SEMI-SUPERVISED AND ADVERSARIAL DOMAIN ADAPTATION FOR SPEAKER RECOGNITION

LI LONGXIN

MPhil

The Hong Kong Polytechnic University

2020

Department of Electronic and Information Engineering

The Hong Kong Polytechnic University

Semi-supervised and Adversarial Domain Adaptation for Speaker Recognition

Li Longxin

A thesis submitted in partial fulfilment of the requirements for the degree of

Master of Philosophy

July 2019

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

Li Longxin
_____(Name of student)

ABSTRACT

The rapid development of technology has driven the society into a new era of AI in which speaker recognition is one of the essential techniques. Due to the unique characteristics of voiceprints, speaker recognition has been used for enhancing the security level of banking and personal security systems. Despite the great convenience provided by speaker recognition technology, some fundamental problems are remaining unsolved, which include (1) insufficient labeled samples from new acoustic environments for training supervised machine learning models and (2) domain mismatch among different acoustic environments. These fundamental problems may result in severe performance degradation in speaker recognition systems.

We proposed two methods to address the above problems. First, to reduce domain mismatch in speaker verification systems, we propose an unsupervised domain adaptation method. Second, to enhance speaker identification performance, we introduce a contrastive adversarial domain adaptation network to create a domain-invariant feature space. The first method addresses the data sparsity issue by applying spectral clustering on in-domain unlabeled data to obtain hypothesized speaker labels for adapting an out-of-domain PLDA mixture model to the target domain. To further refine the target PLDA mixture model, spectral clustering is iteratively applied to the new PLDA score matrix to produce a new set of hypothesized speaker labels. A gender-aware deep neural network (DNN) is trained to produce gender posteriors given an i-vector. The gender posteriors then replace the posterior probabilities of the indicator variables in the PLDA mixture model. A gender-dependent inter dataset variability compensation (GD-IDVC) is implemented to reduce the mismatch

between the i-vectors obtained from the in-domain and out-of-domain datasets. Evaluations based on NIST 2016 SRE show that at the end of the iterative re-training, the PLDA mixture model becomes fully adapted to the new domain. Results also show that the PLDA scores can be readily incorporated into spectral clustering, resulting in high-quality speaker clusters that could not be possibly achieved by agglomerative hierarchical clustering.

The second method aims to reduce the mismatch between male and female speakers through adversarial domain adaptation. The method mitigates an intrinsic drawback of the domain adversarial network by splitting the feature extractor into two contrastive branches, with one branch delegating for the class-dependence in the latent space and another branch focusing on domain-invariance. The feature extractor achieves these contrastive goals by sharing the first and the last hidden layers but having the decoupled branches in the middle hidden layers. We adversarially trained the label predictor to produce equal posterior probabilities across all of its outputs instead of producing one-hot outputs to ensure that the feature extractor can produce class-discriminative embedded features. We refer to the resulting domain adaptation network as a contrastive adversarial domain adaptation network (CADAN). We evaluated the domain-invariance of the embedded features via a series of speaker identification experiments under both clean and noisy conditions. Results demonstrate that the embedded features produced by CADAN lead to 8.9% and 77.6% improvement in speaker identification accuracy when compared with the conventional DAN under clean and noisy conditions, respectively.

ACKNOWLEDGMENTS

By this precious opportunity, I would like to express my very great appreciation to Dr. Man-Wai Mak for his patient and enthusiastic guidance of not only supervising my research in multifarious regions including speaker verification framework, machine learning algorithm, programming and many another fields, but also backing me up during my most depressive period by continuous encouragement. I really desire to show my sincere respect and gratitude to Dr.Mak for the efforts he has paid to review my every paperwork attentively and rectify it with specific details. During the whole researching period, the matter of prime importance he taught me is not programming or other technical skills but the continuous pursuit of self-realization in the region that you love and want to devote yourself to and this spirit definitely will motivate me no matter what I do in the future.

Besides, I also want to thanks Dr. Na Li for her great supports in this work. More importantly, without the steadfast trusts and supports from my parents, I certainly cannot reach this far so I really want to appreciate and thank my parents for whatever they did for me.

Author's Publications

1. L. X. Li and M. W. Mak, "Unsupervised domain adaptation for gender-aware PLDA

mixture models," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 2018, pp. 5269–5273.

2. L. X. Li, M. W. Mak and J. T. Chien, "Contrastive Adversarial Domain Adaptation Networks for Speaker Recognition," IEEE Transactions on Neural Networks and Learning Systems. (*submitted*)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Speaker Recognition

The recent advances in machine learning have improved our quality of life. For instance, the spoken dialog system Siri, installed in every iPhone, is a typical machine learning application. By analyzing the speech signals remotely, Siri can detect the language spoken by a user and reply the user in the recognized language.

Besides speech recognition, speaker recognition has also found applications in various domains as it is one of the most user-friendly authentication methods. Speaker recognition can be divided into two types: speaker verification and speaker identification. Speaker verification is a binary classification problem, in which the voice of a target (client) speaker is compared against the voice of a claimant. On the other hand, speaker identification is a multi-class classification problem in which the identity of a given voice is retrieved from a database. This dissertation will focus on speaker verification.

Since 2010, i-vectors have been regarded as the best feature representation for speaker verification.[1] In the i-vector approach [5], an utterance with arbitrary length is represented as a low-dimensional vector called the i-vector, which is the posterior mean of the latent

[1] After 2018, a number of studies [2, 3] found that x-vectors [4] performs better on recent NIST SRE.

factor in a factor analysis model. To compute the posterior mean (i-vector), it is necessary to align the acoustic vectors of an utterance with a Gaussian mixture model (GMM) to obtain the zero-th and first-order sufficient statistics. However, recent studies [6, 7] have found that that the zero-th order statistics can also be obtained from a deep neural network. Specifically, instead of computing the posterior probabilities of Gaussian mixtures for each frame, the authors in [6, 7] computed the posterior probabilities of senones. The i-vectors based on these DNN-derived senones posteriors are referred to as DNN i-vectors or senone i-vectors. Likewise, the i-vectors based on the GMM-derived posteriors probabilities are called GMM i-vectors.

While i-vectors are elegant representations of utterances, they contain not only speaker information but also other unwanted information such as channels, genders and languages. As a result, a robust back-end that can minimize the effect of this unwanted information is essential. So far, probabilistic linear discriminant analysis (PLDA) [8] is still the best back-end for this purpose. Given the i-vectors of a target speaker and a claimant, the likelihood ratio between the same-speaker hypothesis and different-speaker hypothesis is computed from a PLDA model. During the computation of the marginal likelihood of these two hypotheses, the unwanted variabilities in the i-vectors are marginalized out.

Despite its remarkable performance, PLDA models require a large amount of speech data with speaker labels for training. In particular, to model the speaker subspace reliably, each speaker in the training set should have multiple sessions, preferably collected by different microphones. Most of the current speech corpora (e.g., Switchboard, Fisher, and Mixer) focus on English telephone speech. Therefore, training a reliable PLDA model for English telephone speech is not an issue. However, other languages or acoustic environments may not have such rich resources. Even if we have the speech data of other

languages, we may not have the speaker labels. The NIST 2016 SRE has exactly such situation. In this evaluation, participants were given *unlabelled* speech data for training whatever models for suppressing the channel, language and gender variabilities. Also, the acoustic environments from which the speech data were collected are also very different from those in Switchboard, Fisher and Mixer. Therefore, directly applying a PLDA model trained from these telephone speech corpora to NIST 2016 SRE will lead to poor performance. This calls for domain adaptation that adapts the PLDA model trained from the out-of-domain (but resourceful) data to suit the in-domain data. This is the focus of this dissertation.

As training of PLDA models requires speaker labels, one sensible approach is to apply unsupervised clustering on the i-vectors derived from the in-domain data to produce some hypothesized speaker labels. Agglomerative hierarchical clustering [9], using Bayesian information criterion, can be used for unsupervised clustering. Alternatively, spectral clustering [10–12] which are based on the similarity between pairs of data points, and affinity propagation [13], can be used. The similarity matrix can be derived from the pairwise PLDA scores of training i-vectors. Practically, spectral clustering is able to produce hypothesized speaker labels.

1.2 Robustness Issues in Speaker Recognition

Robustness is always a serious issue in speaker verification. There are plenty of nuisances in speech signals due to disparate recording environments. Different recording environments may lead to distinguished channel noise or background noise which are detrimental to verification systems. Therefore, it is crucial to suppress these noise variabilities. A recent study [14] found that the i-vectors of utterances with similar signal-to-noise ratio

(SNR) tend to form a cluster in the i-vector space and that different regions of the i-vector space correspond to different SNR of the utterances. Based on these observations, an SNR-dependent mixture of PLDA [15] was proposed for robust speaker verification. Instead of computing the posteriors of the latent indicator variables from i-vectors based on the Bayes theorem, the posteriors are estimated from a one-dimensional SNR-GMM using SNR as input. The idea is further extended to using a DNN to compute the posterior probabilities, using i-vectors as input [16]. By computing SNR posteriors, the proposed SNR-dependent mixture of PLDA is able to achieve better performance than conventional PLDA and mixture of PLDA under noisy environments. However, noise is not the only obstacle to construct a robust speaker verification system.

1.3 Domain Mismatch in Speaker Recognition

Besides speaker information, genders and languages are another two crucial characteristics of human voice. Male and female possess different vocal-tract structures, which induce different voice characteristics for the two genders [17, 18]. For example, the pitch frequency of female is typically higher than that of male. Moreover, even for for the same vowel, the formant frequencies of female are higher than those of male. However, a majority of speaker verification systems in the literature are gender-dependent, mainly because the speech corpora for speaker recognition research have gender labels. This means that two gender-dependent systems are trained separately. If gender information is not available during scoring, a gender classifier can be used as a front-end for the gender-dependent systems. Again, i-vectors can be used as features because they contain gender information. For example, in [19], a PLDA model was used as the backend for i-vector based gender classification, which achieves an accuracy of 97.63% on the Fisher English corpus . However,

for the severely distorted RATS corpus, the accuracy drops to 76.48%. Such a low accuracy will certainly affect the performance of the gender-dependent systems. A better approach is to jointly train the gender-dependent PLDA models using the data from both genders. This leads to a gender-independent PLDA mixture model, which is the key contribution of this dissertation.

Besides gender variability, speaker verification systems also need to deal with language and channel mismatch. In particular, a system trained in one language (e.g, English) will have difficulty in distinguishing speakers speaking another language (e.g, Mandarin).

To address the domain mismatch problem, Garcia-Romero and McCree [20] proposed to estimate the within-speaker and between-speaker variability by treating them as random variables and used maximum *a posteriori* (MAP) adaptation to compute these parameters on the basis of labelled in-domain data. These covariances can also be treated as latent variables [21] whose joint posterior distribution can be factorized by using the variational Bayes method. Thus, the point estimates for scoring the in-domain data are computed from the factorized distribution. These earlier methods perform *supervised* domain adapation because they require speaker labels in the in-domain training data. Domain mismatch can be further reduced under some critical conditions like [22] which proposed to a framework to suppress mismatch without sufficient channel information.

One approach to dealing with unlabelled data is to hypothesize speaker labels by performing unsupervised clustering of in-domain data [23, 24]. With the hypothesized speaker labels, an in-domain PLDA model can be trained. An adapted PLDA model can be obtained by interpolating the covariance matrices of the out-of-domain PLDA model and the in-domain PLDA model [23]. The drawback of the clustering approach is that the number of speakers in the in-domain data is usually unknown.

Another way to perform unsupervised adaptation is to find a domain-invariant space from a number of datasets, with each dataset collected from one domain. For example, Aronowitz [25, 26] proposed an inter-dataset variability compensation (IDVC) algorithm to reduce the mismatch among datasets. The algorithm is further extended in [27]. IDVC assumes that within the i-vector space there is a low-dimensional subspace that is more sensitive to dataset mismatch. Therefore, the goal of IDVC is to find this subspace and remove it from all of the i-vectors. To find this subspace, IDVC either divides a big heterogeneous dataset into a number of source-dependent subsets or makes use of multiple datasets with each dataset represents one source. Another approach is to normalize the covariances of out-of-domain i-vectors [28], which has similar notation as within-class covariance normalization [29] but without using speaker labels. The authors in [28] named the method dataset invariant covariance normalization (DICN). Recently, Lin *et al.* [30, 31] showed that the maximum mean discrepancy (MMD) among multiple datasets can be used as a loss function for training an autoencoder so that domain-invariant i-vectors can be extracted from its middle layer. Unlike IDVC and DICN, the MMD loss can reduce domain mismatches beyond the second order.

Domain adversarial training (DAT) [1, 32] is a state-of-the-art domain adaptation method for domain adaptation. The method adversarially trains a set of networks comprising a feature extractor, a label predictor and a domain discriminator. The three components work cooperately but also challenge each other to form a domain-invariant space with maximum class information. In [33], Wang *et al*. demonstrated the effectiveness of domain adversarial training for speaker recognition through creating a domain-invariant and speaker-discriminative space. PLDA was used as the back-end to score the vectors extracted from the adversarial network. The results suggest that DAT outperforms other unsupervised do-

main adaptation methods including IDVC, DICN and matrix interpolation.

In this dissertation, we propose a contrastive adversarial domain adaptation network (CADAN) that utilizes adversarial learning to create a domain-invariant space with maximum speaker information. Features extracted from this space can replace the conventional i-vectors for speaker recognition. Unlike the conventional domain adversarial network (DAN), we separate the feature extractor in the DAN into two parts, one part for maximizing class information in the domain-invariant space and the other part minimizes the domain information. The weights of the two parts are separately updated to achieve these two contrastive goals. Also, unlike the conventional DAN in which the label predictor is trained to minimize class cross-entropy, we purposely weaken the capability of the label predictor in classifying speakers. This has the effect of forcing the feature extractor to work harder to produce more class discriminative features. Because our class-label predictor aims to make the life of the feature extractor harder as opposed to making it easier, we name it *fuzzifier*.

In addition to the comparison with the DAN, this dissertation also uses t-SNE plots to illustrate the domain-invariance and class discrimination of the embedded features created by the CADAN during the course of adversarial training. Experimental results on NIST 2012 SRE demonstrate that the CADAN can achieve nearly ideal domain adaptation for gender mismatch on speaker identification and outperforms state-of-the-art domain adversarial networks in both clean and noisy environments.

1.4 Thesis Organization

The remaining parts of this dissertation are organized as follows. Chapter 2 provides the background information on the machine learning methods used in this study. Chapter 3

explains how the gender- and language-independent PLDA mixture model leverages these machine learning methods to deal with the domain mismatches commonly encountered in practical situations. Chapter 4 explains the contrastive adversarial domain adaptation network and reports the performance of the network on NIST evaluation data. Chapter 5 concludes the study and Chapter 6 highlights some possible future directions.

Chapter 2

Background

2.1 I-vector and PLDA

I-vectors [5] are compact representations of utterances. To extract i-vectors from an utterance, we need to obtain a sequence of acoustic vectors (typically MFCCs [34] plus log-energy together with their first- and second-order derivatives) from the speech regions of the utterance. Then, we *align* the acoustic vectors with a universal background model (UBM), which is essentially a Gaussian mixture model (GMM) trained from the speech of many speakers. Here, the term "align" means computing the posterior probabilities of the Gaussian mixtures for each frame, from which the zero-th and first-order sufficient statistics of the whole utterance can be obtained. Using these sufficient statistics, the i-vector of the utterance can be computed from a factor analysis (FA) model.

The FA model is a generative model of the form:

$$\boldsymbol{\mu} = \boldsymbol{\mu}^{(b)} + \mathbf{T}\mathbf{w} \tag{2.1}$$

where $\boldsymbol{\mu}^{(b)}$ is the universal mean obtained by stacking the mean vectors of the UBM, $\mathbf{T}$ is a low-rank total variability matrix representing a subspace that comprise all sort of variabilities and $\mathbf{w}$ is the latent factor whose posterior mean is the i-vector. According to Eq. 2.1, the supervector $\boldsymbol{\mu}$ of every utterance can be *generated* by sampling an appropriate $\mathbf{w}$ from

the prior distribution of the latent factor $\mathbf{w}$. This is why Eq. 2.1 is a generative model. While $\boldsymbol{\mu}$ is conceptually important in the FA model, it does not need to be computed during the i-vector extraction process, which is elaborated below.

Suppose for each utterance, there is a set of observed acoustic vectors denoted as $\mathcal{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$ where T is the number of frames. The zeroth and the centered first-order sufficient statistics can be computed as follows:

$$n_k = \sum_{n=1}^{T} \Pr\left(C_n = k|\mathbf{o}_n\right) \tag{2.2}$$

$$\tilde{\mathbf{f}}_k = \sum_{n=1}^{T} \Pr\left(C_n = k|\mathbf{o}_n\right)\left(\mathbf{o}_n - \boldsymbol{\mu}_k^{(b)}\right), \tag{2.3}$$

where $C_n \in \{1, \ldots, M\}$ is the mixture index indicating which of the M Gaussians in the UBM is responsible for generating $\mathbf{o}_n$ and $\boldsymbol{\mu}_k^{(b)}$ is the k-th mean vector of the UBM. Given $\{n_k, \tilde{\mathbf{f}}_k\}_{k=1}^{M}$, the corresponding posterior covariance and posterior mean (known as i-vector) of $\mathbf{w}$ can be obtained by:

$$\text{Cov}(\mathbf{w}, \mathbf{w}|\mathcal{O}) = \mathbf{L}^{-1} \tag{2.4}$$

$$\langle \mathbf{w}|\mathcal{O}\rangle = \mathbf{L}^{-1}\mathbf{T}^{\mathsf{T}}\left(\boldsymbol{\Sigma}^{(b)}\right)^{-1}\tilde{\mathbf{f}} \tag{2.5}$$

where

$$\mathbf{L} = \mathbf{I} + \mathbf{T}^{\mathsf{T}}\left(\boldsymbol{\Sigma}^{(b)}\right)^{-1}\mathbf{N}\mathbf{T} \tag{2.6}$$

is a precesion matrix, $\mathbf{I}$ is an identity matrix and $\mathbf{N} = \text{diag}\{n_1\mathbf{I}, \ldots, n_M\mathbf{I}\}$. Note that $\tilde{\mathbf{f}}_s$ is a supervector obtained by stacking the centered first sufficient statistics. Note also that $\boldsymbol{\Sigma}^{(b)}$ obtained the residual covariance which cannot be captured by the total variability matrix $\mathbf{T}$. However, the residual covariance is obtained by stacking the diagonal covariance matrices of the UBM in diagonal form. The i-vector of an utterance can be computed from Eq. 2.5

which is the posterior mean of the latent factor.

Before applying the i-vectors to the PLDA model, they should be pre-processed by whitening and length normalization [35]. This pre-processing step makes the distribution of the i-vectors closer to a Gaussian distribution so that Gaussian PLDA (instead of heavy-tailed PLDA) can be used. Suppose we have a dataset comprising length-normalized D-dimensional i-vectors $\mathcal{X} = \{\mathbf{x}_{ij} \in \Re^D; i = 1, \ldots, N; j = 1, \ldots, H_i\}$ where N is the number of speakers and H_i is the number of sessions of speaker i. Denote the parameters of a PLDA model as $\boldsymbol{\omega} = \{\mathbf{m}, \mathbf{V}, \boldsymbol{\Sigma}\}$, where $\mathbf{m}$ is the global i-vector mean, $\mathbf{V} \in \Re^{D \times R}$ is a low-rank loading matrix, and $\boldsymbol{\Sigma}$ is the covariance of the residue not captured by $\mathbf{V}\mathbf{V}^\mathsf{T}$. Note that the PLDA model further reduces the dimension of the i-vectors from D to R. Then, the i-vectors in $\mathcal{X}$ obey the following generative model (which is also a factor analysis model):

$$\mathbf{x}_{ij} = \mathbf{m} + \mathbf{V}\mathbf{z}_i + \boldsymbol{\epsilon}_{ij}, \tag{2.7}$$

where $\mathcal{Z} = \{\mathbf{z}_i \in \Re^R, i = 1, \ldots, N\}$ are the latent variables and $\boldsymbol{\epsilon}_{ij}$ is a residue that follows a Gaussian distribution, i.e., $\boldsymbol{\epsilon}_{ij} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$.

To compute the PLDA parameters, the expectation-maximization (EM) algorithm can be used to iteratively maximize the likelihood of $\mathcal{X}$. The EM algorithm has 2 steps.

E-Step :

$$\mathbf{L}_i = \mathbf{I} + H_i \mathbf{V}^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \mathbf{V}$$

$$\langle \mathbf{z}_i | \mathcal{X} \rangle = \mathbf{L}_i^{-1} \mathbf{V}^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \sum_{j=1}^{H_i} (\mathbf{x}_{ij} - \mathbf{m})$$

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

M-Step :

$$\mathbf{m}' = \frac{\sum_{i=1}^{N} \sum_{j=1}^{H_i} \mathbf{x}_{ij}}{\sum_{i=1}^{N} H_i}$$

$$\mathbf{V}' = \left\{ \sum_{i=1}^{N} \sum_{j=1}^{H_i} (\mathbf{x}_{ij} - \mathbf{m}') \langle \mathbf{z}_i | \mathcal{X} \rangle \right\} \left[\sum_{i=1}^{N} \sum_{j=1}^{H_i} \langle \mathbf{z}_i \mathbf{z}_i^T | \mathcal{X} \rangle \right]^{-1}$$

$$\boldsymbol{\Sigma}' = \frac{1}{\sum_{i=1}^{N} H_i} \sum_{i=1}^{N} \sum_{j}^{H_i} \left[(\mathbf{x}_{ij} - \mathbf{m}')(\mathbf{x}_{ij} - \mathbf{m}')^{\mathsf{T}} - \mathbf{V}' \langle \mathbf{z}_i | \mathcal{X} \rangle (\mathbf{x}_{ij} - \mathbf{m}')^{\mathsf{T}} \right]$$

Note that the above EM algorithm is for estimating the parameters of a single PLDA model. Practically, PLDA is a data-driven algorithm which means that a large amount of labeled training is required for training.

2.2 Domain Mismatch in Speaker Recognition

Domain mismatch can severly degrade speaker recognition performance [20, 26]. There are various causes of domain mismatch, the most prominent being the discrepancy between the training and test environments arising from different channels, languages and genders. Among them, gender difference is one of the most severe and obvious mismatch due to

the physiological differences between male and female. A recent study demonstrated that speaker verification performance can be improved by predicting the gender of an unknown speaker followed by gender-dependent scoring. In another study [24], a DNN was used for computing the posterior probabilities of genders, which were then used as mixture posteriors in a PLDA mixture model. It was shown that although the gender information could not be perfectly predicted, it is helpful for the PLDA mixture model to score the i-vectors, resulting in performance superior to a gender-independent PLDA model.

2.2.1 Inter Dataset Variability Compensation

To address the domain mismatch, Aronowitz [25, 36] proposed the inter dataset variability compensation (IDVC) algorithm to reduce the mismatch between datasets. The algorithm is further extended in [27]. IDVC assumes that within the i-vector space there is a low-dimensional subspace that is more sensitive to dataset mismatch. Therefore, the goal of IDVC is to find this subspace and remove it from all of the i-vectors. To find this subspace, IDVC either divides a big heterogeneous dataset into a number of source-dependent subsets or makes use of multiple datasets with each dataset represents one source.

In the most basic form of IDVC, only the subset means of i-vectors are considered. Specifically, denote $\boldsymbol{\mu}_i$, $i = 1, \ldots, S$ as the mean vectors of the S datasets. Then, PCA is applied to these mean vectors, which results in a low-rank projection matrix $\mathbf{U}$ comprising r eigenvectors. In [36], these eigenvectors form the subspace called inter dataset variability subspace. Then, each i-vector $\mathbf{x}$ is subject to

$$\mathbf{x} \leftarrow (\mathbf{I} - \mathbf{U}\mathbf{U}^{\mathsf{T}})\mathbf{x},$$

where $\mathbf{I}$ is an identity matrix. This subspace removal process is applied to all i-vectors before length normalization [37] and the training of the final PLDA model.

2.2.2 Adaptation of PLDA Models

With the increasing demand on cross-domain speaker verification systems, Garcia-Romero *et al.* [38] are motivated to construct a domain adaptation framework for speaker verification systems. They developed a novel approach to adapting the out-of-domain PLDA parameters to the in-domain parameters:

$$\mathbf{\Gamma}_{adapt} = \alpha_1 \, \mathbf{\Gamma}_{in} + (1 - \alpha_1)\mathbf{\Gamma}_{out}$$

$$\mathbf{\Sigma}_{adapt} = \alpha_2 \mathbf{\Sigma}_{in} + (1 - \alpha_2)\mathbf{\Sigma}_{out},$$

where $\mathbf{\Gamma}$ is the across-speaker covariance matrix, $\mathbf{\Sigma}$ is the within-speaker covariance matrix, and α_1 and α_2 are control parameters. Given the adapted $\mathbf{\Gamma}_{adapt}$ and $\mathbf{\Sigma}_{adapt}$, an adapted PLDA model can be used for scoring in-domain data.

2.3 Spectral Clustering

Recently, clustering received growing attention due to its wide range of application. As one of the most traditional clustering algorithms, the K-means algorithm possesses the advantage of computation simplicity because each data point only requires to compare with a small number of cluster centers. However, a drawback of K-means is that it assumes that all of the clusters should follow a Gaussian distribution to justify the use of Euclidean distance. This means that for non-Gaussian distributed data, K-means will not be effective.

An alternative to K-means is the pairwise method [10, 12]. Instead of comparing each

data point with the hypothesized cluster means, pairwise methods compare each data point with all (or a subset) of the other data points in the training set to form a similarity matrix. A low-dimensional embedded subspace is then obtained by applying eigen-decomposition on the similarity matrix. K-means can then be easily applied to the low-dimensional vectors in the subspace. Spectral clustering [39, 40] is one of the most popular pairwise methods. The computational complexity of spectral clustering is quite high because each data point in the training set needs to compare with all other data points. Chen $et\ al.$ [11] proposed a parallel spectral clustering algorithm. To overcome the high computation complexity, Chen $et\ al$ utilized a sparse matrix to replace the original one by selecting the most informative data. Specifically, instead of computing the pairwise distances of all data points, for each data point, its k-nearest neighbors are considered. Usually, k should be a small number related to the total number of the data points.

Spectral clustering is applicable to speaker verification when we have lots of unlabeled speech data. Given a data set without speaker labels, we may turn a pairwise PLDA score matrix into a similarity matrix for spectral clustering. More indepth discussions and experimental results of spectral clustering can be found in Chapter 3.

2.4 Deep Neural Netowrks

Deep learning [41, 42] and deep neural networks (DNNs) are currently regarded as the most promising research areas in machine learning primarily due to their recent success in application domains. DNNs are inspired by the structure of human brain. Inside the human brain, billions of neurons are working together to process the signals received by our sensory organs. The signals are transported from layer to layer in the cerebral cortex. Deep neural networks attempt to simulate the same neural structure. The strength of the

connection between two neurons is represented by a connection weight, which stimulates the degree of neurotransmission. A neural network with one hidden layer is shown in Figure 2.1:

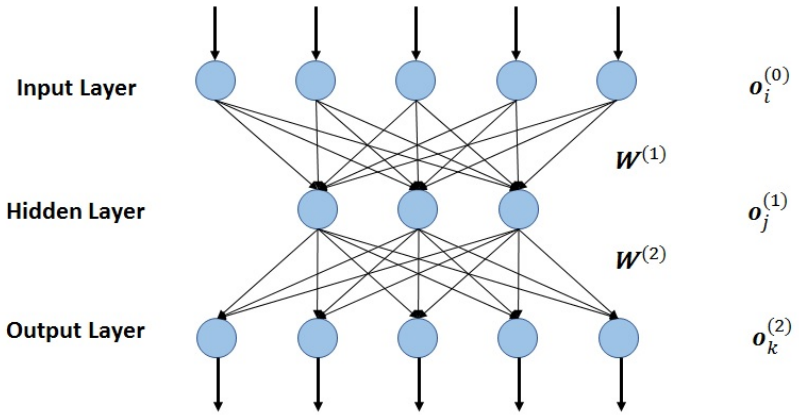


Figure 2.1: A neural network with one hidden layer

The weights of a DNN can be optimized (trained) by backpropagation (BP) [43]. BP is a gradient descent algorithm that aims to minimize the loss between the actual and desired outputs. There are several types of loss functions and the most common two are mean squared error (MSE) and cross entropy (CE). For the former, the total error E_{tot} is given by:

$$E_{tot} = \sum_n \sum_k (y_{n,k} - t_{n,k})^2, \tag{2.8}$$

where n indexes the training sample and k indexes the output nodes, $y_{n,k}$ and $t_{n,k}$ are the actual and target outputs, respectively. For the network in Figure 2.1, we further compute the error gradient with respect to $\mathbf{W}^{(2)}$:

$$\frac{\partial E_{tot}}{\partial w_{kj}^{(2)}} = \sum_n \frac{\partial E_n}{\partial w_{kj}^{(2)}}, \tag{2.9}$$

where $E_n = \sum_k (y_{n,k} - t_{n,k})^2$ is the instantaneous error. Using the chain rule and droping n for notation simplicity. Eq 2.9 can be written as:

$$\frac{\partial E}{\partial w_{kj}^{(2)}} = \frac{\partial E}{\partial a_k^{(2)}} \frac{\partial a_k^{(2)}}{\partial w_{kj}^{(2)}} = \delta_k^{(2)} o_j^{(1)}, \tag{2.10}$$

where $a_k^{(2)} = \sum_j w_{kj}^{(2)} o_j^{(1)}$ is the activation of the k-th neuron at the output layer and

$$\delta_k^{(2)} = \frac{\partial E}{\partial a_k^{(2)}} = \frac{\partial E}{\partial o_k^{(2)}} \frac{\partial o_k^{(2)}}{\partial a_k^{(2)}} = \left(o_k^{(2)} - t_k\right) \frac{\partial h(a_k^{(2)})}{\partial a_k^{(2)}}, \tag{2.11}$$

where $h(a)$ is a non-linear activation function. Similarly, we can use the same technique to compute the error gradient with respect to the hidden layer:

$$\frac{\partial E}{\partial w_{ji}^{(1)}} = \frac{\partial E}{\partial a_j^{(1)}} \frac{\partial a_j^{(1)}}{\partial w_{ji}^{(1)}} = \delta_j^{(1)} x_i, \tag{2.12}$$

where

$$\delta_j^{(1)} = h'(a_j^{(1)}) \sum_k \delta_k^{(2)} w_{kj}^{(2)}. \tag{2.13}$$

In summary, the weights of the output layer and hidden layer can be updated as follow:

- Output layer: $w_{kj}^{(2)} \leftarrow w_{kj}^{(2)} - \eta(y_k - t_k) h'(a_k^{(2)}) o_j^{(1)}$
- Hidden layer: $w_{ji}^{(1)} \leftarrow w_{ji}^{(1)} - \eta \left[h'(a_j^{(1)}) \sum_k \delta_k^{(2)} w_{kj}^{(2)} \right]$

where η is a small learning rate to ensure that the error steadily more towards a local minimum.

In the equation above, $h(a)$ is a non-linear activation function. Traditionally, the sigmoid function $h(a) = \frac{1}{1+e^{-a}}$ is used to as the activation function but recent research has found that other non-linearity such as rectified linear unit (ReLU), hyperbolic tangent and softplus lead to better performance. In particular, the ReLU is very efficient to compute and DNNs that use ReLU do not suffer from the gradient vanishing and exploding problems [44].

For very deep neural networks, the original method may encounter the gradient vanishing problem, especially the sigmoid non-linearity is used. To avoid gradient vanishing, Hinton $et\ al$ [45,46] proposed a learning algorithm called contrastive divergence to train restricted Boltzmann machines, which are very suitable for initializing the weights of DNNs. The development of this pre-training technique is a major milestone that leads to the recent advance in deep learning and artificial intelligence.

2.5 Adversarial Learning

Adversarial learning can be applied to train a domain adversarial network (DAN) to create a domain-invariant space [1] or to align the class distributions of the source task and the target task. Unlike GANs, the DAN does not have random inputs; instead, they either receive domain-dependent feature vectors as inputs or receive simultaneously the features from both the source and target domains. Their goal is to create a representation with minimum domain dependence. The domain adversarial network in [1] incorporates adversarial learning into deep neural networks by creating a latent space in which the domain discrepancy is suppressed while the class-dependent information is maintained.

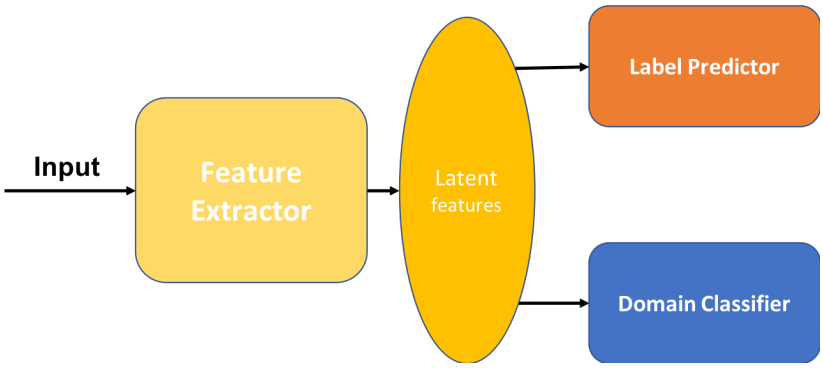


Figure 2.2: The structure of a domain adversarial netowrks (DAN)

Fig. 2.2 shows the structure of a DAN. It comprises three components: a feature extractor, a domain discriminator, and a label predictor. During training, the feature extractor and the label predictor are jointly trained to minimize the cross-entropy errors in the label predictor's output. Also, the feature extractor is jointly trained with the domain discriminator. But unlike the feature-extractor–label-predictor combination, for the feature-extractor–domain-discriminator combination, the feature extractor is adversarially trained so that the resulting features maximize the loss of the domain discriminator. The adversarial learning algorithm is like a two-player game in which the feature extractor is trained to confuse the domain discriminator that is tuned to distinguish the target domain from the source domain. The designate output at the intermediate layer of the domain-adversarial network is not only domain-invariant but also class discriminative.

Chapter 3

Semi-supervised Domain Adaptation for Gender-independent Speaker Verification

Because the totally variability matrix of an i-vector extractor is trained from speech with all sort of variabilities (e.g., speakers, noise, channels, genders and languages), an i-vector contains not only speaker information but also other information that may be detrimental to speaker verification. Therefore, it is crucial to develop a robust speaker verification system that is robust to these undesirable variabilities.

Before 2016, the NIST Speaker Recognition Evaluations (SRE) focused mainly on the channel and noise variabilities. In these evaluations, the development and test data contain mostly English telephone conversations. Because all of the development data have speaker labels, there are abundant data for training the PLDA models, which result in superb performance. In 2016 SRE, however, the i-vector/PLDA framework faces a big challenge. This is because not only the development (dev) and evaluation (eval) data are collected outside north America, but also the speakers speak different languages other than English. These differences cause serious dataset and language mismatches if a PLDA model trained from the data in pre-2016 SRE data and tested on the 2016 SRE data. Worse still, the evaluation data in 2016 SRE do not have gender labels, forcing the participants to develop gender-independent PLDA model or to detect the gender of the test utterances before computing the PLDA scores. This chapter describes the methods that we used to address these

problems.

3.1 Semi-supervised Domain Adaptation

To overcome the dataset and language mismatches, this dissertation proposes an semi-supervised domain adaptation method that leverages a gender-aware PLDA mixture model and the unlabelled development data in 2016 SRE. To guide the training of the mixture model, a gender-aware DNN is trained to estimate the gender posteriors given i-vectors as input. These posteriors replace the posteriors of the indicator variables in the mixture model during both training and scoring. The training procedure is iterative in that the PLDA mixture model is "initialized" by using pre-2016 SRE data using both the gender and speaker labels. Then, the mixture model is applied to compute the pairwise PLDA scores of the unlabelled data in the development set of 2016 SRE, followed by applying spectral clustering on the scoring matrix to produce a set of hypothesized speaker labels. These labels, together with the gender labels produces by the DNN, are used for inter dataset variability compensation (IDVC) to account for the dataset shift. The IDVC-compensated i-vectors are then used for retraining the PLDA mixture model, and the process is repeated. Figure 3.1 shows the block diagram of the training process.

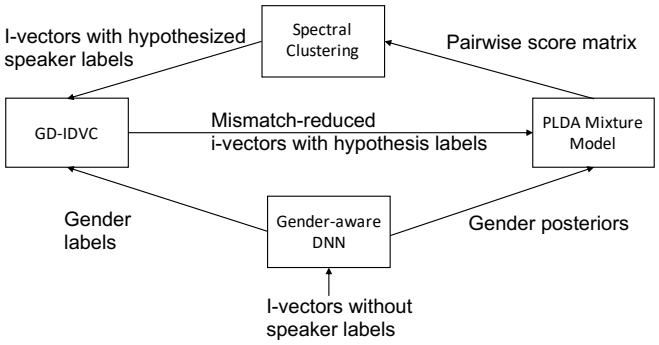


Figure 3.1: The flowchart of the semi-supervised domain adaptation method that addresses the domain and language mismatches.

3.1.1 Hypothesized Speaker Labels

Because speaker labels are not always available in the development datasets, clustering is a sensible strategy to produce hypothesized speaker labels for training the PLDA models. Spectral clustering [47], which makes use of the pairwise similarities of data, is one of the most effective clustering methods. In the proposed system, spectral clustering is the key-step for iteratively training a PLDA mixture model. To perform spectral clustering, we need a similarity matrix comprising the pairwise similarity between the training i-vectors. The similarity matrix can be obtained from the PLDA scores of training utterances. As PLDA scores are log-likelihood ratios, they can be negative. Therefore, we need to convert the PLDA scores to similarity scores that are amenable to spectral clustering.

Given a dataset $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ comprising N i-vectors, we compute a PLDA score matrix $\mathbf{S} \in \Re^{N \times N}$, where the element s_{ij} of $\mathbf{S}$ is the score of $\mathbf{x}_i$ and $\mathbf{x}_j$ based on a PLDA mixture model:

$$s_{ij} = S_{\text{mPLDA}}(\mathbf{x}_i, \mathbf{x}_j) \qquad i \neq j$$

Then, we convert $\mathbf{S}$ to a distance matrix $\mathbf{M}$ with elements:

$$m_{ij} = \begin{cases} s_{\max} - s_{ij} & i \neq j \\ 0 & \text{otherwise,} \end{cases} \tag{3.1}$$

where

$$s_{\max} = \max_{i,j;i\neq j} |s_{ij}|. \tag{3.2}$$

Then, we convert the distance matrix $\mathbf{M}$ to a similarity matrix $\mathbf{A}$ that is suitable for spectral clustering. Specifically, the elements of $\mathbf{A}$ are

$$a_{ij} = \exp\left\{-\frac{m_{ij}^2}{2\sigma^2}\right\}, \qquad 1 \leqslant i, j \leqslant N, \tag{3.3}$$

where σ is a scaling parameter that controls how fast the similarity drops with the distance m_{ij}.

This method is quite reasonable because the similarity of two i-vectors reflects the "distance" or difference between the two utterances. A negative s_{ij} means that the two i-vectors are very dissimilar, which results in a large m_{ij} in Eq. 3.1 and small a_{ij} in Eq. 3.3. On the other hand, a large s_{ij} means that the two i-vectors are very similar, which results in $m_{ij} \approx 0$ and $a_{ij} \to 1$.

With the simiarlity matrix $\mathbf{A}$, we may divide $\mathcal{X}$ into k clusters as follows. First, we compute the Laplacian matrix:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}, \tag{3.4}$$

where $\mathbf{I}$ is an $N \times N$ identity matrix, $\mathbf{D}$ is a diagonal matrix with diagonal elements:

$$d_{ii} = \sum_{j=1}^{N} a_{ij} \tag{3.5}$$

and $\mathbf{D}^{-\frac{1}{2}}$ stands for the inverse of the square root of $\mathbf{D}$.

Then, we compute the K eigenvectors $\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(K)}\}$ of $\mathbf{L}$ with the smallest eigenvalues and pack the K eigenvectors to form a matrix $\mathbf{V} = [\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(K)}] \in \Re^{N \times K}$, followed by renormalization of the rows:

$$v_i^{(k)} \leftarrow \frac{v_i^{(k)}}{\sqrt{\sum_{k=1}^{K} (v_i^{(k)})^2}}. \tag{3.6}$$

Then, we consider the rows of the normalized $\mathbf{V}$ as K-dimensional vectors and the N row vectors can be clustered by K-means to form k clusters. The row vectors and their corresponding utterances in the c-th cluster ($c = 1, \ldots, k$) are considered to be associated with the c-th hypothesized speaker.

3.1.2 Gender-Independent Mixture of PLDA

Gender information is critical for constructing speaker verification systems. The DNN-driven mixture of PLDA in [16, 48] perfroms very well when the test utterances have a wide range of SNR. However, the method is gender-dependent in that it uses the gender information in the training set to separately train two gender-dependent PLDA mixture models. To convert the gender-dependent system to a gender-independent one, we may pool the speech of both genders to train a gender-independent PLDA model. However, this naive approach is undesirable because there are fundamental differences in the vocal-tract

length, vocal-fold size, and larynx anatomy between the two genders. Studies have shown that there are substantial differences between the acoustic features extracted from male and female speakers [49]. These findings together with the limitation of gender-dependent systems motivate us to develop gender-independent mixture of PLDA. In particular, we propose to use a gender-aware deep neural network to guide the training of the mixture of PLDA by replacing the mixture posteriors with the gender posteriors estimated by the DNN. Because there will only be two genders, the number of mixtures is always two. In the verification stage, given the i-vectors of the target-speaker and a claimant, the gender posteriors given by the DNN are used as the linear combination weights (see Section 3.4) to compute the verification score.

Denote the weights of the DNN after training as $\mathbf{w}$. Given the i-vector $\mathbf{x}_{ij}$ of an utterance from the j-th session of the i-th speaker, the network produces the gender posterior:

$$\boldsymbol{\gamma}_{\mathbf{x}_{ij}}(y_{ijk}) = P(y_{ijk} = 1 \mid \mathbf{x}_{ij}, \mathbf{w}), \qquad k = 1, 2 \tag{3.7}$$

where $y_{ijk} \in \{0, 1\}$ is an indicator variable indicating whether the utterance of $\mathbf{x}_{ij}$ is spoken by a male speaker or a female speaker.

The gender-aware DNN not only provides guidance information to the mixture model but also generates gender labels for carry out gender-dependent IDVC. The predicted gender label is then given by

$$\text{Gender} = \begin{cases} \text{male} & \boldsymbol{\gamma}_{x_{ij}}(y_{ijk}) > 0.5 \\ \text{female} & \text{otherwise.} \end{cases} \tag{3.8}$$

3.1.3 Domain Adaptation

Due to the mismatches in languages and communication channels between the data in 2016 SRE and pre-2016 SREs, domain adaptation is crucial for systems that are trained on pre-2016 SRE data but tested on 2016 SRE data. There are two kinds of mismatches in NIST 2016 SRE: within dataset and across dataset. For the former, the datasets may be heterogeneous and comprise speech from different sources, e.g., different languages and genders. For the latter, the channel characteristics of the development and evaluation datasets could be very different, e.g., collected by different instruments in different telephone networks.

To suppress the effect of within-dataset mismatches, gender-dependent inter dataset variability compensation (GD-IDVC), incorporated with the DNN in Section 3.3, is applied to transform the i-vectors of both development and evaluation datasets. As discussed in Section 2.2, IDVC aims to find a low-dimensional subspace that is sensitive to the mismatches and remove this subspace from both the development and evaluation i-vectors. To this end, we partitioned the development dataset of 2016 SRE into 4 subsets (2 per gender) as shown in Table 3.1. To estimate the subspace S_μ, principal component analysis (PCA) is applied to find the eigenvectors that possess the K largest eigenvalues. The dimensionality of subspace S_μ was fixed to 3 in the proposed system. Due to the lack of in-domain data in SRE16, it is not effective to compute the subspaces corresponding to other PLDA parameters.

Group	Data	Mean i-vector
1	Male from major	$\boldsymbol{\mu}_1$
2	Male from minor	$\boldsymbol{\mu}_2$
3	Female from major	$\boldsymbol{\mu}_3$
4	Female from minor	$\boldsymbol{\mu}_4$

Table 3.1: The four partitioned subsets for GD-IDVC training. "Major" and "Minor" are the major and minor languages in 2016 SRE, respectively.

3.1.4 Mixture of PLDA and Likelihood Ratio Scores

The proposed gender-aware mixture of PLDA is trained based on the hypothesized speaker labels and gender posteriors derived in Eq. 3.7. The training process is depicted in Figure 3.2. The reason why we used a gender-aware network is that we need the network to compute the gender posteriors for modeling the i-vectors by mixture of PLDA. Therefore, instead of performing gender-classification (which gives binary classification decisions), we obtained the soft decisions from the gender-aware DNN. This results in gender-aware adaptation of the mixture PLDA model instead of adaptation of gender-dependent PLDA models.

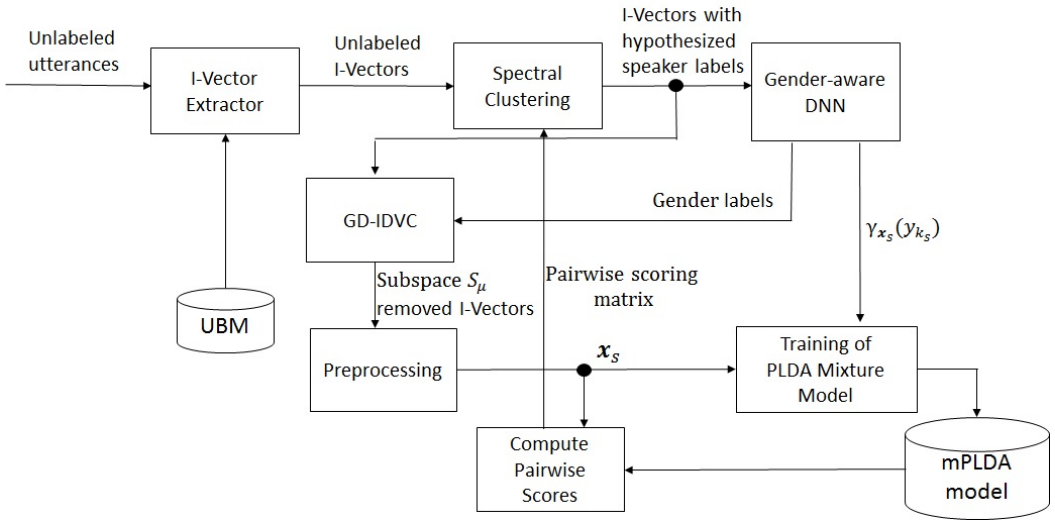


Figure 3.2: Training process of the PLDA mixture model

As shown in Figure 3.2, the gender posterior $\gamma_{\mathbf{x}_{ij}}(y_{ijk})$ of the i-th speaker, j-th session and k-th class (gender) from the gender-aware DNN guides the training process. The EM algorithm given in [48] is used to learn the parameters of the PLDA mixture model, which is parameterized by $\boldsymbol{\theta} = \{\mathbf{m}_k, \mathbf{V}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$, where K is equal to 2 because there are two genders only. To iteratively estimate and update the model parameters, the EM algorithm is applied for calculating the new set of parameters $\boldsymbol{\theta}'$. The E- and M-step are specified

below:

E-Step :

$$\mathbf{L}_i = \mathbf{I} + \sum_{k=1}^{K}\sum_{j=1}^{H_i}\langle y_{ijk}|\mathbf{x}_{ij}\rangle \mathbf{V}_k^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}\mathbf{V}_k$$

$$\langle \mathbf{z}_i|\mathcal{X}\rangle = \mathbf{L}_i^{-1}\sum_{k=1}^{K}\sum_{j=1}^{H_i}\langle y_{ijk}|\mathbf{x}_{ij}\rangle \mathbf{V}_k^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_{ij}-\mathbf{m}_k)$$

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

M-Step :

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

$$\mathbf{V}'_k = \left\{\sum_{i=1}^{N}\sum_{j=1}^{H_i}\left[\langle y_{ijk}|\mathbf{x}_{ij}\rangle(\mathbf{x}_{ij}-\mathbf{m}'_k)\langle \mathbf{z}_i|\mathcal{X}\rangle\right]\right\}\left[\sum_{i=1}^{N} N_{ik}\langle \mathbf{z}_i\mathbf{z}_i^{\mathsf{T}}|\mathcal{X}\rangle\right]^{-1}$$

$$\boldsymbol{\Sigma}'_k = \frac{1}{\sum_{i=1}^{N} N_{ik}}\sum_{i=1}^{N}\sum_{j=1}^{H_i}\left[\langle y_{ijk}|\mathbf{x}_{ij}\rangle(\mathbf{x}_{ij}-\mathbf{m}'_k)(\mathbf{x}_{ij}-\mathbf{m}'_k)^{\mathsf{T}} - \mathbf{V}'_k\langle \mathbf{z}_i|\mathcal{X}\rangle\langle y_{ijk}|\mathbf{x}_{ij}\rangle(\mathbf{x}_{ij}-\mathbf{m}'_k)^{\mathsf{T}}\right]$$

where $N_{ik} = \sum_{j=1}^{H_i}\langle y_{ijk}|\mathbf{x}_{ij}\rangle$. Note that the posteriors of the latent indicator variables $\langle y_{ijk}|\mathbf{x}_{ij}\rangle$ rely on the gender-aware DNN instead of the SNR-GMM as in [16] or SNR-DNN as in [15]. After the EM training, the new mixture PLDA model is used for computing a new pairwise score matrix, which is used for spectral clustering to renew the speaker labels of all i-vectors, and the process repeats.

To compute verification scores, the test dataset is firstly passed through the i-vector extractor, which is identical to that of the training process. After that, the gender-aware DNN computes the gender posteriors of the test set. Meanwhile, the i-vectors are subject to gender-dependent IDVC, using the same subspace as determined in the training process.

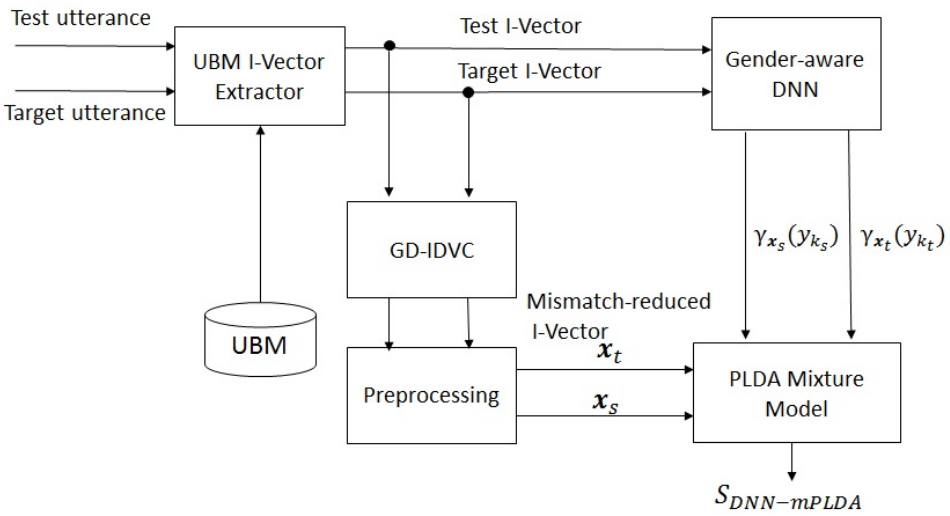


Figure 3.3: Scoring Process of the PLDA mixture model

As shown in Figure 3.3, the mismatch-reduced i-vectors are presented to the PLDA mixture model to compute the verification score:

$$S_{\mathbf{DNN-mPLDA}}(\mathbf{x}_s, \mathbf{x}_t) =$$

$$\ln \left\{ \frac{\sum_{k_s=1}^{K} \sum_{k_t=1}^{K} \gamma_{\mathbf{x}_s}(y_{k_s}) \gamma_{\mathbf{x}_t}(y_{k_t}) \mathcal{N}\left(\left[\mathbf{x}_s^{\mathsf{T}} \ \mathbf{x}_t^{\mathsf{T}}\right]^{\mathsf{T}} \mid \left[\mathbf{m}_{k_s}^{\mathsf{T}} \ \mathbf{m}_{k_t}^{\mathsf{T}}\right]^{\mathsf{T}}, \hat{\mathbf{V}}_{k_s k_t} \hat{\mathbf{V}}_{k_s k_t}^{\mathsf{T}} + \hat{\mathbf{\Sigma}}_{k_s k_t} \right)}{\left[\sum_{k_s=1}^{K} \gamma_{\mathbf{x}_s}(y_{k_s}) \mathcal{N}\left(\mathbf{x}_s | \mathbf{m}_{k_s}, \mathbf{V}_{k_s} \mathbf{V}_{k_s}^{\mathsf{T}} + \mathbf{\Sigma}_{k_s}\right)\right] \left[\sum_{k_t=1}^{K} \gamma_{\mathbf{x}_t}(y_{k_t}) \mathcal{N}\left(\mathbf{x}_t | \mathbf{m}_{k_t}, \mathbf{V}_{k_t} \mathbf{V}_{k_t}^{\mathsf{T}} + \mathbf{\Sigma}_{k_t}\right)\right]} \right\}$$

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

3.2 Experiments

3.2.1 Evaluation Protocol and Speech Data

Evaluations were performed on the evaluation set of NIST 2016 SRE (SRE16-eval). Data from the development set of SRE16 (SRE16-dev) and from SRE05–SRE12 were used for development. The data were divided into the following parts:

- *Enrollment and Test Data*: SRE16-dev has 120 enrollment segments, each with approximately 60 seconds. It also contains 1,207 test segments with duration ranging from 10 seconds to 60 seconds. All segments contain telephone conversations spoken by 20 subjects in either Mandarin or Cebuano. Each target speaker has one or three enrollment segments. The evaluation protocol in SRE16-dev defines which target-speaker models should score against which test segments, with a total of 4,829 target trials and 19,312 non-target trials. SRE16-eval has the same structure as SRE16-dev, excepting that the numbers of enrollment segments and test segments increase to 1,202 and 9,294, respectively. The number of subjects also increases to 201. The evaluation protocol defines 37,063 target trials and 1,949,666 non-target trials. Also, unlike SRE16-dev, all enrollment and test segments in SRE16-eval were spoken in either Cantonese or Tagalog, which causes language mismatch for systems trained on SRE16-dev data.
- *Development Data*: Telephone segments from SRE05–SRE12 were used for training the gender-aware DNN and the initial PLDA mixture model. The unlabelled data in SRE16-dev, including the major and minor languages, were used for training the subspace projection matrices (LDA and WCCN), a 512-mixture UBM, and a 300-factor total variability matrix. They were also used for the iterative retraining of the

PLDA mixture model.

For each speech segment, a 2-channel voice activity detector was applied to remove silence regions. Then, the speech regions were segmented into 25-ms Hamming windowed frames with 10ms frame shift. For each frame, 19 Mel frequency cepstral coefficients and log energy together with their first and second derivatives are packed to form a 60-dimensional acoustic vector, followed by cepstral mean normalization and feature warping [50] with a window size of 3 seconds.

3.2.2 Training of Gender-Aware DNN

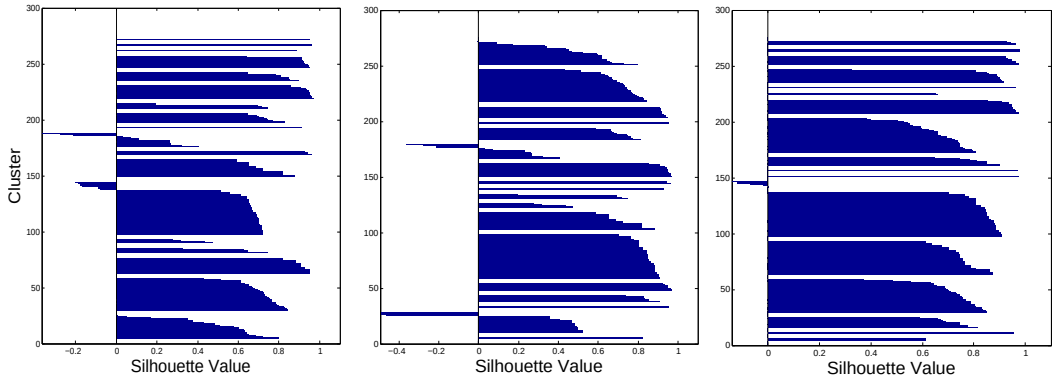
The DNN was constructed by stacking a number of restricted Boltzmann machines (RBMs) [41], which were initialized layer-wise by the contrastive divergency algorithm [45]. After that, a softmax layer was placed on top of the network to ensure that the network can produce gender posteriors. Then, backpropagation was applied to minimize the cross-entropy between desired and actual outputs. In this work, we used the utterances in SRE05–SRE12 and their gender labels to train the gender-aware DNN.

3.2.3 Score Normalization

Adaptive score normalization can improve the performance of i-vector/PLDA systems on NIST 2016 SRE significantly. To reduce scoring time, we applied adaptive z-norm instead of the more computationally demanding adaptive s-norm as a compromise. Specifically, we used the unlabelled utterances in SRE16-dev as the candidate cohorts for the enrollment utterances. For each enrollment utterance, its PLDA scores with respect to the unlabelled i-vectors in SRE16-dev were computed and ranked; then, the top-200 i-vectors were selected as the cohort set for computing the z-norm parameters of the utterance.

3.3 Results and Discussions

To compare the quality of the i-vector clusters produced by agglomerative hierarchical clustering (AHC) and iterative spectral clustering (Iterative-SC), we computed the silhouette values from the clusters produced by these two methods and displayed them as silhouette plots in Fig. 3.4. As AHC can use Euclidean or cosine distance as its distance metric, we refer to the resulting methods as Euclidean-AHC and Cosine-AHC, respectively. Fig. 3.4 shows that Iterative-SC has the highest average silhouette score and has less negative silhouette values. This suggests that Iterative-SC produces clusters with better quality.



(a) avgscore = 0.650 (b) avgscore = 0.668 (c) avgscore = 0.752

Figure 3.4: Silhouette plots showing the quality of i-vector clusters produced by (a) Euclidean-AHC, (b) Cosine-AHC and (c) Iterative SC. Each silhouette pattern represents a cluster, and the silhouette values of individual samples are shown on the horizontal axis.

We used equal error rate (EER) and minimum decision cost function (minDCF) defined in NIST 2016 SRE to evaluate the performance of different systems. Unless stated otherwise, the number of clusters (hypothesized speakers) is 180.

	SRE16-Dev		SRE16-Eval	
No. of Iterations	EER(%)	minDCF	EER(%)	minDCF
1	17.12	0.812	18.72	0.952
2	16.31	0.789	15.32	0.883
3	15.79	**0.751**	13.62	0.829
4	15.68	0.774	12.79	0.798
5	**15.04**	0.799	**12.73**	**0.779**
6	15.74	0.782	13.03	0.792
7	15.79	0.788	13.34	0.801

Table 3.2: Performance of the iterative retraining method for different numbers of iterations on SRE16-dev and SRE16-dev.

Table 3.2 shows that the performance generally improves after a few iterations on both datasets. Because of the mismatch between pre-SRE16 and SRE16 data, the performance in the first iteration is the worst. However, when the number of iterations was increased, the PLDA mixture model gradually adapts to the new domain and both the EER and minDCF drop. We observed that increasing the number of iterations beyond 7 does not bring any further peformance improvement.

In the next experiment, we compared different speaker clustering methods and used AHC as the baseline. Also, we used covariance matrix interpolation as the baseline. Specifically, we interpolated the covariance matrices of the in-domain PLDA mixture model with the covariance matrices of the out-of-domain PLDA mixture model using an interpolation weight of 0.5. Table 3.3 shows the speaker verification performance using the 3 speaker clustering methods. Note that iterative retraining (Fig. 3.2) is meaningful to Iterative-SC

only because the distance metrics of AHC is independent of the PLDA model. Results show that iterative-SC together with the retraining strategy can leverage the limited amount of unlabelled in-domain data to achieves superior performance. Rows 2 and 3 in Table 3.3 suggest that without iterative re-training, covariance interpolation helps to lower the EER and minDCF. However, when iterative re-training is applied (Row 4 and Row 5), the benefit of covariance interpolation diminishes.

Row	Clustering Method	Cov. Interp.	SRE16-Dev		SRE16-Eval	
			EER (%)	minDCF	EER (%)	minDCF
1	Euclid-AHC	N	19.54	0.937	18.68	0.932
2	Cosine-AHC	N	18.23	0.862	16.37	0.846
3		Y	16.36	0.818	14.12	0.832
4	Iterative-SC	N	**15.04**	**0.799**	12.73	**0.779**
5		Y	15.21	0.809	**12.60**	0.816

Table 3.3: Performance of PLDA mixture models on SRE16 using different speaker clustering methods and with and without covariance matrix interpolation (Cov. Interp.).

In the covariance interpolation method [20], the out-of-domain data have a *direct* influence on the adapted model and the degree of influence is controlled by an interpolation weight. The problem is that this weight should be set according some prior knowledge about the two domains, which may not be easily quantified. In our method, however, such influence will be progressively diminished during the iterative training process. As shown in Table 3.2, the PLDA model can be fully adapted to the new domain after 5 iterations.

Chapter 4

Contrastive Adversarial Domain Adaptation Networks for Speaker Recognition

The main challenge in domain adaptation is that we need to minimize the domain information in feature vectors without affecting their class information. We propose a new domain adversarial network called contrastive adversarial domain adaptation network (CADAN) to meet this challenge. This chapter explains the design philosophy, architecture and training algorithm of the CADAN.

4.1 Design Philosophy and Network Architecture

In the original DAN (see Section 2.4), the feature extractor is particularly hard to train because it needs to produce features that meet two contrastive objectives: maximum class discrimination and minimum domain dependency. In practice, its weights are tuned to meet the first objective but will be re-adjusted to meet the second one in the same epoch. This is in analogy to asking a person to learn two different but related tasks at the same time, which of course will not be as effective as learning one task at a time. While we may change the training strategy so that the two tasks can be learned consecutively, it is also undesirable because the network may forget the first task after learning the second one. A better approach is to delegate some task-specific neurons for the respective tasks. To this end, we propose splitting the middle hidden layers of the feature extractor network into

two branches so that they become partially decoupled from each other during adversarial training. In spite of the decoupling, the two sub-networks need to cooperate with each other because for each input vector, the feature extractor needs to produce one embedded feature vector as output. Therefore, the two branches share the input layer and the output layer. The architecture is shown in Fig. 4.1.

In addition to the contrastive feature extractor, another key difference between the proposed architecture in Fig. 4.1 and the DAN of [1] is the label predictor. In DAN, the feature extractor and label predictor are jointly trained to minimize the cross-entropy of the target classes. However, in the proposed architecture, the class encoder is trained to minimize the cross-entropy but the label predictor is trained to produce equal outputs (posterior probabilities). Therefore, instead of making the predictor more capable of classifying the latent feature vectors, we make it less capable of doing so. From the label predictor perspective, the latent features become *fuzzier* after every epoch. The deliberately weakening of the label predictor will encourage the class encoder in Fig. 4.1 to try harder to produce more speaker discriminative features so that they can be discriminated correctly by the adversarially trained label predictor. Because the label predictor is adversarially trained, the embedded features become more confusable to the label classifier. Therefore, we refer to the label classifier as "Fuzzifier".

In all, there are two intrinsic drawbacks of the original DAN. First, in the original DAN, the generator is designed for two different purposes: generating domain-invariant latent features and keeping speaker information. During training, the weights in the generator are updated twice for each epoch. However, each update is in contradiction to each other because of the different purposes. Second, in the original DAN, the classifier aims to classify the latent features into different speakers in the training set, which means that

it will contain some useful speaker information. Unfortunately, this information will not be encoded in the latent features, which will ultimately be used for speaker recognition. By replacing the speaker classifier with a speaker fuzzifier and splitting the generator into a class encoder and a domain suppressor, we can force the class encoder to encode all of the speaker information, which results in more speaker discriminative features in the latent vectors.

In the proposed approach, the feature extractor G is split into a domain suppressor G_{dom} and a class encoder G_{cls}. As shown in Fig. 4.1, the neurons in the feature extractor are separated into the blue group G_{cls}, which is to be trained with the fuzzifier F to maximize class discrimination, and the green group, which is to be trained with the domain discriminator. Because of the different objectives when training the weights (blue) for encoding class-discriminative information and the weights (green) for domain discrimination, both G_{cls} an G_{dom} become better in performing their respective task. Without the separate structure, training will become unstable if the weights are updated twice for different purposes in each epoch.

4.2 Training Algorithm

The training of F and G_{cls} in Fig. 4.1 are as follows:

$$\text{Train } F : \min_{F} \left\{ -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[\sum_{k=1}^{K} \frac{1}{K} \log F(G(\mathbf{x}))_k \right] \right\} \tag{4.1a}$$

$$\text{Train } G_{\text{cls}} : \min_{G_{\text{cls}}} \left\{ -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[\sum_{k=1}^{K} y_{\text{cls}}^{(k)} \log F(G(\mathbf{x}))_k \right] \right\}, \tag{4.1b}$$

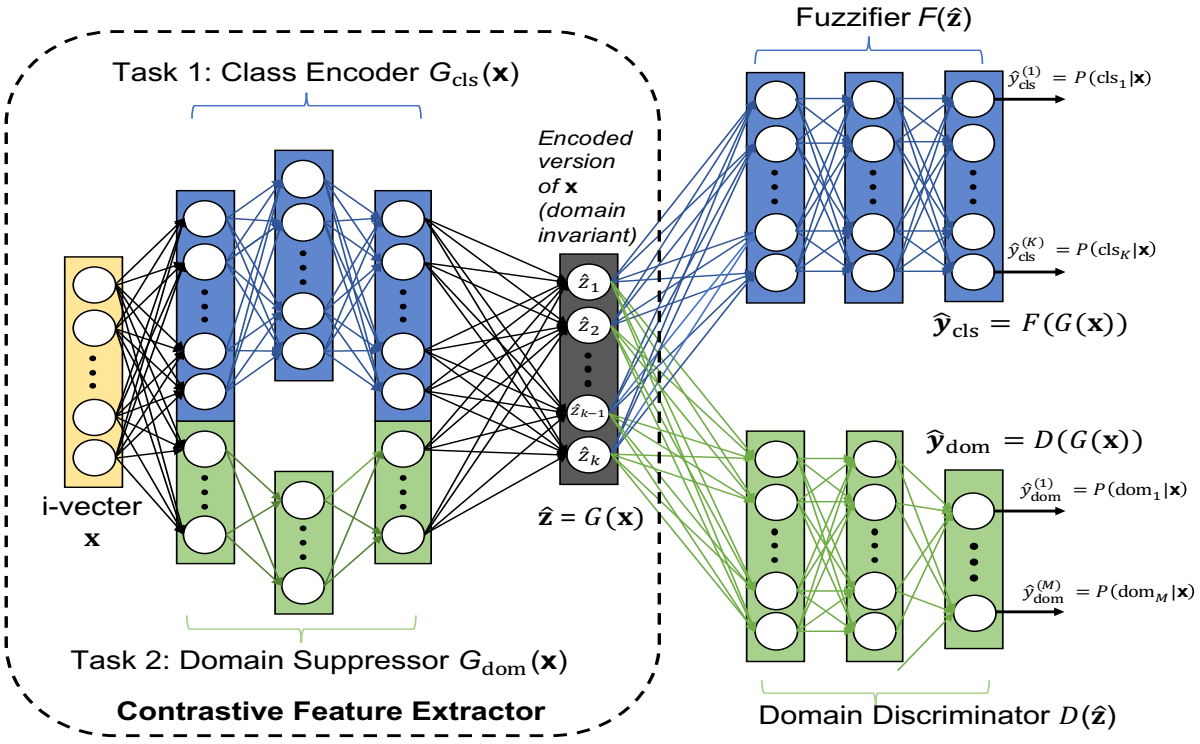


Figure 4.1: Contrastive Adversarial Domain Adaptation Networks (CADAN). The blue layers constitute the adversarial networks for enhancing class information and the green layers are responsible for reducing domain mismatch. The subscript "cls" and "dom" stand for class and domain, respectively.

where $G(\mathbf{x})$ is the output of the contrastive feature extractor, $F(\cdot)_k$ is the k-th output of the fuzzifier whose designated output is the posterior of class k, and $y_{\text{cls}}^{(k)}$ is equal to 1 if $\mathbf{x}$ comes from the k-th class; otherwise it is equal to 0. Unlike ordinary DAN in which the targets of the label predictor are in one-hot format, in CADAN, the targets of F in Eq. 3(a) are set to $[\frac{1}{K}, \ldots, \frac{1}{K}]^{\mathsf{T}}$. It can be shown that the minimum of the cross-entropy in Eq. 3(a) occurs when $F(G(\mathbf{x}))_k = \frac{1}{K}$ for all k. When this happens, the encoded vectors $\hat{\mathbf{z}} = G(\mathbf{x})$'s will be most confusable to the fuzzifier. During the course of training, the classification ability of F will keep on weakening. The weak F will make the class encoder G_{cls} to work

harder to produce class-discriminative features to reduce the cross-entropy in Eq. 3(b).

The encoder G_{dom} in Fig. 4.1 aims to make the embedded vectors $\hat{\mathbf{z}}$'s domain invariant. This can be achieved by the following optimizations:

$$\text{Train } D : \min_{D} \left\{ -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[\sum_{m=1}^{M} y_{\text{dom}}^{(m)} \log D(G(\mathbf{x}))_m \right] \right\} \tag{4.2a}$$

$$\text{Train } G_{\text{dom}} : \min_{G_{\text{dom}}} \left\{ -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[\sum_{m=1}^{M} \frac{1}{M} \log D(G(\mathbf{x}))_m \right] \right\}, \tag{4.2b}$$

where $y_{\text{dom}}^{(m)} = 1$ when $\mathbf{x}$ comes from domain m; otherwise $y_{\text{dom}}^{(m)} = 0$. The weights of G_{dom} are updated to obtain a domain invariant space so that the encoded vectors $\hat{\mathbf{z}}$'s become confusable to the discriminator D. This is achieved by setting the target of D to $[\frac{1}{M}, \ldots, \frac{1}{M}]^{\mathsf{T}}$. The domain discriminator D is trained to best differentiate these confusable vectors into different domains. Algorithm 1 shows the pseudo-code of the training of the CADAN. Note that training algorithm sequentially estimates the domain discriminator, domain suppressor, class encoder and fuzzifier in a learning epoch. The class encoder is updated with R steps in an epoch.

4.3 Experimental Setup

To evaluate the effectiveness of CADAN in suppressing domain mismatch, we applied it to a speaker identification task in which genders are considered as domains and speaker identities are considered as classes. Therefore, K and M in Fig. 4.1 correspond to number of speakers and number of domains, respectively.

4.3.1 Speech Data and Acoustic Features

Speech files from NIST 2004–2012 Speaker Recognition Evaluation (SRE04–12) were used as the training and test datasets. Babble noise was added to the speech files of SRE12 at an SNR of 6dB. Each dataset was first divided into male and female subsets. The speech files of each speaker were further split into training and test sets to ensure that the speakers in the test utterances must exist in the training set.

Because SRE04–12 contain telephone conversations and interviews, this way of splitting the data can also ensure that the contexts of the training utterances are totally different from those of the test utterances. A 2-channel voice activity detector (VAD) [51] was applied to remove silence regions. For each speech frame, 19 MFCCs together with energy plus their 1st and 2nd derivatives were computed, followed by cepstral mean normalization [34] and feature warping [50] with a window size of three seconds. A 60-dim acoustic vector was extracted every 10ms, using a Hamming window of 25ms.

4.3.2 I-Vector Extraction

A subset of the telephone and microphone speech files in SRE05–10 were used for training a gender-independent UBM with 1024 mixtures. Then, MAP adaptation [52] was applied to adapt the gender-independent UBM to gender-dependent UBMs using the speech files of the respective gender as adaptation data. For each gender, a 500-factor total variability (TV) matrix ($\mathbf{T}$) was estimated. The gender-dependent TV matrices and UBMs were used for extracting gender-dependent i-vectors. Using MAP adaptation to create gender-dependent UBMs can ensure that there is a one-to-one correspondence between their Gaussians, which in turn ensures that the GMM-supervectors of both genders ($\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}$) live in the same Euclidean space. As a result, the gender-dependent i-vectors also live in the same 500-

dimensional i-vector space.

4.3.3 Configuration and Training of DAN and CADAN

To ensure fair comparisons between DAN and CADAN, we kept their structure almost the same. Specifically, both of them have 500 input nodes, 3 hidden layers with 1,200 ReLU nodes in each layer, and 500 output nodes in the feature extractor. However, for CADAN, the 2nd hidden layer was split into two parts: 800 nodes for the class (speaker) encoder and 400 nodes for the domain (gender) suppressor. The ratio of 2:1 is motivated by the intuition that speaker information is more diverse than gender information, thereby requiring more nodes to encode. For both DAN and CADAN, the fuzzier and domain discriminator comprise two hidden layers, each with 500 ReLU nodes. The fuzzifier has 67 output nodes corresponding to 67 speakers and the domain discriminator has two output nodes.

We used the i-vectors of both genders in SRE04–10 to train a DAN and a CADAN. After training, we used the feature extractor network of the DAN and the contrastive feature extractor network of the CADAN to produce DAN- and CADAN-transformed vectors $\hat{\mathbf{z}}$ for both training and test i-vectors in the dataset. The training subset of the transformed vectors were then used for training gender-dependent PLDA models. The DAN and CADAN obtained by using SRE04–10 data correspond to the columns "SRE04–10" in Table 4.1 and Table 4.2. To investigate the behavior of DAN and CACAN under noisy environments, we have also used i-vectors extracted from noise-contaminated SRE12 utterances to train a DAN and a CADAN, and their performance is shown in the columns "Noisy SRE12" in Table 4.1 and Table 4.2.

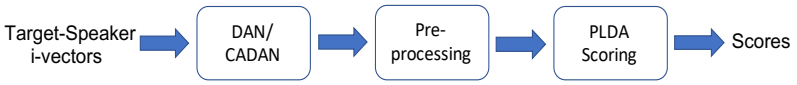


Figure 4.2: Transformation of i-vectors by the feature extractor of DAN or CADAN and PLDA scoring.

4.3.4 PLDA Training and Scoring

A pre-processing step was applied to the transformed vectors before they were used for training PLDA models. Specifically, the DAN- and CADAN-transformed vectors were subjected to within-class covariance normalization [53], length normalization [37], and linear discriminant analysis (LDA). The LDA reduces the dimension of the transformed vectors to 200. The WCCN and LDA matrices are gender-dependent and were estimated from the i-vectors in SRE05–10. Similarly, the WCCN and LDA matrices for "Noisy SRE12" were obtained from the i-vectors of noise contaminated speech in SRE12. The pre-processed vectors were then used for training condition-dependent (clean or noisy) and gender-dependent PLDA models.

In the testing phase, test i-vectors were transformed by the feature extractors of DAN and CACAN, respectively, followed by WCCN, length normalization and LDA. The test i-vector pairs were then passed to the PLDA model for scoring. Fig 4.2 shows the DAN/-CADAN transformation, vector pre-processing and PLDA scoring.

Because each test speaker has multiple sessions (i-vectors), the speaker ID of each test i-vector was identified based on the maximum average PLDA scores with respect to all speakers in the dataset.

4.4 Results and Discussions

4.4.1 Comparing DAN and CADAN

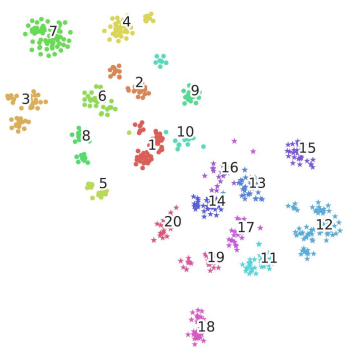
While the DAN and CADAN were trained on the speech (i-vectors) of both genders, the UBMs, T-matrices, and PLDA models are gender-dependent. With these gender-dependent PLDA models, we could have three kinds of experiments: (1) same-gender, (2) cross-gender, and (3) mix-gender.

1. *Same-gender Experiments*. The PLDA models were trained and scored on the DAN- and CADAN-transformed i-vectors derived from the same gender.

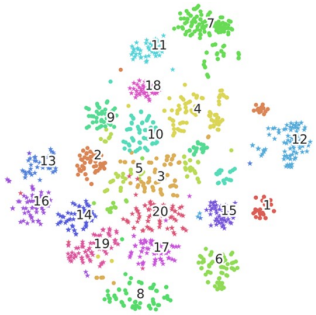
2. *Cross-gender Experiments*. The male PLDA models were tested on female vectors and vice versa for the female PLDA models.[1]

3. *Mix-gender Experiments*. The gender-dependent PLDA models were tested on the vectors from both genders.

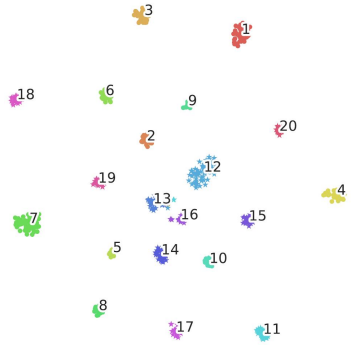
[1]It is possible to do this because the PLDA model is only a scorer; it accepts two vectors as input and computes the score of these two vectors as output. Therefore, a male PLDA model can be used for scoring female i-vectors.



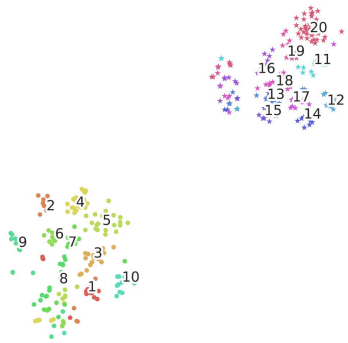
(a) Original i-vectors on SRE04–10



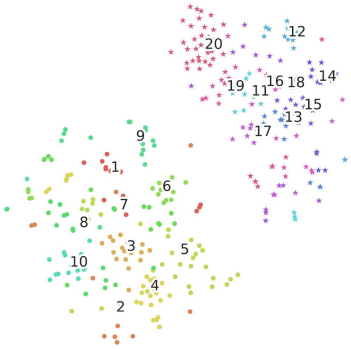
(b) DAN i-vectors on SRE04–10



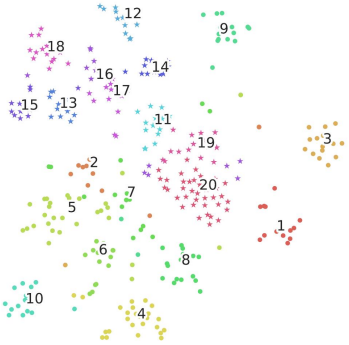
(c) CADAN i-vectors on SRE04–10



(d) Original i-vectors on noised SRE12



(e) DAN i-vectors on noised SRE12



(f) CADAN i-vectors on noised SRE12

Figure 4.3: t-SNE plots of raw i-vectors and DAN and CADAN transformed i-vectors derived from clean SRE04–10 and noisy SRE12. I-vectors were derived from the utterances of 10 male speakers (•) and 10 female speakers ($\star$). The numbers on top of each cluster are the speaker indexes (Speakers 1–10 are male and Speakers 11–20 are female) and each speaker is represented by one color. Note that the DAN- and CADAN- transformed i-vectors are of 500-dimensions which is the same as the dimension of raw i-vectors.

Table 4.1 shows the performance of the baseline (the row with label 'None'), DAN- and CACAN-transformed i-vectors. The baseline performance is based on an i-vector PLDA system in which the PLDA model was trained by raw i-vectors without domain adaptation.

Table 4.1 demonstrates that CADAN performs the best under the cross-gender scenario (out-of-domain columns) and performs well under the same-gender scenario (in-domain

		Out-of-Domain				In-Domain			
		SRE04–10		Noisy SRE12		SRE04–10		Noisy SRE12	
Gender of PLDA models		male	female	male	female	male	female	male	female
Gender of test i-vectors		female	male	female	male	male	female	male	female
I-Vector Transformation Method	None	0.6115	0.6281	0.5439	0.3248	0.8758	0.9375	**0.8619**	**0.6890**
	DAN	0.6337	0.6004	0.6364	0.3912	0.8417	0.9304	0.6932	0.5681
	CADAN	**0.6987**	**0.6723**	**0.7343**	**0.6556**	**0.8887**	**0.9468**	0.8307	0.6541

Table 4.1: Speaker identification accuracies on SRE04–10 and noisy SRE12 with and without i-vector transformation under gender-match and gender-mismatch scenarios. Out-of-domain (in-domain) means that the gender of PLDA models is the same as (different from) that of the test i-vectors. For the noisy SRE12, babble noise was added to the speech files of SRE12 at an SNR of 6dB.

		SRE04–10		Noisy SRE12	
Gender of PLDA models		male	female	male	female
Gender of test i-vectors		Both		Both	
I-Vector Transformation Method	None	0.6687	0.5770	0.6494	0.5391
	DAN	0.6512	0.6051	0.6264	0.5512
	CADAN	**0.7134**	**0.6823**	**0.6807**	**0.5691**

Table 4.2: Speaker identification accuracies on SRE04–10 and noisy SRE12 with and without i-vector transformation when the test i-vectors come from both genders but the PLDA model belongs to one gender only.

columns), although it is out-performed by the baseline under gender-match noisy conditions. The positive results reveal that contrastive-adversarial domain adaptation is capable of producing more effective features with rich speaker information. Under noisy scenario, the CADAN demonstrates superior performance in out-of-domain data by boosting the accuracy by 33%. It is noteworthy that a slight drop in accuracy is observed on the noisy in-domain data, which may be due to the severe domain mismatch under noisy conditions.

We further extended our experiments to gender-mixed scenarios, in which each PLDA model was trained by one gender only but tested on both genders. As shown in Table 4.2, CADAN performs the best under all conditions.

4.4.2 Visualization of CADAN

To investigate the hidden causes of the better performance achieved by CADAN, we used the t-SNE software to display the i-vectors in Fig. 4.3. The t-SNE plots of clean SRE04–10 reveal three interesting observations. (1) There is a significant gender mismatch between the i-vectors of male and female speakers, as evident by the clear gap in the middle of Fig. 4.3a and Fig. 4.3d that separates the two genders ($\bullet$ and $\star$). While Fig. 4.3a shows that the raw i-vectors do contain speaker information (as evident by the speaker clusters), some speakers such as Speakers 13, 14, and 16 are fairly confusable. (2) DAN is able to create a gender-invariant space, as evident by the absence of clear gap between the two genders in Fig. 4.3b. However, as compared to the raw i-vectors in Fig. 4.3a, the feature extractor of DAN removes some of the speaker information when it attempts to make the transformed i-vectors gender indistinguishable, as evident by the larger speaker clusters in Fig. 4.3b. This means that DAN is not able to maximize speaker information and minimize domain information *simultaneously*. (3) Compared with the raw and DAN-transformed i-vectors,

CADAN can produce i-vectors that possess the strongest discriminative information and simultaneously suppress domain information significantly, which result in highly compact speaker clusters in Fig. 4.3c.

Fig. 4.3d shows that noise has detrimental effect on i-vectors. It not only makes the gender gap bigger, but also increases the overlapping among speaker clusters. Under noisy environments, the domain (gender) mismatch is so severe that DAN can only reduce the gender gap but fails to create a domain invariant space, as shown in Fig. 4.3e. On the other hand, as shown in Fig. 4.3f, CADAN is not only able to create a domain-invariant space but also able to reduce the cluster overlapping. This ability makes CADAN significantly outperforms raw i-vectors and DAN-transformed i-vectors in Table 4.1 under the cross-gender scenario.

Fig. 4.4 shows the cross-entropy loss of DAN and CADAN during the course of training. The results clearly show that CADAN enjoys faster convergence, smoother training, and lower training error as compared to DAN.

4.4.3 Insights of Training Process

A deeper investigation was conducted to gain more insights into the training process of CADAN by plotting the intermediate transformed i-vectors at different training epochs in Fig. 4.5. At Epoch = 0, the weights of CADAN was initialized by the Xavier initializer, which leads to scattered i-vectors in Fig. 4.5a. When training progresses (Fig. 4.5b), the fuzzifier F and class encoder G_{cls} dominate the process by minimizing intra-speaker variability but domain mismatch remains intact. After producing a discriminative subspace, the domain discriminator D and domain suppressor G_{dom} work on pulling the male and female groups together. At this stage (Epoch = 150), Fig. 4.5c, the adapted subspace with discrim-

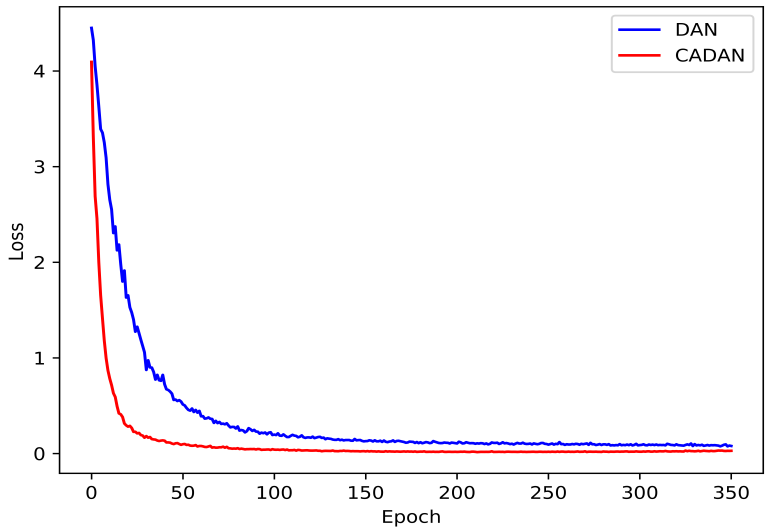
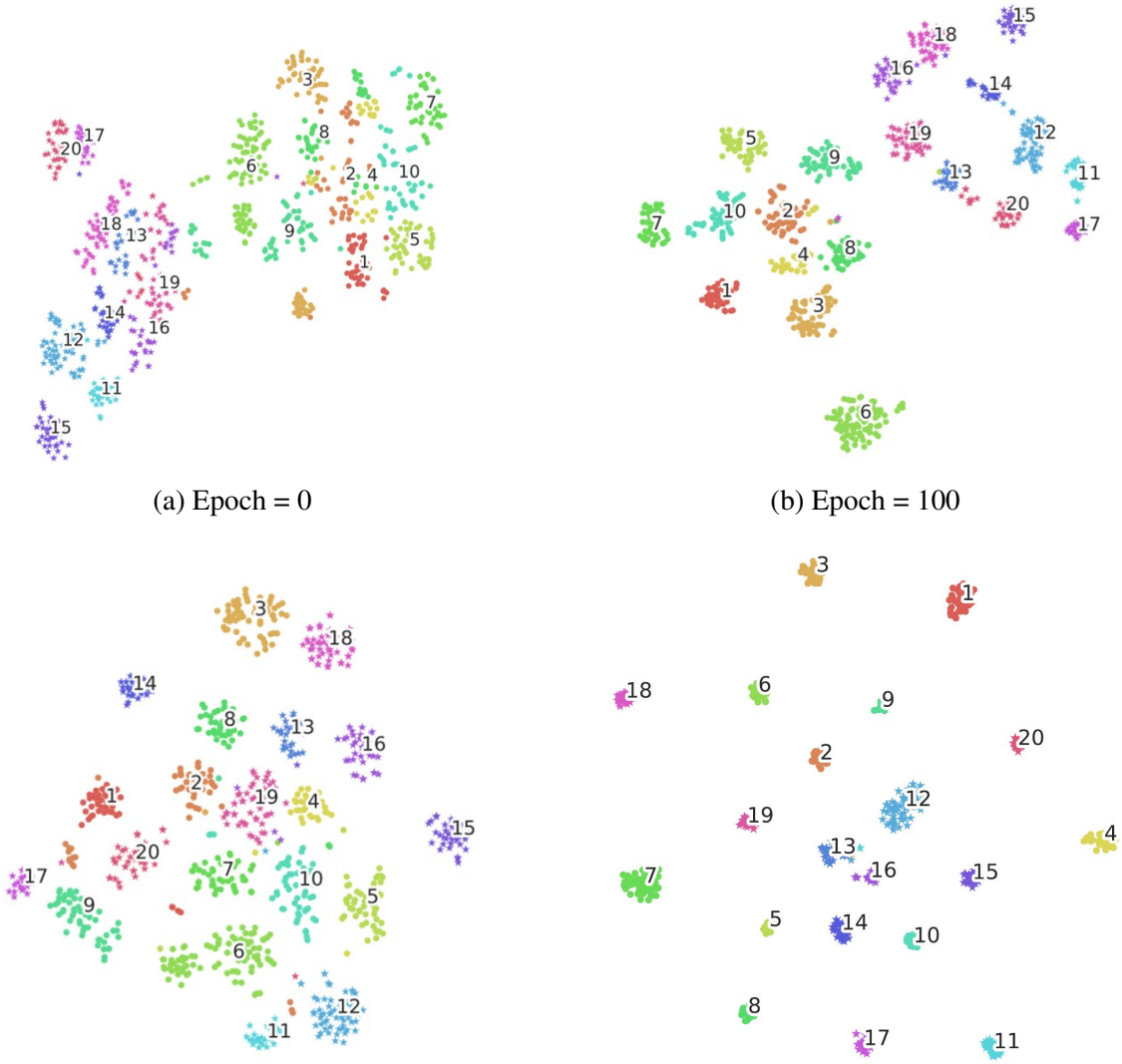


Figure 4.4: The cross-entropy loss of (1) the feature extractor cum the label predictor in DAN [1] and (2) the class encoder $G_{\text{cls}}(\mathbf{x})$ cum the fuzzier F in CADAN. Identical learning rate (0.001) was applied to both cases.

inative information is produced. One of the advantages of CADAN is that the refinement of clusters will be further conducted if training is continued. At the final stage (Epoch = 320, Fig. 4.5d), the clusters are nearly ideal and the subspace is domain-invariant. The training process reveals that CADAN is trained sequentially in response to different training objectives. Within a fixed period, CADAN will either learn to perform domain adaptation or speaker discrimination, which exactly matches our original intention to design two separate feature extractors that response to different training objectives independently.



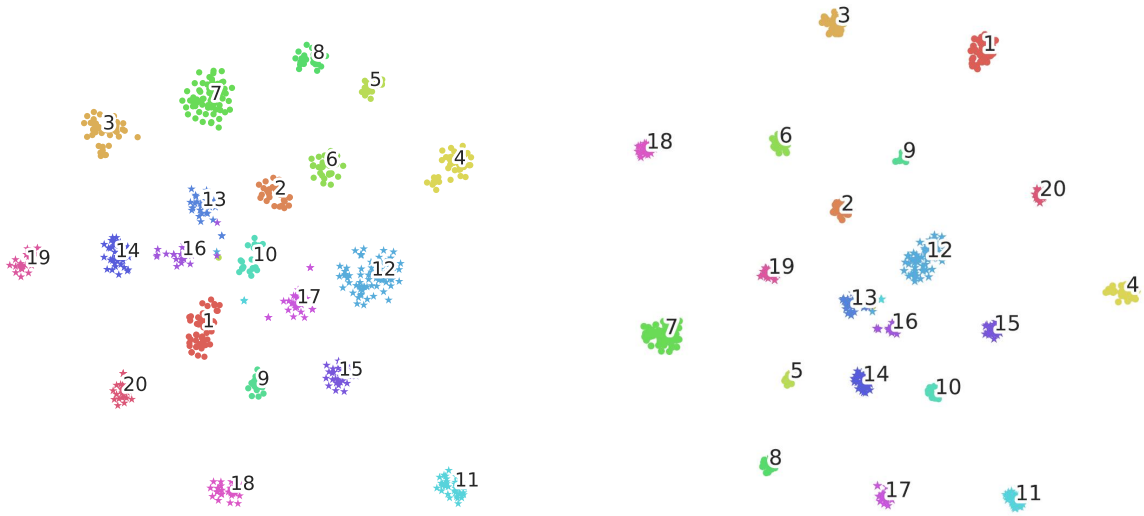
(a) Epoch = 0

(b) Epoch = 100

(c) Epoch = 150

(d) Epoch = 320

Figure 4.5: t-SNE plots at different training stages of CADAN. I-vectors were derived from the utterances of 10 male speakers (•) and 10 female speakers ($\star$). The numbers on top of each cluster are the speaker indexes (Speakers 1–10 are male and Speakers 11–20 are female) and each speaker is represented by one color.



(a) Without Fuzzifier (b) With Fuzzifier

Figure 4.6: t-SNE plots of transformed vectors ($\hat{\mathbf{z}}$) obtained by (a) CADAN with a fuzzifier in Fig. 4.1 and (b) a CADAN with the fuzzifier replaced by a speaker classifier. Refer to the caption of Fig. 4.5 for the meaning of markers and colors.

4.4.4 Fuzzifier vs. Speaker Classifier

Recall that the motivation of using a fuzzifier instead of a speaker classifier in CACAN is that the former is better at forcing the class encoder G_{cls} in Fig. 4.1 to produce more speaker discriminative latent vectors than the latter. To demonstrate that it is indeed the case, we conducted another experiment in which the fuzzifier in Fig. 4.1 was replaced by a speaker classifier C. The network is similar to DAN except for the splitting of the feature extractor into two branches. The objective functions in Eq. 4.1a and Eq. 4.1b are modified as follows:

$$\text{Train } C : \min_{C} \left\{ -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[\sum_{k=1}^{K} y_{\text{cls}}^{(k)} \log C(G(\mathbf{x}))_k \right] \right\} \tag{4.3a}$$

$$\text{Train } G_{\text{cls}} : \min_{G_{\text{cls}}} \left\{ -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[\sum_{k=1}^{K} y_{\text{cls}}^{(k)} \log C(G(\mathbf{x}))_k \right] \right\}. \tag{4.3b}$$

The training of G_{dom} and D in Eq. 4.2a and Eq. 4.2b remains unchanged.

Fig. 4.6 compares the transformed vectors obtained by the proposed CACAN and a CACAN whose fuzzifier is replaced by a speaker classifier. The result clearly shows that the fuzzifier can make the transformed vectors more speaker discriminative. A possible explanation is that minimizing the cross-entropy in Eq. 4.3a will make the lower layers of the speaker classifier to contain speaker information. This information will be wasted because we will only use the feature extractor to produce the transformed vectors after training. On the other hand, minimizing the cross-entropy in Eq. 4.1a encourages confusable input but Eq. 4.1b encourages speaker discriminative transformed vectors. As a result, the fuzzifier ensures that speaker information will be kept in the latent representation $\hat{\mathbf{z}}$.

In other words, the fuzzifier will force the class encoder to take all the responsibility for producing discriminative features so that all of the useable discriminative information is encapsulated inside the class encoder by which a more discriminative subspace is achieved.

Algorithm 1 Training of CADAN (Fig. 4.1). In Line 9, $\mathbf{W}_1$ ($\mathbf{W}_2$) contains the weights connecting G_{cls} (G_{dom}) and the output nodes of the feature extractor. In Lines 11 and 12, M is the number of domains. In Line 13, R is the number of inner iterations for training G_{cls} within an epoch.

1: **procedure** CADAN_TRAIN($\mathcal{X}, \mathcal{Y}_{\text{cls}}, \mathcal{Y}_{\text{dom}}$)
2: *Input*: Training i-vectors $\mathcal{X}$ and their class labels $\mathcal{Y}_{\text{cls}}$ and domain labels $\mathcal{Y}_{\text{dom}}$
3: *Output*: G_{cls}, G_{dom}, D and F
4: Initialize the weights of G_{cls}, G_{dom}, D and F using the Xavier initializer
5: **foreach** epoch **do**
6: Create N mini-batches $\{\mathcal{X}_i, \mathcal{Y}_{\text{cls},i}$ and $\mathcal{Y}_{\text{dom},i}\}_{i=1}^{N}$ of size B from $\{\mathcal{X}, \mathcal{Y}_{\text{cls}}, \mathcal{Y}_{\text{dom}}\}$
7: **for** $i = 1$ to N **do**
8: **for** $j = 1$ to B **do**
9: Compute $\hat{\mathbf{z}}_{ij} = [\mathbf{W}_1\ \mathbf{W}_2]^{\mathsf{T}}[G_{\text{cls}}(\mathbf{x}_{ij})\ G_{\text{dom}}(\mathbf{x}_{ij})]$, where $\mathbf{x}_{ij} \in \mathcal{X}_i$
10: **end for**
11: Train domain discriminator D using $\{\hat{\mathbf{z}}_{ij}\}_{j=1}^{B}$ as input and $\{\mathbf{y}_{\text{dom},ij}\}_{j=1}^{B}$ as target outputs of domain discriminator D, where $\mathbf{y}_{\text{dom},ij} = [y_{\text{dom},ij}^{(1)} \dots y_{\text{dom},ij}^{(M)}]^{\mathsf{T}} \in \mathcal{Y}_{\text{dom},i}$ (Eq. 4.2a)
12: Train adversarially domain suppressor G_{dom} using $\{\mathbf{x}_{ij}\}_{j=1}^{B}$ as input, and $[\frac{1}{M} \dots \frac{1}{M}]^{\mathsf{T}}$ as the target output of domain discriminator D, where M is the number of domains (Eq. 4.2b)
13: **for** $r = 1$ to R **do**
14: Train class encoder G_{cls} using $\{\mathbf{x}_{ij}\}_{j=1}^{B}$ as input, and $\{\mathbf{y}_{\text{cls},ij}\}_{j=1}^{B}$ as output of class encoder G_{cls}, where $\mathbf{y}_{\text{cls},ij} = [y_{\text{cls},ij}^{(1)} \dots y_{\text{cls},ij}^{(K)}]^{\mathsf{T}} \in \mathcal{Y}_{\text{cls},i}$ and K is the number of classes (Eq. 4.1b)
15: **end for**
16: Train adversarially fuzzifier F using $\{\hat{\mathbf{z}}_{ij}\}_{j=1}^{B}$ as input, and $\mathbf{f}_{\text{cls},ij} = [\frac{1}{K} \dots \frac{1}{K}]^{\mathsf{T}}$ as the target output of fuzzifier F (Eq. 4.1a)
17: **end for**
18: **end foreach**
19: **end procedure**

Chapter 5

Conclusions

This thesis firstly demonstrates the capability of a gender-independent speaker verification system based on iterative spectral clustering, i-vectors, inter dataset variability compensation (IDVC), mixture of PLDA and gender-aware DNNs. Evaluations on NIST 2016 SRE show that despite the limited amount of development data and the unavailability of speaker and gender labels in the development data, the proposed system can achieve superior performance. Results also reveal that iterative spectral clustering outperforms traditional clustering methods such as agglomerative hierarchical clustering because the PLDA scoring intrinsically requires i-vector pairs, which can be easily incorporated into the similarity matrix of spectral clustering. A number of factors contribute to this superior performance. Firstly, the gender-dependent IDVC helps to reduce the gender and language mismatch in development data of NIST 2016 SRE. Secondly, iterative spectral clustering can effectively find the hypothesized speaker labels for training the PLDA mixture model. Thirdly, the gender-aware DNN provides not only the gender posteriors for the PLDA mixture model but also accurate gender labels for the gender-dependent IDVC to reduce the gender and language mismatches in the i-vectors.

For the other work, the capability of contrastive adversarial domain adaptation network has also been proved through splitting the feature extractor into two contrastive branches, with one branch delegating for the class-dependence in the latent space and another branch

focusing on domain-invariance. Results demonstrate that the embedded features produced by CADAN significant improvement in speaker identification accuracy when compared with the conventional DAN under clean and noisy conditions, respectively.

One key contribution of the works lie in the novel integration of IDVC, mixture of PLDA and the iterative training process of these components in the speaker verification system and another novelty comes from the splitting structure of neural networks for adversarial learning.

Further investigations are necessary to improve the performance of the proposed methods. These investigation will look at better way to adapt the PLDA mixture models.

Chapter 6

Future Work

In this study, domain adaptation is separated into two parts. The first part uses gender-aware DNN to train a gender-independent PLDA mixture model to handle both genders. The classification accuracy of the DNN is 89.67% on the SRE16 development dataset. However, the gender identification approach proposed by Ranjan *et al* [19] is able to achieve 97.62% accuracy on the Fisher-English corpus. They utilized the unsupervised label generating-max margin clustering (LG-MMC) to maximize the margin of the gender difference. This method utilizes the i-vectors and PLDA to construct a gender classifier.

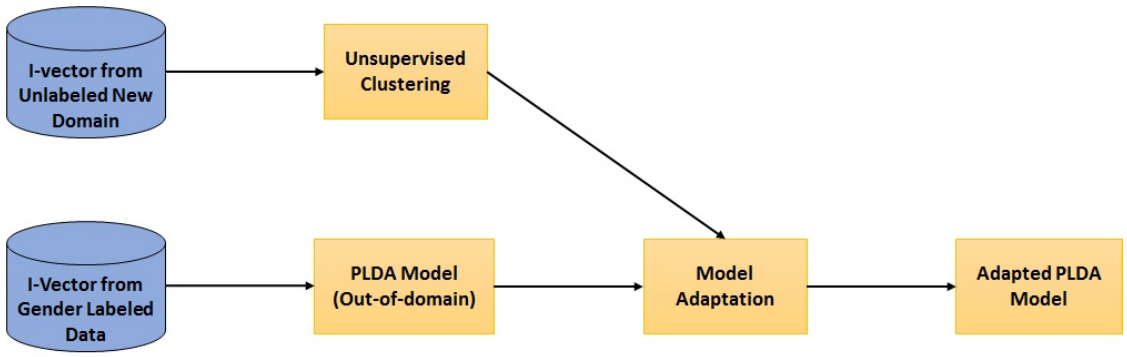


Figure 6.1: Adapting the out-of-domain PLDA model to a new domain

Out-of-domain data with gender labels are used to train an out-of-domain PLDA model. The in-domain data without gender labels are given hypothesized gender labels by LG-MMC to train an in-domain PLDA model. After training, the two sets of across-class

covariance matrix and within-class covariance matrix $(\mathbf{\Gamma}_{in}, \mathbf{\Sigma}_{in})$ and $(\mathbf{\Gamma}_{out}, \mathbf{\Sigma}_{out})$, are obtained. $\mathbf{\Gamma}$ represents the across-gender covariance and $\mathbf{\Sigma}$ represents the within-gender covariance. The adapted covariance matrices are computed as follows:

$$\mathbf{\Gamma}_{adapt} = \alpha_1 \, \mathbf{\Gamma}_{in} + (1 - \alpha_1)\mathbf{\Gamma}_{out}$$

$$\mathbf{\Sigma}_{adapt} = \alpha_2 \mathbf{\Sigma}_{in} + (1 - \alpha_2)\mathbf{\Sigma}_{out},$$

where α_1 and α_2 are control parameters and $\alpha_1, \alpha_2 \in [0, 1]$. This method provides a reliable gender classification system when the in-domain data do not have gender labels but out-of-domain data have sufficient gender labels, which can perfectly match the situation in SRE16.

The GD-IDVC can be improved. The results of this thesis suggest that GD-IDVC alone cannot achieve superior performance. The reason might be that the 4 sub-groups are based on genders, major and minor languages. However, within the major languages, there are two disparate languages. Similar situation also occurs in the minor languages. This may lead to inaccurate separation of languages. Thus, it is necessary to construct a reliable language classifier to further suppress the language mismatches.

A key element of our experiments is that we used a gender-independent model as the backbone to design the mixture of PLDA model. In future work, we can use a gender-dependent framework to design the experiments and determine the speaker verification accuracy for male and female, separately. The motivation of performing speaker verification in a gender-dependent manner is that with today's technology, gender classification can be very accurate (up to 99%).

To further enhance the CADAN framework, we firstly can modify the model to an end-

to-end DNN model. The DNN will be responsible for two purposes: (1) feature extraction and (2) domain transformation. Due to time constraint, the main goal of the CADAN framework in this work is to reduce gender mismatch. In future work, we can extend the model to tackle a more general domain mismatch scenario in which channel and language mismatches also appear in the data. Beside, we may compare the CADAN framework with the more recent domain adaptation methods.

BIBLIOGRAPHY

[1] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[2] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5796–5800.

[3] F. Fang, X. Wang, J. Yamagishi, I. Echizen, M. Todisco, N. Evans, and J. Bonastre, "Speaker anonymization using x-vector and neural waveform models," *arXiv preprint arXiv:1905.13561*, 2019.

[4] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[5] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[6] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1695–1699.

[7] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer, "Study of senone-based deep neural network approaches for spoken language recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, no. 1, pp. 105–116, 2016.

[8] P. Kenny, "Bayesian speaker verification with heavy-tailed priors.," in *Proc. Odyssey*, 2010, p. 14.

[9] K. J. Han and S. S. Narayanan, "Agglomerative hierarchical speaker clustering using incremental Gaussian mixture cluster modeling," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[10] J. B. Shi and Malik. J., "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[11] W. Y. Chen, Y. Q. Song, H. J. Bai, C. J. Lin, and E. Chang, "Parallel spectral clustering in distributed systems," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 568–586, 2011.

[12] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nystrom method," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, 2004.

[13] W.B Liu, Z.D. Yu, and M. Li, "An iterative framework for unsupervised learning in the PLDA based speaker verification," in *The 9th International Symposium on Chinese Spoken Language Processing*, 2014, pp. 78–82.

[14] N. Li and M. W. Mak, "SNR-invariant PLDA with multiple speaker subspaces," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5565–5569.

[15] M. W. Mak, X. M. Pang, and J. T. Chien, "Mixture of PLDA for noise robust i-vector speaker verification," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 1, pp. 130–142, 2016.

[16] N. Li, M. W. Mak, and J. T. Chien, "DNN-driven mixture of PLDA for robust speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1371–1383, 2017.

[17] S. Riley and A. Evans, "Gender," in *The Palgrave Handbook of Critical Social Psychology*, pp. 409–431. Springer, 2017.

[18] R. R. Patel, K. Forrest, and D. Hedges, "Relationship between acoustic voice onset and offset and selected instances of oscillatory onset and offset in young healthy men and women," *Journal of Voice*, vol. 31, no. 3, pp. 389–399, 2017.

[19] S. Ranjan, G. Liu, and J. H. Hansen, "An i-vector PLDA based gender identification approach for severely distorted and multilingual DARPA RATS data," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 331–337.

[20] D. Garcia-Romero and A. McCree, "Supervised domain adaptation for i-vector based speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4047–4051.

[21] J. Villalba and E. Lleida, "Bayesian adaptation of PLDA based speaker recognition to domains with scarce development data," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2012, pp. 641–647.

[22] Suwon Shon, Seongkyu Mun, Wooil Kim, and Hanseok Ko, "Autoencoder based domain adaptation for speaker recognition under insufficient channel information," *arXiv preprint arXiv:1708.01227*, 2017.

[23] S. H. Shum, D. A. Reynolds, D. Garcia-Romero, and A. McCree, "Unsupervised clustering approaches for domain adaptation in speaker recognition systems," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2014, pp. 265–272.

[24] L. X. Li and Man Wai Mak, "Unsupervised domain adaptation for gender-aware PLDA mixture models," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5269–5273.

[25] H. Aronowitz, "Compensating inter-dataset variability in PLDA hyper-parameters for robust speaker recognition," in *Proc. Speaker Odyssey: Speaker and Language Recognition Workshop*, 2014, pp. 282–286.

[26] H. Aronowitz, "Inter dataset variability compensation for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4002–4006.

[27] H. Aronowitz, "Inter dataset variability modeling for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 5400–5404.

[28] M. H. Rahman, A. Kanagasundaram, D. Dean, and S. Sridharan, "Dataset-invariant covariance normalization for out-domain PLDA speaker verification," in *Proc. Interspeech*, 2015, pp. 1017–1021.

[29] O. Glembek, J. Ma, P. Matejka, B. Zhang, O. Plchot, L. Burget, and S. Matsoukas, "Domain adaptation via within-class covariance correction in i-vector based speaker recognition systems," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4032–4036.

[30] W. W. Lin, M. W. Mak, L. X. Li, and J. T. Chien, "Reducing domain mismatch by maximum mean discrepancy based autoencoders," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2018, pp. 162–167.

[31] W.W. Lin, M. W. Mak, and J. T. Chien, "Multisource i-vectors domain adaptation using maximum mean discrepancy based autoencoders," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 26, no. 12, pp. 2412–2422, 2018.

[32] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *arXiv preprint arXiv:1409.7495*, 2014.

[33] Q. Wang, W. Rao, S. Sun, L. Xie, E.S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *Proc. International conference on Acoustic, Speech, and Signal Processing*, 2018, pp. 819–824.

[34] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[35] A. O. Hatch, S. S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition.," in *Interspeech*, 2006.

[36] H. Aronowitz, "Inter dataset variability compensation for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4002–4006.

[37] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems.," in *Proc. Interspeech*, 2011, pp. 249–252.

[38] D. Garcia-Romero, A. McCree, S. Shum, N. Brummer, and C. Vaquero, "Unsupervised domain adaptation for i-vector speaker recognition," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2014, pp. 409–431.

[39] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[40] I. S. Dhillon, Y. Q. Guan, and B. Kulis, "Kernel k-means: Spectral clustering and normalized cuts," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 551–556.

[41] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[42] G. E. Hinton, "Learning multiple layers of representation," *Trends in cognitive sciences*, vol. 11, no. 10, pp. 428–434, 2007.

[43] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," Tech. Rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[44] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[45] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[46] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[47] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[48] N. Li, M. W. Mak, and J. T. Chien, "Deep neural network driven mixture of PLDA for robust i-vector speaker verification," in *Proc. Spoken Language Technology Workshop*. IEEE, 2016, pp. 186–191.

[49] K. Wu and D. G. Childers, "Gender recognition from speech. Part I: Coarse analysis," *The journal of the Acoustical society of America*, vol. 90, no. 4, pp. 1828–1840, 1991.

[50] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, Crete, Greece, Jun. 2001, pp. 213–218.

[51] M. W. Mak and H. B. Yu, "A study of voice activity detection techniques for NIST speaker recognition evaluations," *Computer Speech & Language*, vol. 28, no. 1, pp. 295–313, 2014.

[52] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 19–41, Jan. 2000.

[53] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. Ninth International Conference on Spoken Language Processing*, 2006, p. 1471–1474.