA speaker recognition.

7.3.1 Cosine Generative Adversarial Network

As we know, cosine distance is usually used as the scoring measure or similarity measure for i-vectors between target speaker and test speaker. This distance measure is an alternative to PLDA scoring based on the likelihood ratio. A Cosine measure generally works well as the scoring measure. A meaningful idea for implementing GAN for

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

speaker recognition is to shape up the synthesized i-vector $\widehat{\mathbf{x}}$ based on the cosine distance between the generated i-vector and real i-vector $\mathbf{x}$. We introduce a regularization loss that is formed by the negative cosine distance between synthesized i-vector $\widehat{\mathbf{x}}$ and original i-vector $\mathbf{x}$

$$\mathcal{L}_{\text{cosx}} = -\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),\widehat{\mathbf{x}}\sim p_{\text{gen}}(\mathbf{x})}\left[D_{\cos}(\mathbf{x},\widehat{\mathbf{x}})|G\right]. \tag{7.15}$$

This measure is affected by the generator with function $G(\mathbf{z},\mathbf{c})$, which is characterized by parameter $\boldsymbol{\theta}_{\text{gen}}$. We treat this measure as the regularization term and minimize it to pursue *distribution matching*.

In real implementation, the evaluation of similarity between original i-vector and synthesized i-vector may be degraded since i-vectors are sometimes deteriorated by channel factors and other variabilities. The representation capability for speaker identities may be insufficient. One reasonable idea is to extend the evaluation of cosine similarity between original i-vector and synthesized i-vector by measuring in i-vector raw space, which is denoted by $\mathcal{X}$ rather than in i-vector feature space, which is denoted by $\mathcal{Y}$. As a result, the cosine loss as regularization loss is calculated by using the hidden feature representations $\{\mathbf{y},\widehat{\mathbf{y}}\}$ to replace the i-vector representations $\{\mathbf{x},\widehat{\mathbf{x}}\}$:

$$\mathcal{L}_{\text{cosy}} = -\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),\widehat{\mathbf{x}}\sim p_{\text{gen}}(\mathbf{x})}\left[D_{\cos}(\mathbf{y},\widehat{\mathbf{y}})|G,D\right]. \tag{7.16}$$

Notably, this loss function is related to both generator G and discriminator D given by the parameters $\boldsymbol{\theta}_{\text{gen}}$ and $\boldsymbol{\theta}_{\text{dis}}$, respectively. In the implementation, the discriminator D is modeled by a fully connected neural network and used as a binary classification for input vector that is either original i-vector $\mathbf{x}$ or synthesized i-vector $\widehat{\mathbf{x}}$. It is meaningful to extract the hidden units in the last layer of discriminator and use them as the training samples $\{\mathbf{y},\widehat{\mathbf{y}}\}$ to calculate the cosine loss $\mathcal{L}_{\text{cosy}}$ in feature space. We therefore investigate the cosine distance in feature domain $\mathcal{Y}$. Correspondingly, we implement the adversarial augmentation PLDA (AA-PLDA) by using the so-called cosine GANs, Cosx-GAN, and Cosy-GAN based on the cosine losses Cosx and Cosy, which are measured in the spaces $\mathcal{X}$ and $\mathcal{Y}$, respectively. As a result, there are two generative models that are estimated according to individual multi-objectives for solving minimax optimization problems as shown below

$$\text{Cosx-GAN: } \mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{ac}} + \mathcal{L}_{\text{cosx}} \tag{7.17}$$

$$\text{Cosy-GAN: } \mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{ac}} + \mathcal{L}_{\text{cosy}}. \tag{7.18}$$

Using AA-PLDA, we can carry out two variants of GAN, Cosx-GAN, and Cosy-GAN, where the generator and discriminator are trained in accordance with four loss functions including $\mathcal{L}_{\text{adv}}$, $\mathcal{L}_{\text{ac}}$, $\mathcal{L}_{\text{cosx}}$, and $\mathcal{L}_{\text{cosy}}$. The calculation of these loss functions is shown in Figure 7.6. Notably, the calculation of gradient of cosine distance with respect to the synthesized i-vector $\widehat{\mathbf{x}}$ is required for SGD parameter updating in Cosx-GAN and Cosy-GAN. The approximation to this gradient can be derived by

$$\frac{\partial}{\partial\widehat{\mathbf{x}}}D_{\cos}(\mathbf{x},\widehat{\mathbf{x}}) \approx \frac{\mathbf{x}}{|\mathbf{x}|\,|\widehat{\mathbf{x}}|} - D_{\cos}(\mathbf{x},\widehat{\mathbf{x}})\frac{\widehat{\mathbf{x}}}{|\widehat{\mathbf{x}}|^2} \tag{7.19}$$

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

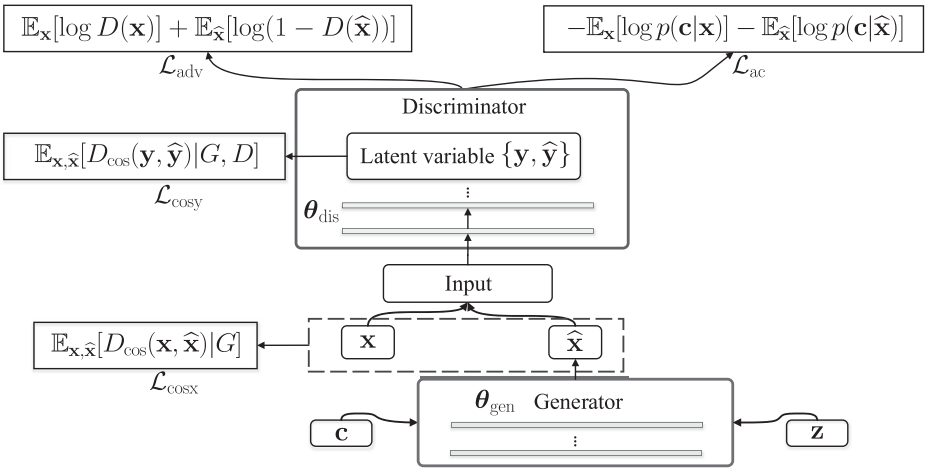

Figure 7.6 The loss functions in the minimax optimization procedure in the cosine GAN, where $\mathcal{L}_{\text{cosx}}$ and $\mathcal{L}_{\text{cosy}}$ denote the cosine loss functions in i-vector space $\mathcal{X}$ and feature space $\mathcal{Y}$, respectively. [Reprinted from *Adversarial Learning and Augmentation for Speaker Recognition (Figure 3), by J.T. Chien and K.T. Peng, Proceedings Odyssey 2018 The Speaker and Language Recognition Workshop, pp. 342–348, with permission of ISCA*]

because

$$
\begin{aligned}
D_{\cos}(\mathbf{x},\widehat{\mathbf{x}}+d\widehat{\mathbf{x}}) &= \frac{\mathbf{x}\cdot\widehat{\mathbf{x}}+\mathbf{x}\cdot d\widehat{\mathbf{x}}}{|\mathbf{x}|\,|\widehat{\mathbf{x}}+d\widehat{\mathbf{x}}|} \\
&\approx \frac{\mathbf{x}\cdot\widehat{\mathbf{x}}+\mathbf{x}\cdot d\widehat{\mathbf{x}}}{|\mathbf{x}|\left(1+\frac{\widehat{\mathbf{x}}}{|\widehat{\mathbf{x}}|^2}\cdot d\widehat{\mathbf{x}}\right)|\widehat{\mathbf{x}}|} \\
&\approx \frac{\mathbf{x}\cdot\widehat{\mathbf{x}}+\mathbf{x}\cdot d\widehat{\mathbf{x}}}{|\mathbf{x}|\,|\widehat{\mathbf{x}}|}\left(1-\frac{\widehat{\mathbf{x}}}{|\widehat{\mathbf{x}}|^2}\cdot d\widehat{\mathbf{x}}\right) \\
&\approx \frac{\mathbf{x}\cdot\widehat{\mathbf{x}}}{|\mathbf{x}|\,|\widehat{\mathbf{x}}|}+\left(\frac{\mathbf{x}}{|\mathbf{x}|\,|\widehat{\mathbf{x}}|}-\frac{\mathbf{x}\cdot\widehat{\mathbf{x}}}{|\mathbf{x}|\,|\widehat{\mathbf{x}}|}\frac{\widehat{\mathbf{x}}}{|\widehat{\mathbf{x}}|^2}\right)\cdot d\widehat{\mathbf{x}} \\
&= D_{\cos}(\mathbf{x},\widehat{\mathbf{x}})+\left(\frac{\mathbf{x}}{|\mathbf{x}|\,|\widehat{\mathbf{x}}|}-D_{\cos}(\mathbf{x},\widehat{\mathbf{x}})\frac{\widehat{\mathbf{x}}}{|\widehat{\mathbf{x}}|^2}\right)\cdot d\widehat{\mathbf{x}}.
\end{aligned}
\tag{7.20}
$$

As illustrated in Algorithms 7 and 8, the learning procedures based on Cosx-GAN and Cosy-GAN are implemented as two realizations of AC-GAN where the model regularization based on the cosine similarity using original i-vector representation $\mathcal{X}$ and feature space $\mathcal{Y}$ is considered, respectively. The feedforward neural network parameters in generator $\boldsymbol{\theta}_{\text{gen}}$ and discriminator $\boldsymbol{\theta}_{\text{dis}}$ are accordingly estimated. The parameter updating is performed by using SGD algorithm with momentum where minibatches of i-vectors and noise samples are used. The empirical setting is to run three updating steps (i.e., $k = 3$) for discriminator, and then run one updating step for generator at each training iteration. In implementation of Cosx-GAN, the cosine loss in i-vector space $\mathcal{L}_{\text{cosx}}$ is considered in updating the generator without updating the discriminator. We do perform the minimax optimization with respect to discriminator and generator. It is

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

Algorithm 7 Adversarial augmentation learning where Cosx-GAN is implemented

Initialize the parameters $\{\boldsymbol{\theta}_{\text{gen}}, \boldsymbol{\theta}_{\text{dis}}\}$
For number of training iterations
  For k steps
    sample a minibatch of i-vectors $\{\mathbf{x}_n\}_{n=1}^N$ from $p(\mathbf{x})$
    sample a minibatch of noise examples $\{\mathbf{z}_n\}_{i=n}^N$ from $p(\mathbf{z})$
    update the discriminator $\boldsymbol{\theta}_{\text{dis}}$ by descending the stochastic gradient
      $\frac{1}{N}\sum_{n=1}^N \nabla_{\boldsymbol{\theta}_{\text{dis}}}(-\mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{ac}})$
  End For
  sample a minibatch of i-vectors $\{\mathbf{x}_n\}_{n=1}^N$ from $p(\mathbf{x})$
  sample a minibatch of noise examples $\{\mathbf{z}_n\}_{n=1}^N$ from $p(\mathbf{z})$
  update the generator $\boldsymbol{\theta}_{\text{gen}}$ by descending the stochastic gradient
    $\frac{1}{N}\sum_{n=1}^N \nabla_{\boldsymbol{\theta}_{\text{gen}}}(\mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{ac}} + \mathcal{L}_{\text{cosx}})$
End For

Algorithm 8 Adversarial augmentation learning where Cosy-GAN is implemented

Initialize the parameters $\{\boldsymbol{\theta}_{\text{gen}}, \boldsymbol{\theta}_{\text{dis}}\}$
For number of training iterations
  For k steps
    sample a minibatch of i-vectors $\{\mathbf{x}_n\}_{n=1}^N$ from $p(\mathbf{x})$
    sample a minibatch of noise examples $\{\mathbf{z}_n\}_{i=n}^N$ from $p(\mathbf{z})$
    update the discriminator $\boldsymbol{\theta}_{\text{dis}}$ by descending the stochastic gradient
      $\frac{1}{N}\sum_{i=1}^N \nabla_{\boldsymbol{\theta}_{\text{dis}}}(-\mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{ac}} + \mathcal{L}_{\text{cosy}})$
  End For
  sample a minibatch of i-vectors $\{\mathbf{x}_n\}_{n=1}^N$ from $p(\mathbf{x})$
  sample a minibatch of noise examples $\{\mathbf{z}_n\}_{n=1}^N$ from $p(\mathbf{z})$
  update the generator $\boldsymbol{\theta}_{\text{gen}}$ by descending the stochastic gradient
    $\frac{1}{N}\sum_{i=1}^N \nabla_{\boldsymbol{\theta}_{\text{gen}}}(\mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{ac}} + \mathcal{L}_{\text{cosy}})$
End For

because the gradients of adversarial loss function $\mathcal{L}_{\text{adv}}$ in updating discriminator and generator have different signs. On the other hand, Cosy-GAN is implemented by using the cosine loss function $\mathcal{L}_{\text{cosy}}$, which depends both on the parameters of discriminator and generator.

7.3.2 PLDA Generative Adversarial Network

In this subsection, we further explore how to incorporate PLDA constraint in adversarial augmentation learning for speaker recognition. Here, the property of PLDA is sufficiently reflected or represented as the constraint or regularization in GAN training. The variational autoencoder (VAE) [237, 249] is considered to move forward an advanced solution where different regularization terms are derived and forced to artificially synthesize i-vectors with good-quality PLDA speaker recognition is performed by using

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

real i-vectors as well as synthesized i-vectors. The resulting PLDA GAN is different from the cosine GAN mentioned in Section 7.3.1. Cosine GAN uses a fully connected neural network to extract *deterministic* latent features $\mathbf{y}$ to calculate the cosine loss function $\mathcal{L}_{\text{cosy}}$. An extension of combining cosine GAN and PLDA scoring is to carry out the so-called PLDA-Cos-GAN where the stochastic or variational neural network is introduced with the regularization terms from cosine similarity and some others. The variational distribution is used as the encoder to find the stochastic latent variable of $\mathbf{x}$ or $\widehat{\mathbf{x}}$ based on a Gaussian distribution driven by mean vector $\boldsymbol{\mu}(\mathbf{x}) = \{\mu_d(\mathbf{x})\}$ and diagonal covariance matrix $\sigma^2(\mathbf{x})\mathbf{I} = \text{diag}\{\sigma_d^2(\mathbf{x})\}$ using feedforward neural network parameters $\boldsymbol{\theta}_{\text{enc}}$, i.e.,

$$\mathbf{y} \sim q(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \sigma^2(\mathbf{x})\mathbf{I}). \tag{7.21}$$

The adversarial augmentation learning is developed through variational learning where a decoder with PLDA parameters $\boldsymbol{\theta}_{\text{dec}} = \{\mathbf{m}, \mathbf{V}, \boldsymbol{\Sigma}\}$. Such a decoder was similarly adopted in the variational manifold learning as mentioned in Section 7.1.2 and the adversarial manifold learning as addressed in Section 7.2.2. According to the variational inference, we minimize the negative evidence lower bound (ELBO) based on negative log-likelihood using a collection of i-vectors $\mathbf{x} = \{\mathbf{x}_n\}$ [30, 249]:

$$\begin{aligned}
-\log p(\mathbf{x}) &= -\int q(\mathbf{y}|\mathbf{x}) \log(p(\mathbf{x}))d\mathbf{y} \\
&= -\int q(\mathbf{y}|\mathbf{x}) \log\left(\frac{p(\mathbf{y},\mathbf{x})}{q(\mathbf{y}|\mathbf{x})}\right) d\mathbf{y} - \int q(\mathbf{y}|\mathbf{x}) \log\left(\frac{q(\mathbf{y}|\mathbf{x})}{p(\mathbf{y}|\mathbf{x})}\right) d\mathbf{y} \\
&\leq -\int q(\mathbf{y}|\mathbf{x}) \log(p(\mathbf{x}|\mathbf{y}))d\mathbf{y} - \int q(\mathbf{y}|\mathbf{x}) \log\left(\frac{p(\mathbf{y})}{q(\mathbf{y}|\mathbf{x})}\right) d\mathbf{y} \\
&= \underbrace{-\mathbb{E}_{q(\mathbf{y}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{y})]}_{\mathcal{L}_{\text{rec}}} + \underbrace{\mathcal{D}_{\text{KL}}(q(\mathbf{y}|\mathbf{x})\|p(\mathbf{y}))}_{\mathcal{L}_{\text{gau}}}.
\end{aligned} \tag{7.22}$$

We can see that the PLDA reconstruction loss $\mathcal{L}_{\text{rec}}$ and the Gaussian regularization $\mathcal{L}_{\text{gau}}$ are simultaneously minimized. In this learning objective, the conditional likelihood function

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{m} + \mathbf{V}\mathbf{y}, \boldsymbol{\Sigma}) \tag{7.23}$$

and the standard Gaussian prior $p(\mathbf{y}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ are used. To implement Eq. (7.22), we calculate the PLDA scoring or PLDA loss $\mathcal{L}_{\text{rec}}$ by using Eq. (7.13) and also calculate the Gaussian regularization term $\mathcal{L}_{\text{gau}}$, which is derived as

$$\mathcal{D}_{\text{KL}}(\mathcal{N}(\boldsymbol{\mu}, \sigma^2)\|\mathcal{N}(\mathbf{0}, \mathbf{I})) = \frac{1}{2}\sum_d \left[\mu_d^2 + \sigma_d^2 + \log(\sigma_d^2) - 1\right]. \tag{7.24}$$

We therefore develop a so-called PLDA-GAN, which is a special realization of adversarial autoencoder [122] for speaker recognition based on the PLDA scoring.

Using PLDA-Cos-GAN, there are five loss terms derived in adversarial augmentation learning for speaker recognition. The original i-vector $\mathbf{x}$ and the synthesized i-vector $\widehat{\mathbf{x}}$ from the *generator* are merged in calculation of these five loss functions. The

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

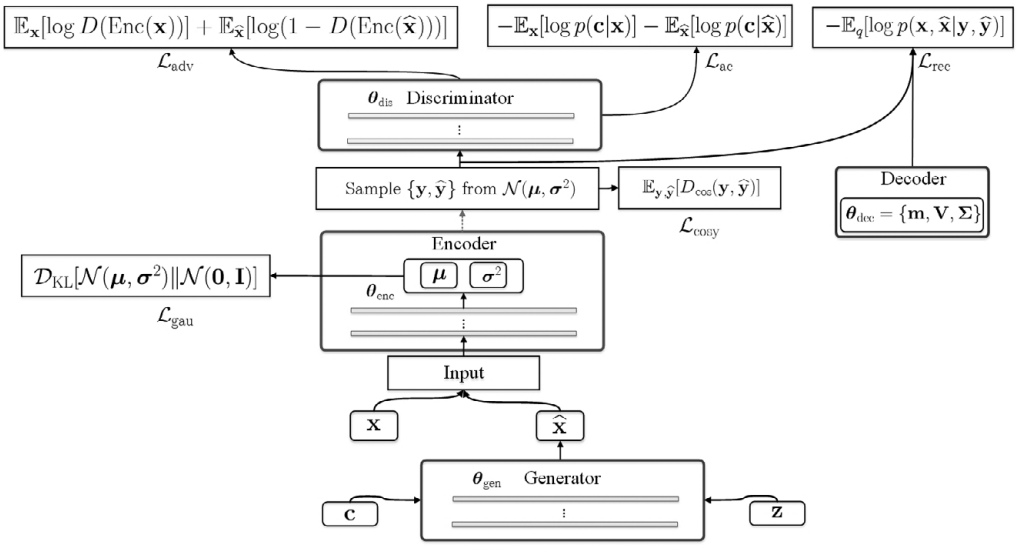

Figure 7.7 The loss functions in the minimax optimization procedure in the PLDA-Cos-GAN, where $\mathcal{L}_{\mathrm{gau}}$ and $\mathcal{L}_{\mathrm{rec}}$ denote the loss functions for Gaussian regularization and PLDA reconstruction, respectively. [Reprinted from *Adversarial Learning and Augmentation for Speaker Recognition (Figure 4), by J.T. Chien and K.T. Peng, Proceedings Odyssey 2018 The Speaker and Language Recognition Workshop, pp. 342–348, with permission of ISCA*]

training of *encoder*, *discriminator*, and *decoder* in PLDA-Cos-GAN is based on the loss functions $\{\mathcal{L}_{\mathrm{gau}}, \mathcal{L}_{\mathrm{cosy}}\}$, $\{\mathcal{L}_{\mathrm{adv}}, \mathcal{L}_{\mathrm{ac}}\}$, and $\mathcal{L}_{\mathrm{rec}}$, respectively, as illustrated in Figure 7.7. When calculating the objectives $\mathcal{L}_{\mathrm{adv}}$, $\mathcal{L}_{\mathrm{ac}}$, $\mathcal{L}_{\mathrm{cosy}}$, and $\mathcal{L}_{\mathrm{rec}}$, the samples of variational Gaussian distribution $q(\mathbf{y}|\mathbf{x})$ are used. The key difference between Cosy-GAN and PLDA-Cos-GAN is the calculation of cosine loss $\mathcal{L}_{\mathrm{cosy}}$. Cosy-GAN calculates this term from last layer of discriminator while PLDA-Cos-GAN calculates this loss by using encoder. In addition, the hidden variable $\mathbf{y}$ is random in PLDA-Cos-GAN while the latent code $\mathbf{y}$ is deterministic in Cosy-GAN. Using PLDA-Cos-GAN, the parameters of decoder $\boldsymbol{\theta}_{\mathrm{dec}}$ based on PLDA model and the parameters of generator, encoder, and discriminator using the fully connected neural network $\{\boldsymbol{\theta}_{\mathrm{gen}}, \boldsymbol{\theta}_{\mathrm{enc}}, \boldsymbol{\theta}_{\mathrm{dis}}\}$ are jointly estimated according to minimax optimization. The training procedure for different PLDA-Cos-GAN parameters is formulated and implemented in Algorithm 9. The updating of individual parameters is performed by taking derivatives for those related loss terms. $\mathbf{z}$ and $\mathbf{y}$ are stochastic and used as the latent codes for generator and decoder, respectively. In Algorithm 9, we introduce regularization parameter λ to adjust the tradeoff between different loss functions. This hyperparameter can be chosen by using a validation set in accordance with some selection criterion.

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press

Algorithm 9 Adversarial augmentation learning for PLDA-Cos-GAN

Initialize the parameters $\{\boldsymbol{\theta}_{\text{gen}}, \boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dec}}, \boldsymbol{\theta}_{\text{dis}}\}$
For number of training iterations
  For k steps
    sample a minibatch of i-vectors $\{\mathbf{x}_n\}_{n=1}^N$ from $p(\mathbf{x})$
    sample a minibatch of noise examples $\{\mathbf{z}_n, \mathbf{y}_n\}_{n=1}^N$ from $\{p(\mathbf{z}), p(\mathbf{y})\}$
    update the discriminator $\boldsymbol{\theta}_{\text{dis}}$ by descending the stochastic gradient
      $\frac{1}{N}\sum_{n=1}^N \nabla_{\boldsymbol{\theta}_{\text{dis}}} \lambda(-\mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{ac}})$
    update the encoder $\boldsymbol{\theta}_{\text{enc}}$ by descending the stochastic gradient
      $\frac{1}{N}\sum_{n=1}^N \nabla_{\boldsymbol{\theta}_{\text{enc}}} (\lambda(\mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{ac}} - \mathcal{L}_{\text{cosy}}) + \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{gau}})$
    update the decoder $\boldsymbol{\theta}_{\text{dec}}$ by descending the stochastic gradient
      $\frac{1}{N}\sum_{n=1}^N \nabla_{\boldsymbol{\theta}_{\text{dec}}} (\mathcal{L}_{\text{rec}})$
  End For
  sample a minibatch of i-vectors $\{\mathbf{x}_n\}_{n=1}^N$ from $p(\mathbf{x})$
  sample a minibatch of noise examples $\{\mathbf{z}_n, \mathbf{y}_n\}_{n=1}^N$ from $\{p(\mathbf{z}), p(\mathbf{y})\}$
  update the generator $\boldsymbol{\theta}_{\text{gen}}$ by descending the stochastic gradient
    $\frac{1}{N}\sum_{n=1}^N \nabla_{\boldsymbol{\theta}_{\text{gen}}} (\lambda(\mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{ac}} - \mathcal{L}_{\text{cosy}}) + \mathcal{L}_{\text{rec}})$
End For

7.4 Concluding Remarks

We have presented two deep adversarial learning approaches to PLDA speaker recognition. The generative adversarial networks were implemented in different model architectures to cope with various problems for speaker verification. A two-player game was formulated and handled by a learning procedure where the PLDA parameters and fully connected neural network parameters were optimized to build the decoder, generator, encoder, and discriminator. An adversarially learned latent representation for PLDA reconstruction or scoring was obtained by solving a minimax optimization with multiple objectives. For an integrated work, we might carry out a hybrid adversarial manifold learning and adversarial augmentation learning that jointly optimized multi-objectives to meet neighbor embedding, classification accuracy, PLDA reconstruction, and adversarial training for speaker recognition. We considered various regularization terms including cosine distance, Gaussianity, and PLDA scoring to benefit the training procedure. A theoretical solution to deep machine learning was presented to carry out state-of-the-art speaker recognition system. We integrated different learning models in this chapter including stochastic neighbor embedding, variational autoencoder, generative adversarial network, and conditional adversarial neural network.

https://doi.org/10.1017/9781108552332.008 Published online by Cambridge University Press