# ADVANCEMENTS IN DOMAIN ADAPTATION FOR SPEAKER RECOGNITION AND EFFECTIVE SPEAKER DE-IDENTIFICATION

by

Fahimeh Bahmaninezhad


APPROVED BY SUPERVISORY COMMITTEE:

John H. L. Hansen, Chair

Carlos Busso

P. K. Rajasekaran

Chin-Tuan Tan

*To my family.*

# ADVANCEMENTS IN DOMAIN ADAPTATION FOR SPEAKER RECOGNITION AND EFFECTIVE SPEAKER DE-IDENTIFICATION

by

FAHIMEH BAHMANINEZHAD, BS, MS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

May 2020

# ACKNOWLEDGMENTS

First, I would like to express my appreciation and thanks to my PhD advisor, Dr. John H.L. Hansen, who gave me endless support and guided me through my research path and made me to be a far better researcher by sharing his valuable knowledge and expertise. I am extremely fortunate to have an advisor who gave me the freedom to explore many research directions. His hardworking and passionate attitude has inspired me in many ways which not only made me a better researcher in speech and language processing, but also helped me learn other aspects of being a successful researcher.

I am also sincerely thankful to my committee members, Dr. Carlos Busso, Dr. P.K. Rajasekaran, and Dr. Chin-Tuan Tan, for their highly valuable feedback and time that have helped improve this dissertation.

I would like to thank Chunlei Zhang and Shivesh Ranjan, my fellow researchers at the Center for Robust Speech Systems (CRSS), for valuable discussions; I had a great time working with them on the various projects at CRSS. I am very thankful to my mentors at Microsoft and Tencent during my internships as well, Kazuhito Koishida and Shi-Xiong Zhang. I am very fortunate to have had the wonderful opportunity to learn from them to enrich my PhD experience.

Last but not least, I am grateful to my parents, Jahangir and Zivar, for their unconditional love and understanding that gave me the strength to chase my dreams. I would like to express my gratitude to my sisters and brother for their continuous support and love. I am deeply thankful to my husband, Soheil, for his love, understanding and companionship during my PhD years.

December 2019

ADVANCEMENTS IN DOMAIN ADAPTATION FOR SPEAKER RECOGNITION AND EFFECTIVE SPEAKER DE-IDENTIFICATION

Fahimeh Bahmaninezhad, PhD
The University of Texas at Dallas, 2020

Supervising Professor: John H. L. Hansen, Chair

Recent advancements in machine learning and artificial intelligence have significantly impacted the way humans interact with machines. Voice assistant based solutions are examples of emerging technology advancements that impact human-machine interaction. Since, speech is the most natural form of human communication, voice assistant devices have received wide user acceptance, and have become a pleasant way to facilitate and address everyday living needs, including access to the current news, events, etc. These voice-based technologies have been made possible through advanced robust processing of speech signals. Depending on the application, various speech processing techniques are required to achieve an effective overall robust solution. Speech recognition is required when text content of spoken words is needed; for example adding text captions to broadcast news or YouTube videos. If a service should become available based on who is interacting with the device, speaker recognition becomes a required step; for example, if an individual gains access to a data account (e.g., music, voice-mail, health or financial records), effective speaker recognition is needed for that service. Overall, a range of solutions in speech processing can be required to address an overall request. Other areas of speech processing that benefit the human-machine interaction include language/dialect recognition, speech enhancement, machine translation, speech synthesis, voice conversion, general diarization, etc.

The environment where a person interacts with a device and input tools employed (such as phone or microphone) can impact performance. It is common to have intrinsic/extrinsic mismatch between train data and application data; in other words, data used for training the speech processing tasks is often different than those at the test time. These variations need to be considered while developing effective speech systems, especially when performance is impacted significantly due to mismatch conditions. In this dissertation, we study the problem of speaker recognition for domain mismatch. Recognizing the identity of a speaker is an important task in speaker-dependent applications, and providing robust performance regardless of how data is captured for model training and considering environmental/extrinsic changes within the application phase is very important.

In this dissertation, we propose two categories of solutions to address the mismatch problem in speaker recognition: discriminant analysis based adaptation methods (generalized discriminant analysis-GDA, and support vector discriminant analysis-SVDA) and deep learning based adaptation technique (a-vector speaker embeddings). The proposed solutions are evaluated on NIST SRE-10, NIST SRE-16 and NIST SRE-18 tasks. The GDA and SVDA achieved 20% and 32% improvement in terms of EER for SRE-10 task. A-Vectors with incorporating SVDA achieved up to 18% improvement over the previous best performing solution on SRE-16 task. In addition, we propose a solution for speaker de-identification task.

In more detail, the first category of solutions we propose is based on domain mismatch compensation with discriminant analysis methods. Traditional speaker recognition use linear discriminant analysis to reduce the dimensionality of speaker embeddings and provide a better discriminant feature representations for speaker classes. We propose non-linear discriminant analysis to compensate for variabilities included during recording through generalized discriminant analysis. In addition, domain adaptation is also incorporated through

our proposed support vector discriminant analysis method; which also provides improved discrimination by considering the boundary structure of speaker classes.

The second category of solutions are based on domain mismatch compensation with deep learning approaches. We propose a deep learning based technique to compensate for unwanted directions and information included in speaker embeddings, and provide domain-invariant speaker representations.

Finally, we address speaker de- identification advancements to help protect confidential speaker or text-content within a given audio stream. Taken collectively, these three domains highlight technological advancement, which strengthen and make speaker recognition more useful in commercial, personal, and governmental/society applications, which incorporate human-speech engagement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| BLSTM | Bidirectional Long Short Term Memory |
| BN | Batch Normalization |
| CNN | Convolution Neural Network |
| DNN | Deep Neural Network |
| EER | Equal Error Rate |
| EM | Expectation Maximization |
| GDA | Generalized Discriminant Analysis |
| GMM | Gaussian Mixture Model |
| HMM | Hidden Markov Model |
| JFA | Joint Factor Analysis |
| LDA | Linear Discriminant Analysis |
| MAP | Maximum A Posterior |
| ML | Maximum Likelihood |
| MOS | Mean Opinion Score |
| PLDA | Probabilistic Linear Discriminant Analysis |
| SAD | Speech Activity Detection |
| SRE | Speaker Recognition Evaluation |
| SVDA | Support Vector Discriminant Analysis |
| SVM | Support Vector Machine |

| | |
|---|---|
| TV | Total Variability |
| UBM | Universal Background Model |

# CHAPTER 1

# INTRODUCTION

Recent advancements in machine learning and artificial intelligence have significantly impacted the way humans interact with machines. The growth of voice assistant devices through these advancements shows the desire for integrating machine learning into our everyday needs. As speech is the most natural way of communication, voice assistant devices have received a great acceptance, and they have became a pleasant way to interact with these devices. The voice-based human-machine interactions have been made possible through robust processing of speech signals. Depending on the application, various speech processing techniques can be integrated into the device/service. Speaker recognition (recognizing the identity of a speaker from his/her voice, or just verifying the identity of a speaker) provides personalized communication, and secure access to speaker-specific information. Therefore, accurate speaker recognition systems can play a significant role in establishing a secure connection between human and machines. In this dissertation, we focus on the speaker recognition task specifically for mismatch conditions; when data used for development of the system and data used for testing the system are collected under different conditions.

Generally speaking, speaker recognition refers to the task of recognizing whether a target/desired speaker is talking during a given test segment or not [3, 4], which is the main task we study here. Approaches proposed for speaker recognition have evolved significantly over the past few years to overcome the limitations and variations of the training data as well as to provide consistent performance on naturalistic audio streams [4]; however, challenges still remain, especially when there is a mismatch between data used to develop the system and data used at the application phase. Speaker recognition provides very good performance when there is a minor noise or distortions in testing utterances (or generally when there is a minor mismatch between training and test data); however, with adding more distortions and variabilities, the task becomes more difficult. Mismatch conditions can be divided into

two categories, extrinsic (channel, noise, etc) and intrinsic (duration, language, and speaker traits including stress, emotion, Lombard effect, vocal effort, accent). Both have been widely studied for developing robust speaker recognition systems. With current advancements in speaker recognition, intrinsic mismatch introduces more difficulty and challenge; which we specifically focus on this type of mismatch during this dissertation. Domain mismatch problem as an intrinsic mismatch, i.e., when train (system development data) and enrollment/test data (application data) are collected from different sources, is one of the problems which is addressed further here. We investigate speaker recognition for the domain mismatch problem (intrinsic mismatch); specially for those challenges introduced by NIST (National Institute of Standards and Technology) SRE (speaker recognition evaluation) in 2010, 2016 and 2018.

In addition, we study the performance of speaker recognition systems when speakers are de-identified by a voice conversion based system. With growing the amount of data become available everyday through the internet, speaker de-identification can provide secure access to this data; therefore, from one aspect de-identifying the identity of a speaker introduces an approach to use the huge amount of available data securely. Another application is in related with medical data; processing medical data is very difficult as we cannot access private information of patients, i.e., information of the patient should not be revealed; both the voice characteristics and the content. One strategy for processing medical records while protecting the privacy of the patients is de-identifying the utterance and voice. Removing personal information in linguistic features and para-linguistic features is a direction towards making those data available for accurate processing through automatic systems.

## 1.1 Dissertation Contributions

The key contributions of this dissertation are summarized as follows:

- **Generalized discriminant analysis (GDA):** Dimension reduction with linear discriminant analysis (LDA) is usually employed after normalizing the speaker embeddings

to further discriminate the speaker classes. Here, we propose to extend the linear discriminator to a nonlinear one with utilizing kernel functions. The GDA simply provides discriminatory directions non-linearly, and it is shown to compensate for the distortion and mismatch problem in developing the speaker recognition systems [5].

- **Support-vector discriminant analysis (SVDA):** Here, we further improve the discriminant analysis and dimension reduction of speaker embeddings with SVDA. The discrimination is provided with considering boundary structure of the speaker classes rather than the mean centroid of them. On the other hand, domain mismatch is the most common and challenging problem for speaker recognition systems. SVDA can successfully be trained to address the domain adaptation in supervised as well as unsupervised approaches. Our experiments show that SVDA successfully performs adaptation when language of training data is different from enrollment and test data [2, 6, 7].

- **Domain adaptive auxiliary features for i-Vector (a-Vector):** Appending auxiliary information features to the input features in order to compensate for the mismatch and normalize language related information is studied here. We trained a model to learn domain adaptive features and concatenated them with the i-Vectors; we call the new embeddings as a-Vectors. Experiments show that a-Vectors along with the SVDA technique provide strong performance on NIST SRE-16 data [8].

- **Voice conversion based speaker de-identification:** With growing artificial intelligence based assistant devices and availability of more speech data, one of the important problems is related to protecting the privacy of speakers. We provide a solution for that based on voice conversion systems, and evaluate how speaker recognition systems can be successful to recognize the de-identified speakers. From one point, experiments show that simply applying voice conversion techniques provides promissing

performance evaluated by speaker recognition systems. In addition, voice conversion techniques can help to generate new identities for each individual given speaker [9], which serves as an augmentation technique to generate more speakers and data.

## 1.2 Dissertation Organization

The dissertation is organized as follows:

- **Chapter 2:** This chapter presents background on the speaker recognition task, including problems and issues related to this task. We introduce the configuration of our developed modules in a traditional speaker recognition system (used in the experiments as our baseline systems). The general description of data and metrics used to evaluate the performance of our solutions are also covered in this chapter.

- **Chapter 3:** Here, we introduce our two novel methods for domain adaptation in speaker recognition. The methods are developed based on machine learning techniques and address the problems of linear discriminant analysis in concern with the domain mismatch. Our two methods named GDA and SVDA are shown to be effective for NIST SRE-10 and SRE-16 tasks.

- **Chapter 4:** This chapter describes our novel deep learning based solutions for domain adaptation in speaker recognition. Our proposed method specifically targets the unsupervised domain adaptation, where only very limited unlabeled in-domain data is available.

- **Chapter 5:** In this chapter, we integrate multiple techniques for domain adaptation together. Throughout experiments are designed to emphasize the effect of domain adaptation and integration of multiple techniques for NIST SRE-16 and NIST SRE-18 tasks.

- **Chapter 6:** In this chapter, we study the performance of speaker recognition systems towards speaker de-identification. With using voice conversion techniques we modify speakers voice, and the results show that it achieves high de-identification accuracy by evaluating with the state-of-the-art speaker recognition solution.

- **Chapter 7:** In this chapter, we summarize our proposed dissertation contributions for domain adaptation in speaker recognition; and our additional study on speaker de-identification. This chapter also highlights areas for future work.

# CHAPTER 2

# SPEAKER RECOGNITION: BACKGROUND

Text-independent speaker recognition is defined as recognizing whether a specific target or desired speaker is talking during a given speech segment or not [3, 4].

To address the problems and challenges in speaker recognition, the proposed solutions over the past few years have migrated from GMM (Gaussian mixture model)-UBM (universal background model) [10] based systems towards i-Vector and deep learning based solutions (i.e., t-Vector [11] and x-Vector [12]), where a speaker embedding is extracted for each utterance of the speaker. The speaker embeddings are next fed into a classification or scoring module to perform recognition/verification task. During this migration, we can also include methods based on: joint factor analysis (JFA) [13], i-Vector [14] solutions with cosine distance scoring or support vector machine (SVM) classification [14], and i-Vector with PLDA (probabilistic linear discriminant analysis) scoring [14, 15] as well. These also include both UBM/i-Vector and DNN/i-Vector, where i-Vector/PLDA is the state-of-the-art method for speaker recognition (depending on the data configuration) as well as other speech areas, such as language recognition. These advancements in speaker recognition had provided satisfactory performance on NIST (National Institute of Standards Technology) SRE (speaker recognition evaluation) tasks until 2012. However, for challenges introduced in the SRE-16 and SRE-18, current solutions are not sufficiently effective and require further investigation. More specifically, the focus of NIST SRE-16 and SRE-18 are on the domain mismatch problem (training data used for development had different language sets than those in enrollment/test data which are used at the application phase; there were handset and microphone mismatch options as well), where current solutions in speaker recognition do not provide effective performance for them. Here, we mainly focus on the speaker recognition under mismatch conditions; specifically, for those challenges introduced in SRE-16 and SRE-18.

Domain mismatch compensation for speaker recognition has been previously studied for diverse datasets and tasks (other than SRE-16 and SRE-18), including [5, 16, 17, 18, 19, 20]. For NIST SRE tasks specifically, multiple studies have proposed methods to compensate for the domain mismatch. Here, we review some earlier works on these tasks. Generally speaking, domain mismatch compensation techniques can be applied to speaker recognition systems at different phases: *front-end* level compensation; (e.g., MAP - maximum a posterior - adaptation of GMMs model [21], speaker embedding extraction), and *back-end* level (e.g., PLDA adaptation [12]). Figure 2.1 represents an overall block-diagram of an i-Vector based speaker recognition system specifying front-end and back-end level processing. From an alternative viewpoint, domain mismatch compensation methods can be categorized into *supervised* or *unsupervised* techniques as well. When in-domain data are unlabeled, pseudo labeling can be integrated into the system to provide for supervised adaptation.

To compensate for the domain mismatch at speaker-embedding extraction level (i.e., front-end), [21] introduced GMM-SVM with Nuisance Attribute Projection (NAP) trained using clustered unlabeled in-domain data for SRE-16 task. They also studied other methods for unsupervised domain mismatch compensation, using in-domain data for MAP adaptation of GMM models which both were shown to be effective. In addition, [22] proposed training a speaker classifier neural network for extraction of d-vectors. Interestingly, they did not attempt to assign pseudo speaker labels to the unlabeled data. [23] applied an unsupervised Bayesian adaptation method and achieved promising results. [24] replaced i-Vectors with two new proposed embeddings which are derived based on a DNN architecture. They evaluated the performance of the embeddings on both SRE-10 and SRE-16 tasks, although the idea is general and not necessarily developed for domain adaptation, experiments show that the discriminative training of speaker embeddings can be helpful towards the domain mismatch compensation rather than the traditional i-Vector embeddings. x-Vector [12] which uses data augmentation as well as PLDA adaptation is among the top performing systems for

SRE-16 and SRE-18 (where a small unlabeled in-domain data is provided for the adaptation purpose) tasks.

Overall, the performance of most of these methods have been reported along with other modifications at the back-end level. For the x-Vector, the back-end level techniques are shown to directly affecting the superiority of x-Vector over the i-Vector. Therefore, it would be difficult to drew a conclusion on how much the front-end level domain adaptation is successful, considering the fact that available in-domain data for SRE tasks is very limited.

To compensate for domain mismatch at the dimension reduction step, [22] used the LDA with within class covariance correction (WCC) technique, which updates the within class covariance matrix using in-domain data. Mismatch compensation at score calculation and score normalization steps are also studied in [21] where they added the replicate copies of in-domain data to the training set for modeling the classifiers. In addition, [21] used the in-domain data in multiple score normalization techniques. [25] not only applied whitening and mean centralization using in-domain data (both labeled and unlabeled), but also proposed multi-stage PLDA adaptation technique (which uses clustered unlabeled data). They also incorporated in-domain data into score normalization as well. [22] normalized the resulting scores using speaker dependent s-norm with a cohort created from training and unlabeled in-domain data. [25, 26] also mentioned they used unlabeled data for score calibration. These techniques were all proposed to compensate the domain mismatch at the back-end level and they are shown to be effective for NIST SRE tasks. Other studies on the SRE tasks targeting the domain mismatch compensation include [27, 28, 29, 30].

The block-diagram of a traditional i-Vector based speaker recognition system with PLDA scoring is shown in Figure 2.1 with focusing on the front-end and back-end level processing. In continue, we provide detailed explanation on the front-end processing and back-end processing of speaker recognition systems we developed in this dissertation and use as our baseline models throughout our experiments.

## 2.1 Front-End Processing

The front-end processing in speaker recognition systems refers to extracting speaker embeddings for each utterance, which captures the speaker-specific information. In traditional speaker recognition frameworks, concatenation of GMM (Gaussian mixture model) mean supervectors was used as speaker embeddings. However, they are very high dimensional and besides the speaker information they carry other unrelated information too. Therefore, over the past few years new methods are proposed for speaker embedding extraction. In this dissertation, we use three different speaker embeddings including i-Vector, x-Vector and t-Vector. Detailed description of these embeddings are presented in the following subsections. These embeddings have lower dimensions and are trained to exclusively contain information related to the voice characteristics of the speaker rather than other interfering information, such as channel and noise.

### 2.1.1 i-Vector

i-Vectors are successfully applied in speaker recognition [3, 7, 14, 26] and language recognition [31] to compactly represent speaker-dependent features while discarding channel and noise dependent directions. The block diagram of an i-Vector/PLDA speaker recognition is shown in Figure 2.1. It is worth mentioning that, the back-end processing is shared between other speaker embeddings in this study, which will be discussed in detail in section 2.2.

In i-Vector speaker embedding extraction, first mel-frequency cepstral features (MFCCs) are extracted for utterances in the dataset. Next, non-speech segments are removed with a speech activity detection module (SAD). Universal background model (UBM) and total-variability matrix (TV-matrix) are trained using the training data. The TV-matrix is used to map the high-dimensional GMM supervectors (GMMs are speaker adaptation models of the UBM) which include both speaker and channel related directions into a lower-dimension representation with focus only on speaker discriminatory features.

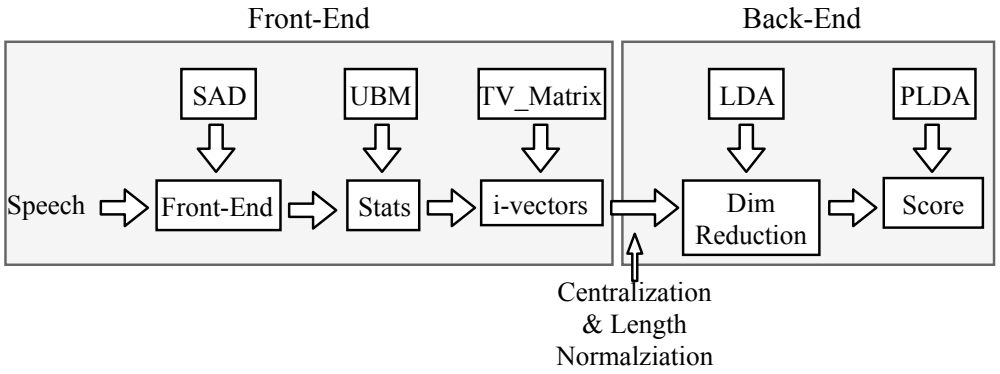


Figure 2.1: Block-diagram of i-Vector/PLDA speaker recognition.

In more detail, in the i-Vector framework, a channel and speaker dependent GMM supervector is factorized as,

$$M = m + Tw, \tag{2.1}$$

where $m$ is the UBM speaker and channel dependent supervector, and $T$ is the low rank total variability matrix (TV-matrix) which maps the high-dimensional GMM supervector into $w$, known as i-Vector. Alternatively, i-Vectors can be extracted from the output of DNN rather than the UBM. For the DNN/i-Vector framework [32, 33], $w$ is extracted by mapping the senones (frame posteriors for tied triphone states) of the DNN network using the total variability matrix.

### 2.1.2 x-Vector

The x-Vector is another speaker embedding extraction method that has been reported to achieve very effective speaker recognition performance in recent studies [12, 24]. First, for each utterance in the dataset, filterbank features are extracted and non-speech segments are then removed with a SAD module. Next, the extracted features are passed to a speaker discriminative model. The model is a deep neural network (DNN) based framework benefiting from practical techniques such as data augmentation and statistical pooling. The

embeddings are extracted over the entire utterance instead of at the frame-level. The last layer of the network is softmax activation and the network is trained by multi-class cross entropy objective function using corresponding speaker labels given by:

$$\mathcal{L}_s = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T f(\mathbf{x}_i)+b_{y_i}}}{\sum_{j=1}^{C} e^{W_j^T f(\mathbf{x}_i)+b_j}}, \tag{2.2}$$

where $N$ is the batch size, $C$ is the total speaker number in the training set, $f(\mathbf{x}_i)$ is the output of the embedding layer of the network ( i.e., speaker embedding). Here, $y_i$ is the corresponding class label, and $W$ and $b$ are the weights and bias for the last softmax layer of the network which acts as a classifier. Therefore, in contrast to i-Vector framework where embeddings are extracted from a generative model, x-Vectors are extracted from a discriminative framework.

### 2.1.3 t-Vector

The other speaker embedding extraction framework which we developed in this dissertation is t-Vector. t-Vectors are extracted from a discriminative model using triplet loss objective function. First, filter bank features are extracted for each utterance in the training set; then non-speech segments are discarded using SAD. The features are fed into inception-resnet-v1 network [1] to extract speaker embeddings. The parameters of the model are trained with jointly optimizing the triplet loss function and softmax loss. Triplet loss is popular objective function for training face or speaker verification systems [34, 35].

Inspired by the success of the softmax loss used in x-Vector models, the training of the t-Vector model is performed with a multi-task learning framework; and it is formulated by adding a $L_2$ normalized softmax loss ($\mathcal{L}_{s_{L_2}}$), which is an upgrade of original softmax loss:

$$\mathcal{L}_{s_{L_2}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T f(\mathbf{x}_i)+b_{y_i}}}{\sum_{j=1}^{C} e^{W_j^T f(\mathbf{x}_i)+b_j}}, \tag{2.3}$$
$$\text{subject to } \|f(\mathbf{x}_i)\|_2 = \alpha, \quad \forall i = 1, 2, ..., N$$

where a simple $L_2$ normalization is applied to the embedding layer before softmax layer, $\alpha$ is a constant that constrains the radius of the speaker embedding hypersphere. Finally, $\alpha$ is set to 24 empirically. The total loss function is an integration of three components: a triplet loss term $\mathcal{L}_{triplet}$, a $L_2$-norm softmax loss term $\mathcal{L}_{s_{L_2}}$, and a regularization term $\mathcal{L}_2$ which alleviates the over-fitting issue during training,

$$\mathcal{L}_{total} = \mathcal{L}_{triplet} + \omega_1 \mathcal{L}_{s_{L_2}} + \omega_2 \mathcal{L}_2. \tag{2.4}$$

Practically, the $\omega_1 = 0.1$ and $\omega_2 = 2e-5$ is found to result in a good overall combination.

The triplet sampling module plays an important role in the performance of t-Vector embedding extraction framework. Previously in [11], a subset of speakers in the training pool was selected for triplet formulation in each epoch. With the additional $\mathcal{L}_{s_{L_2}}$, it is better to see all the speakers in one epoch. In our experiments, always randomly selected segments from all training speakers are used for the triplet generation, and they are shuffled to ensure all classes can be seen within one epoch.

## 2.2 Back-End Processing

After extracting the speaker embeddings, usually mean-centralization, length normalization, discriminant analysis are applied to provide a robust speaker-dependent representation. Next, a scoring technique such as PLDA (probabilistic linear discriminant analysis) or simply cosine distance scoring performs the verification or recognition task. In order to fuse and combine multiple systems together, the back-end process continues with score calibration, score normalization and fusion. In this section, we describe the main components at the back-end level, which we integrated into our system development. All of these steps with no change are applicable to all the speaker embeddings introduced in the front-end section.

### 2.2.1 Normalization and Whitening

Immediately after extracting the speaker embeddings, it is shown that mean centralization, length normalization [36] and whitening are effective approaches to further discard speaker invariant information.

Mean centralization of the speaker embeddings provide an effective yet simple technique for removing information that are fixed over the whole training data and definitely not representing the speaker related information. On the other hand, if mean statistics are calculated from a target domain, it provides a simple domain adaptation technique as well. Mean statistics are calculated from training data (or adaptation data) and it is subtracted from all test and train data.

Length normalization simply ensures the speaker embeddings can be modeled with Gaussian distribution and are consistent with PLDA scoring assumptions.

In our systems, we include both mean centralization and length normalization.

### 2.2.2 Dimension Reduction

The block-diagram in Figure 2.1 shows that after extracting i-Vectors (as an example), mean-centralization and length-normalization, usually LDA (linear discriminant analysis) is applied to reduce the dimension size of the resulting i-Vectors (or generally speaker embeddings) as well as improve the discrimination of the speaker classes.

LDA is a supervised approach for learning a transformation matrix which maps the input into a lower dimensional space where the distance between speaker classes is larger and the classes are more centered around their mean. As a result, the covariance between samples in a class is smaller while the covariance between mean of different classes is larger.

### 2.2.3 PLDA Scoring

The task of speaker recognition is for given two embeddings to decide are they belong to one speaker or not. To perform this task, we can either calculate the cosine similarity between the two embeddings and comparing the score to the threshold value make a decision. Another approach is based on the PLDA (probabilistic linear discriminant analysis) scoring, which is effective for further suppressing the channel and noise distortions. In this study, we apply PLDA (as it is the state-of-the-art scoring method for speaker recognition) to calculate the likelihood scores representing either the inputs are from the same speaker or not.

PLDA models the speaker embedding $w_{ij}$ as,

$$w_{ij} = m + \Phi\beta_i + \epsilon_{ij}, \tag{2.5}$$

where $j$ is the j-th utterance for speaker $i$, $\Phi$ contains the basis for speaker subspace, $\beta_i$ corresponds to the coordinates in the speaker i-th subspace, and $\epsilon_{ij}$ represents a Gaussian distribution with zero mean and a covariance matrix (as one of the parameters of the PLDA model). The parameters of the model are estimated with the expectation maximization (EM) algorithm. At the test time, given two embeddings $\hat{w}_1$ and $\hat{w}_2$, we need to determine whether these two belong to one speaker (target) or not (non-target) with the following log-likelihood ratio,

$$\text{log-likelihood} = log\frac{p(\hat{w}_1, \hat{w}_2|\text{target})}{p(\hat{w}_1, \hat{w}_2|\text{non-target})}. \tag{2.6}$$

### 2.2.4 Score Normalization, Calibration and Fusion

To combine multiple complementary speaker recognition systems (which is an effective approach to boost the performance), it is essential to normalize the scores or calibrate them. In this dissertation, score calibration is performed using PAV calibration methods from the

BOSARIS toolkit [37]. Alternatively, score normalization such as s-norm can be applied to prepare the scores for the fusion phase. Score normalization, s-norm is applied with an adaptive cohort selection scheme followed by a top score selection [38].

We accomplish the score fusion by building a fused model. This step is performed based on the training of a logistic regression model. Let $x = \{x_1, x_2, ..., x_n\}$ be a feature vector by concatenating each single system scores. The target variable $y$ is a Bernoulli random variable and the probability of occurrence for that is dependent on the prediction given in Eq. 2.7. Regression coefficients $\omega$ are estimated using maximum-likelihood (ML) estimation. Scores from each single system are finally combined together with the estimated coefficients to obtain the fusion score $\hat{y}$.

$$p(y = 1|x, \omega) = \frac{1}{1 + \exp(-\omega^T x)} \tag{2.7}$$

$$\hat{y} = \omega^T x, \tag{2.8}$$

## 2.3 Corpora and Experimental Setup

The methods we proposed in this dissertation are examined on three different datasets released for NIST SRE evaluations. These data are widely used for development and evaluation of speaker recognition systems and they present the recent challenges for speaker recognition task. The general description of data is provided in the following sub-sections. In addition, we introduce the metrics we used to perform the evaluation and compare different systems.

### 2.3.1 NIST SRE-10

NIST SRE-10 telephone condition (condition 5) [39] is used in our experiments. Training data includes data collected from SRE2004, 2005, 2006, 2008, and Switchboard II Phase 2 and 3, and Switchboard Cellular Part 1 and 2 (both male and female speakers). Training data for

the back-end processing is restricted to only male speakers from NIST SRE2004, 2005, 2006, 2008 data. In addition, the trials used for experiments just contain male enrollment and test segments. The enrollment/test segment condition combinations that have been evaluated in this study includes core/core, core/10sec, and Coreext/3, 5, 10, 20, 40s, full [39]. Core and extended core (coreext) conditions have duration ranging between 3 to 5 minutes. To examine the effectiveness of the proposed method for short test segments, the coreext test data was truncated into 3-sec, 5-sec, 10- sec, 20-sec, and 40-sec segments. Extracting these short test data has been carried out after applying SAD; therefore, they do not contain non-speech frames. In addition, no modifications have been applied to the enrollment or training data. The number of speakers/segments for training UBM and TV-matrix are 5756/57273. And, number of speakers/segments for training the back-end modules are 1115/13605. For core/core, core/10sec, coreext/3,5,10,20,40s,full the number of target/non-target trials are 353/13707, 290/11700 and 3465/175873, respectively. In addition, number of enrollment segments for core and coreext are 2426 and 5237.

In any of the experiments, if the setup is different than the above mentioned one, it is explicitly mentioned.

### 2.3.2 NIST SRE-16

NIST SRE-16 fixed condition includes data from Call My Net corpus, previous Mixer/SRE data, both landline and cellular Switchboard and Fisher [3]. Here, we did not use Fisher data and Call My Net corpus for the training. Therefore, in our system, the total number of speakers and segments used for training UBM and TV-matrix are 5756 and 57273 respectively. At the back-end, we also did not use any of the Switchboard data, which leads to a total of 3794 speakers and 36410 segments for training LDA/PLDA.

Data assigned to the development and evaluation sets were collected from the Call My Net corpus. Data was collected outside of North America and consists of two subsets:

(1) *Major*: contains Tagalog and Cantonese languages, (2) *Minor*: contains Cebuano and Mandarin languages. Development data includes data from both minor and major language sets; evaluation data only contains data from the major set [3].

In addition, development data includes labeled and unlabeled sets. The labeled set is only from minor languages; 10 speakers talking Cebuano and 10 speakers talking Mandarin, with each possessing 10 segments. The unlabeled one has 2272 and 200 calls from major and minor languages, respectively (they do not have speaker id, language, gender, etc information) [3].

Overall, the total number of speakers/segments in enrollment set for development and evaluation are 80/120, and 802/1202, respectively. In addition, number of target/non-target trials for development and evaluation are 4828/19312 and 1986729/1949666, respectively. Throughout the dissertation, we refer to the development as DEV and evaluation as EVAL.

### 2.3.3 NIST SRE-18

The NIST SRE-18 as well targets a similar challenge with some modifications. For the fixed condition, the training data includes all previous SRE data, consisting of switchboard, Fisher, VoxCeleb, SITW (speaker in the wild); and the development set of SRE-16 were allowed to be used. The task includes two separate parts: CMN2 (Call My Net), and VAST (Video Annotation for Speech Technology), where for our study here we mainly focus on the CMN2 part. The CMN2 dataset used for the development and evaluation purposes contains data with the Tunisian Arabic language; while the training data is mostly in American English.

In contrast to SRE-16 where DEV and EVAL sets did not share the same languages, SRE-18 DEV and EVAL are considered to belong to one domain (i.e., the language for both are the same). The CMN2 part of the DEV set includes 25 speakers (with approximately 10 utterances per speaker). The SRE-18 DEV set also includes in-domain unlabeled data (no speaker ID, gender, or language labels) with 2332 utterances and speech duration ranging between 10s to 60s uniformly.

### 2.3.4 Performance Measurement

For SRE-10 evaluation metric includes equal error rate (EER) and minimum detection (minDCF) cost function [39] calculated by,

$$C_{Det} = C_{Miss} \times P_{Miss|Target} \times P_{Target} + C_{FalseAlarm} \times P_{FalseAlarm|NonTarget} \times (1 - P_{Target}). \quad (2.9)$$

Default values for parameters in this equation has been set as $C_{Miss} = C_{FalseAlarm} = 1$ and $P_{Target} = 1/1000$ from the NIST SRE-10 definition for minDCF.

For SRE-16, NIST provided a scoring software to the participating sites; it calculates the equal error rate (EER), minimum primary cost (min-Cprimary), and actual primary cost. In addition, the software reports both equalized (false alarm and false reject counts were equalized over various partitions) and unequalized scores. Details on these costs are provided in [3, 40].

For SRE-18 as well, NIST provided a scoring software to calculate the equal error rate (EER), minimum primary cost (min-Cprimary), and actual primary cost [41].

## 2.4 Summary

This chapter has provided a broad overview of the problem of speaker recognition with specific background advancements and references to prior work. This chapter has established the necessary foundation needed to advance speaker recognition to address intrinsic/extrinsic variability mismatch. Next chapter describes two of our proposed solutions for speaker recognition under mismatch conditions.

# CHAPTER 3

# DOMAIN ADAPTATION WITH DISCRIMINANT ANALYSIS



Mismatch between data at the development and application steps is very common, not only for speaker recognition but also in other speech research areas. Developing systems specifically for an application domain can be a solution to this problem. However, it is not reasonable to build a system from scratch for every possible mismatch scenario. There are challenges in general which make this approach less effective. From one aspect, available data for the target domain might be limited; therefore, the system may not be robust and overall performance is poor. On the other hand, when there is sufficient amount of data available from alternate domains, this can be used to help to capture other informative features for system development. In addition, in some scenarios, the development data may consist of only unlabeled data or poorly labeled data for the target domain. Accurate labeling of data is a difficult task; therefore, it is more beneficial to have a large pool of correctly labeled data at the development phase. With this content, at the application step it is possible to introduce a small amount of data from the target domain to adjust model parameters. Therefore, incorporating all available data within the adaptation or normalization methods can be a promissing approach to address variabilities present in the application phase.

In this chapter, we present our proposed methods to compensate for the mismatch conditions. Generalized discriminant analysis (GDA) places more emphasis on noise and channel mismatch conditions, which are non-linearly included in the speaker embedding representations. Therefore, GDA is a very general solution to suppress speaker-irrelevant information

from the embeddings. Next, the support vector discriminant analysis (SVDA) solution provides an adaptation technique specifically for language mismatch conditions. Therefore, SVDA provides an adaptation technique for mismatch conditions, which introduces a small amount of unlabeled data from the target domain which can effectively adapts the models.

## 3.1 Introduction

Here, we focus on the domain mismatch compensation at the back-end level; specifically when discriminant analysis is applied to reduce the dimension of speaker embeddings and to further discriminate the speaker classes.

Traditional LDA (linear discriminant analysis) finds the transformation matrix for a dimensionality reduction with the objective of minimizing the ratio of the within-class to between-class covariance matrices. LDA assumes different classes have a Gaussian distribution and share the same covariance matrix. Many variations of discriminant analysis have been proposed to partly relax the LDA assumptions. Kernel discriminant analysis or generalized discriminant analysis (GDA) [42, 43] finds a non-linear transformation, heterocedastic LDA (HLDA) [44] employs alternate covariance matrices for different classes, and mixture discriminant analysis (MDA) [45] assumes the distribution of each class is a mixture of Gaussians.

In speaker recognition systems, specifically those based on i-Vector representation, the effectiveness of various discriminant analysis methods have also been studied. [46] employed non-parametric or nearest neighbor discriminant analysis (NDA). The experimental results showed that NDA outperforms LDA especially when data are multimodal [47]. In addition, [48] used source-normalized LDA (SN-LDA), and [49] employed weighted LDA (WLDA) and weighted SN-LDA which were shown to be more effective in special mismatch conditions.

Here, we first propose generalized discriminant analysis (GDA) for speaker recognition [5], which assumes that the speaker-irrelevant information are not necessarily added to the

speaker embeddings in a linear linear manner. Therefore, GDA addresses the issues of non-linear distortions.

In addition, we propose another variation of LDA named discriminant analysis via support vectors (SVDA) into the speaker recognition framework. SVDA calculates the within and between class covariance matrices using only the support vectors. In contrast to LDA, SVDA captures the boundary structure of the speaker classes (which is important in classification), and is shown to perform well for small sample size scenarios which is present in NIST SRE tasks (i.e. when the dimensionality is greater than sample size). The idea of using support vectors with discriminant analysis has been previously introduced in [50] which made significant improvement over LDA. In this chapter, the effectiveness of SVDA within an i-Vector/PLDA system will be evaluated for both long and short duration test segments and extended to a domain adaptation technique as well.

### 3.2 Linear Discriminant Analysis

LDA finds a linear transformation that maximizes the Fisher-Rao criterion. The separation of speaker classes in the direction of $W$ is equal to,

$$\lambda = \frac{W^T S_B W}{W^T S_W W}, \tag{3.1}$$

where $S_B$ and $S_W$ represents the between class and within class covariance matrices, respectively. When $W$ maximizes $S_W^{-1} S_B$, the class separation will be maximized as well. In other words, the eigenvectors corresponding to the largest eigenvalues in solving $\lambda S_W W = S_B W$ leads to the optimal projection matrix $W$.

For dimensionality reduction to $k$, the eigenvectors of the $k$ largest eigenvalues are placed in matrix $W$. Thereafter, the projected feature vectors are calculated by $W^T x$, where $x$ represents the input feature vector.

In Eq. 3.1, the terms $S_B$ and $S_W$ are defined as,

$$S_B = \frac{1}{C}\sum_{c=1}^{C} n_c(\mu_c - \mu)(\mu_c - \mu)^T \tag{3.2}$$

$$S_W = \frac{1}{C}\sum_{c=1}^{C}\sum_{k\in c}(x_k - \mu_c)(x_k - \mu_c)^T, \tag{3.3}$$

where $C$ represents the number of speaker classes and $n_c$ represents the number of samples in class $c$. In addition, $\mu_c$ and $\mu$ are the mean of class $c$ and overall mean of the samples, respectively.

## 3.3 Generalized Discriminant Analysis

i-Vector speaker embeddings are not accurate when utterances are of short duration; even with the presence of noise their robustness can be affected. Many studies have attempted to make i-Vector based systems more robust to noise [51], or short duration utterances [52, 53]. In other previous studies, the uncertainty of i-Vectors has been propagated back through the system, or different score calibration methods [52] have been introduced to partially address the problem. Here, we aim to study the effectiveness of GDA for long and short duration test segments on the NIST SRE-10 [39] task that already spans a range of distortions. We will assess the effectiveness of an i-Vector/PLDA system when GDA post-processed i-Vectors are employed. GDA is expected to help suppress general mismatch conditions which can occur at the application time; such as, when the recording condition is different than the development phase.

### 3.3.1 Method

Traditional LDA assumes data to be normally distributed and distinct classes share the same covariance matrix; from which it finds a linear transformation to map the input feature

vectors into a new subspace. On the other hand, GDA first maps the data into a new feature space and then finds a mapping linear transformation. Mapping to the new space is carried out using kernel methods. As the mapped feature vectors are non-linearly related to the input versions, GDA effectively provides a non-linear discriminant analysis for the input feature data [43].

More specifically, GDA first maps the feature vectors $x$ in space $X$ to the feature vectors $\phi(x)$ in space $F$. Next, the between and within class scatters are updated as (assuming observations are centered in $F$) as follows,

$$S_{Bf} = \frac{1}{C}\sum_{c=1}^{C} n_c \bar{\phi_c} \bar{\phi_c}^T \tag{3.4}$$

$$S_{Wf} = \frac{1}{C}\sum_{c=1}^{C} \sum_{k\in c} \phi(x_k)\phi(x_k)^T \tag{3.5}$$

where $\bar{\phi_c}$ is the mean of class $c$ in feature space $F$ (i.e., mean of $\phi(x)$ for $x$ in class $c$). To generalize LDA, we need to formulate the eigenvalue resolution problem in a dot-product format. Let us define the following kernel function:

$$k(x_i, x_j) = \phi(x_i)^T \phi(x_j), \tag{3.6}$$

where $i$ and $j$ range from 1 to the total number of training samples, (i.e., $n_x$). Next, define $K$ to be an $n_x \times n_x$ matrix containing $k(x_i, x_j)$. By defining the block-diagonal matrix $M = (M_c)_{c=1,\ldots,C}$ with the same size as $K$ for each $M_c = \frac{1}{n_c} \times I(n_c \times n_c)$; then Eq. 3.1 in the feature space $F$ can be formulated as,

$$\lambda_f = \frac{\alpha^T K M K \alpha}{\alpha^T K K \alpha}, \tag{3.7}$$

where $\alpha$ are the coefficient vectors that satisfy $\nu = \sum_{c=1}^{C} \sum_{k\in c} \alpha_k \phi(x_k)$; and $\nu$ are the eigenvectors of $\lambda_f S_{Wf} \nu = S_{Bf} \nu$. Since, the eigenvectors are linear combinations of feature

vectors in space $F$, there exist a non-unique set of $\alpha$ coefficients. More details are provided in [43].

To solve Eq. 3.7, matrix $K$ can be decomposed as:

$$K = P\Gamma P^T. \tag{3.8}$$

By defining $\beta = \Gamma P^T \alpha$ and replacing $K$ with Eq. 3.8 in Eq. 3.7 and simplifying the equation, we can reach the following eigenvector system:

$$\lambda_f \beta = P^T M P \beta. \tag{3.9}$$

For eigenvectors $\beta$, there exists $\alpha = P\Gamma^{-1}\beta$. From $\alpha$, the eigenvectors $\nu$ can be computed which leads to the necessary projection matrix in feature space $F$.

### 3.3.2 Experimental Setup

For all systems in the experiments, 60 dimensional Mel-frequency features are extracted, that include 19 dimensional static features as well as frame energy along with their delta and delta-delta coefficients. Speech signals have been framed using 25-ms length windows with a 10-ms skip rate. In addition, features have been normalized using a 3-sec sliding window. Next, energy-based speech activity detection (SAD) is used to remove non-speech frames.

Experiments are carried out on the NIST SRE-10 task [39], telephone condition (condition 5). A 2048-mixture UBM and total variability matrix have been trained using data collected from SRE2004, 2005, 2006, 2008, and Switchboard-II Phase 2 and 3, and Switchboard Cellular Part 1 and 2 (both male and female speakers). Next, 600-dimensional i-Vectors are extracted for all utterances. For LDA and GDA, the dimension size is reduced to 400, followed by length normalization. Training data for LDA, GDA, and PLDA is restricted to

male speakers from the NIST SRE2004, 2005, 2006, 2008 data. In addition, trials used for experiments contain only male enrollment and test segments.

The enrollment/test segment condition combinations that have been evaluated in this study, the number of speakers and segments for training the UBM, total variability matrix, LDA, GDA and PLDA, data used for enrollment, and statistics of trials have previously been described in Section 2.3.1. Core and extended core conditions (coreext) have durations ranging between 3 to 5 minutes.

To examine the effectiveness of GDA for short test segments, the coreext test data was truncated into 3-sec, 5-sec, 10-sec, 20-sec, and 40-sec segments. Extracting these short test data has been carried out after applying SAD; therefore, they do not contain non-speech frames. In addition, no modifications have been applied on the enrollment or training data.

### 3.3.3 Kernel Variations for GDA

The various kernel functions that are used in the experiments are presented in this subsection. The linear kernel leads to traditional LDA, where the between i-Vectors $w_1$ and $w_2$ are defined as,

$$k(w_1, w_2) = \langle w_1, w_2 \rangle. \tag{3.10}$$

We use this as a baseline discriminant analysis method. The cosine kernel is also used and defined as,

$$k(w_1, w_2) = \frac{\langle w_1, w_2 \rangle}{\parallel w_1 \parallel \parallel w_2 \parallel}. \tag{3.11}$$

The angles between i-Vectors is the only aspect captured by the cosine kernel. The study by authors in [14] states that the magnitude of i-Vectors may simply contain information about the channel and session which is not valuable in speaker recognition; therefore, when the cosine kernel suppresses the magnitude, we expect an improvement over a linear kernel.

The Within Class Covariance Normalization (WCCN) suppresses channel affects without removing any dimensions in the feature space. The projection matrix $B$ for WCCN is

achieved by a Cholesky decomposition of the within class covariance matrix in Eq. 3.3 as $S_w^{-1} = BB^T$. Here, we apply WCCN to the cosine kernel which updates it as (this kernel will be referred to "WCCN-Cosine" kernel in the experiments):

$$k(w_1, w_2) = \frac{(B^T w_1)^T (B^T w_2)}{\sqrt{(B^T w_1)^T (B^T w_1)}\sqrt{(B^T w_2)^T (B^T w_2)}}. \tag{3.12}$$

The other kernel variation (named as "LDA-Cosine") uses the LDA projection matrix in the cosine kernel. The background on LDA was provided earlier in Section 3.2. If we name the projection matrix as $A$, which are the ordered eigenvectors based on the highest values of eigenvalues, then the kernel would be,

$$k(w_1, w_2) = \frac{(A^T w_1)^T (A^T w_2)}{\sqrt{(A^T w_1)^T (A^T w_1)}\sqrt{(A^T w_2)^T (A^T w_2)}}. \tag{3.13}$$

Here, we have extracted 600-dimensional i-Vectors, and for this kernel the eigenvectors of the 600 largest eigenvalues have been selected as the projection matrix $A$. Therefore, we did not reduce the dimensionality of the i-Vectors with this projection matrix $A$; however, after transforming with $A$ and applying the cosine kernel, the dimension is reduced to 400, as is the case for the other kernels as well.

We also use the cascade of LDA and WCCN to project the feature vectors, and then employ the cosine kernel. LDA and WCCN have different objectives in finding the projection matrix; therefore, we examine their combination in the application of the kernel for GDA. We will refer to this kernel as the "LDA-WCCN-Cosine" kernel.

In [54], the authors proposed Gaussianized Cosine Distance Scoring (GCDS) that improves traditional cosine distance scoring. It was claimed there that estimating the WCCN projection matrix in noisy and/or channel mismatched conditions is difficult. Therefore, they replaced the cascade of LDA, WCCN, and cosine distance scoring with the GCDS method. Here, we take advantage of this idea and modify the algorithm to be used as a kernel function. Therefore, our proposed *Gaussianized cosine kernel* is derived based on the the following routine:

- The mean $m$ and standard deviation $v$ of the training data are first calculated.
- All data including training, enrollment and test data are Gaussianized. In other words, for every i-Vector $w$, the new vector will be modified to $w = \frac{w - m}{v}$.
- The Gaussianized i-Vectors are length normalized.
- The LDA projection matrix $A$ trained over the training data is calculated next.
- All data are then projected into the new feature space. In other words, for every i-Vector $w$, the transformed i-Vector will be $w = A^T w$.
- The new i-Vectors are length normalized again.
- Finally, these data are used in calculating the cosine kernel defined in Eq. 3.11.

The linear kernel in GDA is equivalent to the traditional LDA method, and is compared to the above-mentioned variations of cosine kernel. For training GDA and LDA, a smaller subset of the training speech segments (e.g., 10000 vs. 13605) are used (to limit the amount of memory needed); while, PLDA is trained on the entire male data set (e.g., 13605).

### 3.3.4 Results and Analysis

This subsection provides evaluation of speaker recognition comparing the effectiveness of LDA and GDA methods for discriminant analysis and dimensionality reduction.

To assess the system, we use Equal Error Rate (EER) and the minimum of decision cost function (minDCF) calculated as,

$$C_{Det} = C_{Miss} \times P_{Miss|Target} \times P_{Target} + C_{FalseAlarm} \times P_{FalseAlarm|NonTarget} \times (1 - P_{Target}). \quad (3.14)$$

Default values for parameters in this equation have been set as $C_{Miss} = C_{FalseAlarm} = 1$ and $P_{Target} = 1/1000$ as defined for the NIST SRE-10.

Results in terms of EER and minDCF are summarized in Table 3.1. The EER results show that in all enrollment/test segment condition combinations, GDA improves LDA. Specifically, Cosine and WCCN-Cosine achieve the best results for EER. For minDCF, just in coreext/coreext, coreext/coreext10sec, coreext/coreext40sec combinations, the linear kernel provides slightly better results, but in other cases GDA performed better.

In general, the improvement of GDA over LDA is more clear in longer duration test segments. Because i-Vectors extracted for shorter test data are not as accurate as the longer ones; therefore, non-linear discrimination cannot effectively locate them in their correct speaker classes.

Here, the cosine kernel performs the best among all kernel functions including the linear kernel (or LDA). After that, WCCN-Cosine and Gaussianized-Cosine and Linear kernels achieved effective performance. However, the LDA-Cosine kernel and LDA-WCCN-Cosine unexpectedly did not provide sufficient improvement over the other kernel functions. In summary, experimental results show that GDA is a promising dimensionality reduction and discrimination approach for i-Vector/PLDA system. With GDA, the relative improvement of 20% in EER and 18% in minDCF with core/core condition is achieved.

### 3.4 Support Vector Discriminant Analysis

Here, we apply another variation of LDA which we name as discriminant analysis via support vectors (SVDA) into the i-Vector/PLDA system. This approach can be integrated with other speaker embeddings as well, which is studied in more detail in Chapter 5. SVDA calculates the within and between class covariance matrices using only the support vectors, which represents the structure of speaker classes. In contrast to LDA, SVDA captures the boundary of the classes (which is important in classification), and performs well for small sample size problems which are typical in NIST SRE tasks (i.e., when the dimensionality is greater than the sample size). The idea of using support vectors with discriminant analysis

Table 3.1: Speaker recognition results comparing LDA and GDA using various kernel functions in terms of EER/minDCF. LDA and GDA reduce the dimension of i-Vectors from 600 to 400.

| | **LDA** | **GDA** | | | | |
|---|---|---|---|---|---|---|
| Enrollment/Test Segments | Linear | Cosine | WCCN-Cosine | LDA-Cosine | LDA-WCCN-Cosine | Gaussianized-Cosine |
| Core/Core | 1.416/.0353 | **1.133**/.0351 | 1.539/.0346 | 1.416/.0337 | 1.641/**.0289** | **1.133**/.0339 |
| Core/10Sec | 4.838/.0586 | 5.172/.0624 | 5.517/**.0575** | **4.828**/.0621 | 4.854/.0671 | **4.828**/.0596 |
| Coreext/Coreext | 1.438/**.0301** | **1.384**/.0309 | 1.558/.0319 | 1.44/.0320 | 1.789/.0315 | 1.414/.0323 |
| Coreext/Coreext3sec | 14.170/.0988 | 14.343/**.0978** | **14.113**/.0987 | 14.430/.0988 | 14.660/.0988 | 14.343/.0988 |
| Coreext/Coreext5sec | 9.770/.0949 | 9.783/**.0936** | **9.610**/.0943 | 9.755/.0959 | 10.085/.0956 | 9.812/.0956 |
| Coreext/Coreext10sec | 5.672/**.0755** | 5.722/.0791 | **5.628**/.0784 | 5.887/.0785 | 6.147/.0776 | 5.830/.0777 |
| Coreext/Coreext20sec | 3.319/.0592 | **3.27**/.0598 | 3.377/.0612 | 3.282/.0606 | 3.603/.0616 | 3.280/**.0590** |
| Coreext/Coreext40sec | 2.424/**.0451** | 2.403/.046 | 2.612/.0466 | 2.444/.0452 | 2.751/.0462 | **2.395**/.0460 |

has been previously introduced in [50] which made a significant improvement over LDA. In this study, the effectiveness of SVDA in the i-Vector/PLDA system has been evaluated on NIST SRE-10 and NIST SRE-16 speaker recognition evaluation [39] tasks with the telephony condition for both long and short duration test segments. In Chapter 5, it is extended to SRE-18 and other speaker embeddings as well. Compared to other dimension reduction approaches, from the aspect of the number of hyper-parameters, training time, and also the equal error rate (EER) and minimum detection cost function (minDCF) criteria, SVDA will be shown to be effective.

### 3.4.1 Method

The class separation measure for SVDA is similar to LDA; however, only the distinct support vectors will be used to calculate the within class and between class covariance matrices. More specifically, if we define $w_{c_1c_2} = \sum_{i=1}^{l} y_i \alpha_i x_i$ as the optimal directions to classify two classes $c_1$ and $c_2$ by a linear SVM ($y_i$ represents target value (+1 for first class, -1 for second class) of learning pattern $x_i$, and $\alpha_i$ is its coefficient), then the between class covariance matrix is updated as,

$$V_b = \sum_{1 \leq c_1 \leq c_2 \leq C} w_{c_1c_2} w_{c_1c_2}^T. \tag{3.15}$$

Also, let $\hat{X} = [\hat{x}_1, \hat{x}_2, ..., \hat{x}_{\hat{N}}]$ be the support vectors and $\hat{N}$ represent their total number. Therefore, the within class covariance matrix is formulated as,

$$V_w = \sum_{c=1}^{C} \sum_{i \in \hat{I}_c} (\hat{x}_i - \hat{\mu}_c)(\hat{x}_i - \hat{\mu}_c)^T, \tag{3.16}$$

where $\hat{I}_c$ includes the index of support vectors in class $c$, and $\hat{\mu}_c$ denotes their mean. Finally, similar to LDA, the optimum transformation $\hat{A}$ contains the $k$ eigenvectors corresponding to the $k$ largest eigenvalues of $V_w^{-1}V_b$.

From the aspect of classification, [50] showed that SVM performs better than LDA. For multi-class problems, the Fisher criterion in LDA finds the subspace that gives well-separated

classes more importance than those that are closer. Therefore, the classes that are already distinct will move further away from each other; however, the closer classes will not be treated the same. In contrast, SVM focuses more on the hard-to-separate classes. From this viewpoint, we expect that a transformation found by SVDA should work better for closer classes (or hard to separate speaker classes).

Moreover, the within class and between class covariance matrices calculated by SVDA only uses the support vectors instead of using all the training samples. Obviously, SVDA finds the discriminatory directions using the boundary structure of the classes; and also SVM is a well-known method for small sample size problems [50]. On the other hand, while addressing the SVM problem, we can adjust the tolerance of the classification error; therefore, generalization can be controlled more conveniently in SVDA rather than LDA.

To calculate the between and within class covariance matrices using SVDA, three strategies are considered:(i) traditional one-versus-one, (ii) weighted one-versus-one, and (iii) one-versus-rest. In the one-versus-one strategy, the SVM is applied to just two classes, therefore we need to model a total of $C(C-1)/2$ SVMs; in contrast to the one-versus-rest approach, where each class is classified against all data from all other speakers (i.e., need to train a total of $C$ SVM classifiers). As stated earlier, $C$ represents the number of speaker classes. It is worth mentioning that the one-versus-one strategy is more appropriate for an imbalanced data problem. The weighted one-versus-one has been designed to punish classes that do not have sufficient number of samples to define their structure (or may have noisy or random samples). In other words, some of the classes do not have well-defined structure and when we apply SVM, all the samples in the class are recognized as the support vectors. Therefore, by giving these types of classes smaller weight for their contribution in simply calculating $V_b$ in Eq. 3.15, the SVM classifier is forced to place more emphasis on the well-defined classes.

For NIST SRE-16 data and SRE-18 data, where a small unlabeled data set is provided from the target domain, SVDA can be extended to a domain adaptation technique as well.

The target domain data can be added to the rest class in the one-versus-rest strategy or we can assume all data belongs to only the one class and use them with a one-versus-one strategy. More details on using SVDA as a domain adaptation technique and the experimental results are provided in Chapter 4 and Chapter 5. In this chapter, we examine specifically the effectiveness of SVDA for the purpose of general dimensionality reduction in order to address non-linear distortions included in speaker embeddings.

### 3.4.2 Experimental Setup

The extracted feature vectors contain 19 Mel-frequency features as well as the frame energy appended with delta and delta-delta coefficients. The window length and shift size are 25-ms and 10-ms, respectively. In addition, a 3-s sliding window cepstral mean normalization is applied on the feature vectors. Non-speech frames are also discarded using energy-based speech activity detection (SAD).

Here, a 2048-mixture full covariance UBM and total variability matrix are trained using both male and female data collected from SRE2004, 2005, 2006, 2008 and Switchboard II phase 2,3 and Switchboard Cellular Part1 and Part2. Next, 600-dimensional i-Vectors are extracted. The dimension of the i-Vectors are then reduced to 400 using LDA/SVDA technique. Data used for training LDA, SVDA and PLDA is restricted to the male speakers (for the sake of tractability) from SRE2004, SRE2005, SRE2006, and 2008. To evaluate the system, we use male trials of the core and extended core conditions of NIST SRE-10. All experiments are carried out on the telephony condition (condition 5) of NIST SRE-10.

The enrollment/test condition combinations used in the experiments and the statistics of training and enrollment data, as well as trials were previously discussed in Section 2.3.1. In addition, to evaluate the performance of the system on short duration test segments, after applying SAD, the first 3, 5, 10, 20 and 40s of the extended core test data are extracted. For training the SVM, the publicly available LIBSVM [55] toolkit is used.

### 3.4.3 Results and Analysis

This subsection provides the experimental results comparing SVDA and LDA. Here for performance assessment, we used equal error rate (EER) and minimum detection cost function (minDCF) defined as,

$$C_{Det} = C_{Miss} \times P_{Miss|Target} \times P_{Target} + C_{FalseAlarm} \times P_{FalseAlarm|NonTarget} \times (1 - P_{Target}), \quad (3.17)$$

to evaluate our system. Based on the SRE-10 task, the performance weights are set as $C_{Miss} = C_{FalseAlarm} = 1$ and $P_{Target} = 1/1000$.

Table 3.2 summarizes the performance of i-Vector/PLDA speaker recognition comparing SVDA against LDA.

The results demonstrate that SVDA consistently improves LDA in terms of both EER and minDCF. For SVDA, the weighted one-versus-one strategy approximately works better than the traditional one (and both are better than one-versus-rest); these results meet our expectation that: first, the imbalance problem present in the data (some classes have less than 10 samples and some around 94) will be partly addressed with SVDA. More specifically, EER and minDCF is relatively improved by 32% and 5.6% (respectively) with traditional one-versus-one strategy. In addition, with weighted one-versus-one approach and punishing those classes that are not well-distinguishable with SVM (and probably contain noisy and error-full data), the relative improvement attained is 25% and 9% for EER and minDCF, respectively. Second, the capability of SVM for the small sample size problem has been confirmed (i-Vectors are 600 dimensional but there is not any class with more than 100 samples in the training set).

Table 3.3 reports results comparing LDA and SVDA without dimension reduction for core and extended core conditions. In terms of EER, SVDA outperforms LDA significantly; however, in terms of minDCF there is just a marginal improvement.

Table 3.2: EER/minDCF results comparing LDA and SVDA. The dimension of i-Vectors is reduced from 600 to 400.

| Enrollment/Test | **LDA** | **SVDA** | | |
|---|---|---|---|---|
| | | traditional 1-vs-1 | weighted 1-vs-1 | 1-vs-rest |
| Core/Core | 1.66 / .037 | **1.13** / .0399 | 1.25 / **.0364** | 1.42 / .0368 |
| Coreext/Coreext | 1.5 / .0297 | 1.35 / .0308 | **1.3** / **.0287** | 1.39 / .029 |
| Coreext/Coreext3sec | 14.5 / .0984 | 14.22 / .0974 | 14.23 / **.0974** | **14.2** / .0975 |
| Coreext/Coreext5sec | 9.71 / .0924 | 9.64 / .0915 | **9.55** / **.0909** | 9.81 / .092 |
| Coreext/Coreext10sec | 5.61 / .0759 | **5.58** / .0749 | 5.60 / **.0737** | 5.72 / .076 |
| Coreext/Coreext20sec | 3.17 / .0585 | **3.12** / .0574 | 3.17 / **.0573** | 3.35 / .0593 |
| Coreext/Coreext40sec | 2.48 / .0448 | 2.4 / .0423 | **2.37** / **.0407** | 2.42 / .0411 |

In summary, with regard to the number of hyper-parameters, computation time, and relative performance (approximate 32% relative improvement in EER), this section has shown that SVDA works well.

## 3.5 Summary

In this chapter, we studied the effectiveness of GDA in a state-of-the-art i-Vector based speaker recognition scenario using PLDA scoring. Most speaker recognition approaches use LDA to separate speaker classes and reduce the dimensionality of the feature vectors. Alternatively, GDA relaxes the linear separability of classes, which can be effective if unknown distortion or mismatch is present. We used the NIST SRE-10 core and coreext conditions for experiments, and results showed that GDA achieves effective gains for improving i-Vector/PLDA systems.

The cosine kernel, using WCCN before cosine kernel, and the Gaussianized cosine kernel achieved better performance compared to other kernel functions. The combination of LDA and WCCN before the cosine kernel did not provide improvement. The LDA used here merely separates the speaker classes and does not perform dimensionality reduction. In future effort, a cascade of LDA and WCCN with dimensionality reduction to various sizes is worth studing in combination with the cosine kernel. In addition, the relative gain for short test segments is not as comparable as that for the original long versions, which can be studied further.

In this chapter, the effectiveness of SVDA in the i-Vector/PLDA speaker recognition was also studied. Both EER and minDCF scores achieved from the experiments carried out on NIST SRE-10 task proved that SVDA consistently works better than LDA. In contrast to LDA that limits the discriminatory information to the centroid of classes, SVDA captures their boundary structure. In addition, the small sample size problem is well treated with SVDA. Although SVDA has a considerable improvement for longer duration test segments,

Table 3.3: Speaker recognition results comparing LDA and SVDA in terms of EER/minDCF without dimension reduction.

| Enrollment/Test | **LDA** | **SVDA** | |
|---|---|---|---|
| | | traditional 1-vs-1 | weighted 1-vs-1 |
| Core/Core | 1.58 / .039 | **1.45** / **.038** | 1.46 / .04 |
| Coreext/Coreext | 1.46 / .0302 | 1.37 / **.0301** | **1.36** / .0302 |

the decrease in EERs and minDCF is less for short duration test segments. In future work, the application of kernel SVM instead of traditional SVM in SVDA could be studied as well. In the next chapter, we consider our proposed deep learning based solution for compensating for domain mismatch in speaker recognition.

# CHAPTER 4

# DOMAIN ADAPTATION WITH DEEP LEARNING

Deep learning has made a great impact on various speech problems, specifically for applications on naturalistic data [56]. With the growing amount of data becoming available and improving computational resources, deep learning has became a powerful solution for many problems, including speech and language processing as well. Domain mismatch is a very general problem, which is currently under study for tasks such as image recognition and speech recognition. Deep learning based solutions have been shown to be effective in compensating for domain mismatch conditions as well. Here, we propose a solution to normalize speaker embeddings with domain-related information and make the speaker embedding independent of the domain, specifically designed for NIST SRE-16 task with integrating deep learning techniques.

Domain adaptation methods can be categorized into supervised and unsupervised techniques. Unsupervised domain adaptation is a very general solution and applicable to many scenarios. In addition, usually in-domain (or target-domain) data are very limited at the development phase, and even if they are labeled, they are not always effective. Depending on the task and the size of the models, very limited labeled data can even degrade the performance, as each class does not convey informative features. Therefore, combining all data together and use them in an unsupervised domain adaptation techniques has the potential of achieving a better performance. Here, we mainly focus on the unsupervised domain adaptation and in the next chapter we will compare them against supervised domain adaptation techniques.

## 4.1 Introduction

Speaker recognition is the task of recognizing whether an unknown speech segment was produced by a target speaker or not [4]. NIST has been organizing a series of speaker

recognition evaluations (SRE) for many years to evaluate new advancements in this area and continue to explore new challenges to address the recent concerns of automatic speaker recognition systems as well as more realistic data [3]. SRE-16 was focused primarily on domain mismatch problem (i.e., train, development and evaluation data belong to separate sets of languages). In addition, some other differences compared to previous SREs were introduced in SRE-16; such as, greater duration variability, providing a pool of unlabeled in-domain data, etc [3]. Interested sites world-wide submit their systems, where results confirm that there is still a wide gap to achieve effective performance for current mismatch challenges. In this chapter, we present our continued research for the NIST SRE-16 task and introduce new insights towards compensating for specific domain mismatch cases seen in the SRE-16.

In general, most submitted systems to the NIST SRE-16 challenge (as well as ongoing research after the challenge) used i-Vectors [26] to compress speaker identity of given speech segments to a fixed low-dimensional representation. However, variations are introduced in the traditional steps of extracting i-Vectors or calculating scores to suppress the domain mismatch. The key point here is investigating how to adopt unlabeled in-domain data.

In our solution [7], we extracted i-Vectors using both UBM and DNN based frameworks, where the UBM/i-Vector had significantly better performance, but UBM-based and DNN-based i-Vectors are complimentary and their score fusion helped with overall performance. Support vector discriminant analysis (SVDA), unlabeled probabilistic linear discriminant analysis (PLDA), mean normalization using unlabeled data are among the strategies we adopt to compensate for domain mismatch.

One group [21] used different feature sets, two classifiers and three alternate models. Their submitted system consisted of a fusion of four GMM/i-Vector systems with pairwise support vector machine (SVM), two DNN/i-Vector with pairwise SVM, and one GMM-SVM with Nuisance Attribute Projection (NAP). The latter system was trained on unlabeled data

which was clustered. They also studied other methods for unsupervised compensation, using in-domain data for MAP adaptation of GMM models which were shown to be effective.

For another team [25], their primary submission consisted of the fusion of four different i-Vector based systems. These four systems differed with respect to the feature vector which was then used for training the UBM, total variability (TV)-matrix and extracting the i-Vectors. For domain mismatch compensation, they applied multiple techniques: (1) whitening and mean centralization using in-domain data, and (2) multi-stage PLDA adaptation which also uses clustered unlabeled in-domain data.

Another submission [22] used different features (i.e., MFCC, PLP, and BNF) and classifiers (PLDA, discriminative PLDA, SVM, cosine distance, Latent Dirichlet Allocation). One new aspect in their submission was training a speaker classifier neural network for extraction of d vectors. They did not attempt to assign pseudo speaker labels to the unlabeled data.

The submissions to the challenge confirm that SRE-16 is a difficult task and needs further investigation. After the SRE-16 competition, different techniques were also proposed to overcome the challenges introduced in SRE-16 further. As an example, [23] applied an unsupervised Bayesian adaptation method and achieved promising results. On the other hand, [24] replaced i-Vectors with two new proposed embeddings which are derived based on a DNN architecture. They evaluated performance of the embeddings on both SRE-10 and SRE-16 tasks. In addition, domain mismatch has been previously studied for other databases or tasks as well, including [5, 16, 17, 18, 19, 20].

In this chapter, we focus on the NIST SRE-16 task for domain mismatch compensation, and present a solution based on deep learning techniques. Unfortunately, most of the previous studies on domain mismatch compensation for SRE-16 are a fusion of multiple complementary systems. Here, we focus mainly on developing a strong single system to more efficiently explain why one solution may be more effective than other.

## 4.2 a-Vector

Domain mismatch continues to be a major research challenge for speaker recognition in naturalistic audio streams. In this chapter, we present a new technique for domain mismatch compensation within a text-independent speaker recognition scenario. The proposed method is designed for the NIST speaker recognition evaluation 2016 (SRE-16) task, where speakers from training, development and evaluation data belong to different sets of languages. An i-Vector/PLDA speaker recognition system is adopted for this study. To address the mismatch problem, we propose to append auxiliary features to the i-Vectors. These auxiliary features are adapted representations of the i-Vectors to the specific in-domain data; therefore, the new feature vector has two parts: (1) the i-Vectors which represent speaker identity, and (2) the auxiliary features which are representations of i-Vectors in the in-domain data feature space (and may not contain speaker identity information). This new concatenated feature vector (we call this the *a-Vector*) is then post-processed with support vector discriminant analysis (SVDA) for further domain compensation. Evaluations based on the SRE-16 confirm the effectiveness of this proposed technique. In terms of minimum Cprimary cost, a-Vector outperforms i-Vector consistently. Moreover, comparing to previous single systems introduced for SRE-16, we achieved an 8.5%-18% improvement in terms of equal error rate. Here, our goal is to employ in-domain unlabeled data to achieve further compensation of domain mismatch. After the challenge, NIST provided ground-truth labels, but here we are not using them or applying any clustering to generate pseudo labels. The goal of our study here is leveraging unlabeled data to improve our system in an unsupervised manner. In addition, we note that score fusion of multiple complimentary systems always helps. However, here we do not want to focus on score normalization or calibration, our goal is to just focus on developing an effective single system. We propose using auxiliary and complimentary features in addition to i-Vectors. These features are specifically designed to only carry directions related to the in-domain languages. For this purpose, we train a simplified version

of the inception-v4 network [1] and propose a new loss function (we call it *domain-adapted triplet loss*).

### 4.2.1 Method

For NIST SRE-16 challenge, CRSS had 4 baseline systems (2 UBM/i-Vector and 2 DNN/i-Vector) and then developed 11 single systems based on that with different strategies to address the domain mismatch. Details of these systems are provided in [7]. Our best single system used a UBM/i-Vector speaker representation that was post-processed with SVDA, LDA and final scores were calculated with PLDA. SVDA was able to use unlabeled in-domain data without any pseudo labels. Based on our experiments and other participating sites, leveraging unlabeled data for the purpose of adaptation or normalization was the key point to achieve a good performance.

In this section, we propose a new method for domain mismatch compensation and our work has been inspired by [57]. The focus of [57] is on speech recognition and authors propose to incorporate i-Vectors as well for the input of DNN to provide speaker, channel and background normalization, and achieved a significant reduction in word error rate. Here, we propose new auxiliary features to be concatenated with the i-Vectors. These features are domain-adapted representations of i-Vectors and we derived them based on a convolutional neural network (CNN) and a new proposed loss function (which is a variation of triplet loss function and we call it *domain-adapted triplet loss*). In the rest of the dissertation, the concatenation of i-Vectors and the auxiliary features are referred to as *a-Vectors*. i-Vectors represent speaker-dependent information while auxiliary features are domain adapted representations which are used for the purpose of domain normalization. a-Vectors are post-processed with SVDA/LDA and likelihoods are calculated by PLDA similar to our best single system which is used here as the baseline. Details on the network architecture and the proposed loss function are provided in the following subsections.

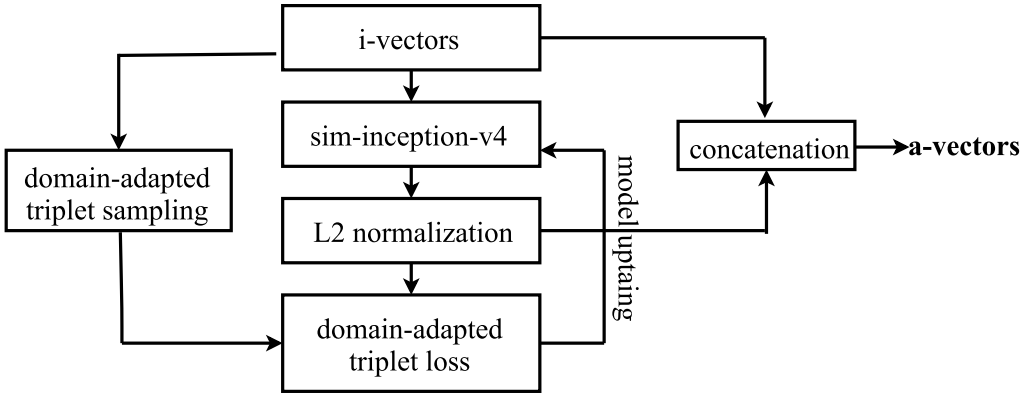


Figure 4.1: Overview of the system designed for generating auxiliary domain-adapted features and a-Vectors.

### Convolutional Neural Network a-Vector Representation

The proposed system for extracting auxiliary features is a simplified version of inception-v4 [1] and we call that sim-inception-v4. Our network takes i-Vectors as the input and generates the auxiliary features that minimize the loss function introduced in the following subsections. These auxiliary features are next concatenated with the i-Vectors and created the a-Vectors.

The overall system representation is shown in Fig. 4.1. The network architecture and the loss function are explained in more detail in continue.

### Network Architecture

The network is illustrated in Fig. 4.2. It has the stem part, inception-A and reduction-A part of the original inception-v4 network [1] (because of the limitations of our GPU we restricted the network layers). Details of the network is exactly the same as the inception-v4; however, tensors here are 1-D therefore the weight shapes of the convolution neural network are also changed to 1-D. The filter size on the remaining dimension is set to the exact values of the inception-v4. Please refer to [1] for more details of the system.

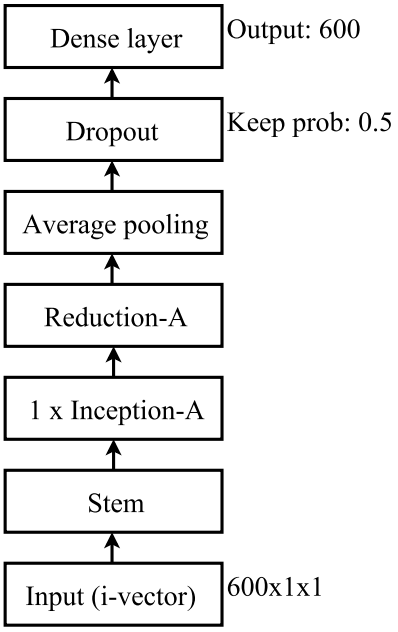


Figure 4.2: Simplified inception-v4 (sim-inception-v4) used for generating domain-adapted i-Vector. Please refer to [1] for details on the Stem, Inception-A and Reduction-A.

### Proposed Domain-Adapted Triplet Loss Function

The loss function proposed here is inspired by the triplet loss function. Triplet loss function was originally developed for FaceNet [34] and is also successfully applied to speaker recognition [58]. In [58] an end-to-end speaker recognition system is developed to estimate a new embedding as a replacement for the i-Vector, and triplet loss is applied to make sure that the embedding carries the speaker-related information. Here, we present a domain-adapted triplet loss which maps the inputs of the network to the in-domain feature space.

As Fig.4.1 shows, first i-Vectors are sampled into triplet sets. In the original triplet sampling, for an anchor feature vector one positive and one negative feature vectors are sampled; the positive one has the same speaker identity as the anchor one and the negative one has a different identity. Different strategies can be adopted for the selection of triplets [34, 58].

Domain-adapted triplet loss in contrast has a different meaning for the positive and negative samples. Here, for each anchor feature vector, the positive samples are in-domain

unlabeled vectors (both from minor and major languages) and the negative samples are out-domain vectors which are chosen from previous years SRE data subset.

The loss function used for the training of the network minimizes the distance between the anchor and positive samples and maximizes the distance between the anchor and negative samples. It is clear that loss function applies to the output of the sim-inception-v4 which is our output auxiliary feature vector.

If we represent anchor, positive and negative i-Vectors with $x^a$, $x^p$ and $x^n$ respectively and define $f(x)$ as the output auxiliary feature vector, then the network training process makes the $f(x)$ to satisfy the following relation:

$$||f(x_i^a) - f(x_i^p)||_2^2 + \alpha < ||f(x_i^a) - f(x_i^n)||_2^2, \quad \forall x_i^a, x_i^p, x_i^n \in T \tag{4.1}$$

where $T$ contains all possible triplets $(x^a, x^p, x^n)$, and $\alpha$ is a margin enforced between negative and positive pairs (we set $\alpha = 0.2$ in our experiments). Therefore, the loss function is defined as:

$$loss = \Sigma_{i \in T} max(0, \Delta_i). \tag{4.2}$$

where $\Delta$ is defined as:

$$\Delta_i = ||f(x_i^a) - f(x_i^p)||_2^2 - ||f(x_i^a) - f(x_i^n)||_2^2 + \alpha. \tag{4.3}$$

Generally, $T$ contains all possible triplets, but this set will be huge if we consider all combinations and makes the convergence slower [34]; therefore, we selected a smaller subset of that in the experiments. We chose 5 random samples from the in-domain data as the positive i-Vectors and 5 random i-Vectors from the previous SREs data as the negative ones.

The sampling for the domain-adapted triplet loss moves all auxiliary features toward the in-domain features and far from out-domain auxiliary features, in contrast to the original triplet loss which makes the same speaker embeddings closer and different speaker embeddings farther.

### 4.2.2 Evaluation Setup

#### UBM/i-Vector with PLDA Scoring

60-D MFCC features within a 25-ms window with 10-ms skip rate are extracted first. Next, non-speech frames are removed with energy based SAD. A 2048-mixture full covariance UBM and TV-matrix are trained using parts of fixed training data of SRE-16 (i.e., SRE2004, 2005, 2006, 2008 and Switchboard II phase 2,3 and Switchboard Cellular Part1 and Part2). Extracted i-Vectors are centralized with global mean calculated from major and minor in-domain unlabeled data, and then they are length normalized. Now, i-Vectors are concatenated with auxiliary features (output of sim-inception-v4). Output of the system is 600-D which is reduced to 150-D by PCA. The 750-D a-Vectors are then fed into SVDA and their dimension reduced to 500, next LDA reduces the dimension to 400. For training LDA and PLDA only previous years SRE data is used (SWD data is not used at the back-end at all). For SVDA, in addition to the SREs data, unlabeled in-domain data is also used; which is added to the rest class while training the SVM.

#### Sim-Inception-V4

In our experiments, for each epoch we randomly choose 500 speakers. All utterances of these 500 speakers are selected as anchors, among previous years SRE data 5 utterances are chosen randomly as negative samples and 5 utterances from in-domain data are chosen as positive samples. Learning rate starts with 1e-2 and after 50 epochs is set to 1e-3 and after

200 iterations is 1e-4. The maximum number of epochs is 1000. RMSprop optimizer is also used for the learning process.

### 4.2.3 Results and Analysis

This section presents experimental results comparing 4 different systems: (1) i-Vector + LDA, (2) a-Vector + LDA, (3) i-Vector + SVDA, (4) a-Vector + SVDA. In the tables i-Vector and a-Vector are referred to as ivec and avec for simplicity.

Table 4.1 and 4.3 summarize EERs for the DEV and EVAL respectively, results are reported for each language as well as on the pooled data. Table 4.2 and 4.4 also represent min-Cprimary for DEV and EVAL sets, respectively. For all cases, the SVDA-based systems perform better than the LDA-based ones. In table 4.2, ivec+SVDA has 3%/6% relative improvement over ivec+LDA; for avec-based one also SVDA has 4%/6% improvement over LDA. In table 4.4, ivec+SVDA achieved better performance over ivec+LDA with 12%/14% rate; and for avec one also SVDA achieved 13%/14% improvement over avec+LDA. Improvements for EVAL data is more significant comparing to the DEV data. Comparing a-Vector against i-Vector in table 4.2, the a-Vector one achieved 0.7%/0.6% and 2%/0.3% relative improvements for LDA and SVDA based systems. And for the EVAL data also has a similar range of improvements.

The results show that, SVDA consistently outperforms LDA, and improvements are more significant for min-Cprimary. Comparing i-Vector and a-Vector, in terms of min-Cprimary there is always a marginal improvement with a-Vectors. However, in terms of EER improvements are not consistent.

The results show that the proposed a-Vector is a promising representation; however, we believe that if in each iteration we present a better selection of triplet sets, clear and consistent improvement might achieve.

Comparing our proposed system against those **single systems** (systems with no score fusion) introduced in [23, 24], we achieved 8.5% and 18% improvements respectively in terms

Table 4.1: EER(%) equalized/unequalized scores on DEV

| System | Cebuano | Mandarin | Pool |
|---|---|---|---|
| ivec + LDA | 21.42 / 21.78 | 9.14 / 9.75 | 15.59 / 16.08 |
| avec + LDA | 21.09 / 21.66 | 9.02 / 9.66 | 15.93 / 16.28 |
| ivec + SVDA | 20.47 / 21.66 | 8.14 / 8.76 | 15.58 / 15.95 |
| avec + SVDA | 20.69 / 21.70 | 8.31 / 8.88 | 15.35 / 15.91 |

Table 4.2: min-Cprimary equalized/unequalized scored on DEV

| System | Cebuano | Mandarin | Pool |
|---|---|---|---|
| ivec + LDA | 0.9 / 0.841 | 0.488 / 0.481 | 0.701 / 0.671 |
| avec + LDA | 0.894 / 0.839 | 0.471 / 0.475 | 0.696 / 0.667 |
| ivec + SVDA | 0.877 / 0.799 | 0.464 / 0.453 | 0.679 / 0.629 |
| avec + SVDA | 0.868 / 0.797 | 0.462 / 0.452 | 0.668 / 0.627 |

Table 4.3: EER(%) equalized/unequalized scores on EVAL

| System | Tagalog | Cantonese | Pool |
|---|---|---|---|
| ivec + LDA | 17.08 / 17.02 | 7.65 / 8.46 | 12.42 / 12.68 |
| avec + LDA | 17.21 / 17.05 | 7.48 / 8.25 | 12.41 / 12.6 |
| ivec + SVDA | 15.20 / 15.23 | 6.05 / 6.88 | 10.66 / 10.95 |
| avec + SVDA | 15.27 / 15.27 | 6.01 / 6.88 | 10.7 / 11.04 |

Table 4.4: min-Cprimary equalized/unequalized scores on EVAL

| System | Tagalog | Cantonese | Pool |
|---|---|---|---|
| ivec + LDA | 0.902 / 0.906 | 0.606 / 0.617 | 0.797 / 0.806 |
| avec + LDA | 0.902 / 0.905 | 0.59 / 0.607 | 0.791 / 0.8 |
| ivec + SVDA | 0.829 / 0.818 | 0.53 / 0.55 | 0.698 / 0.697 |
| avec + SVDA | 0.828 / 0.815 | 0.527 / 0.55 | 0.689 / 0.691 |

of EER (their best performing single systems have been compared against here); and in terms of min-Cprimary a-Vector is competitive with those single systems (a-Vector achieved 0.689 and for those systems min-Cprimary are 0.686 and 0.689 respectively).

## 4.3 Summary

This chapter has presented a new method for compensation of the domain mismatch problem in SRE-16; and also extended the SVDA to a domain adaptation technique. The proposed solution was based on concatenation of domain-adapted auxiliary features and the original i-Vectors to normalize for specific language-dependent directions. For this purpose, we modeled a simplified version of the inception-v4 network to map i-Vectors to these new auxiliary features. During the training process, we also proposed a new loss function called domain-adapted triplet loss function. Evaluations are based on SRE-16 data, with reported EERs and min-Cprimary costs on DEV and EVAL sets confirming that the proposed method is promising in effectively addressing mismatch. In the next chapter, we consider multi-stage domain adaptation in speaker recognition.

# CHAPTER 5

# MULTI-STAGE DOMAIN ADAPTATION FOR SPEAKER RECOGNITION

Domain adaptation techniques can be applied at different stages in the speaker recognition framework; either in the front-end or in the back-end. For front-end domain adaptation, speaker embedding training module needs to use target domain data as well while learning the parameters of the model or adjusting the parameters. However, the small unlabeled data from the target domain makes the front-end domain compensation task harder and riskier. Robust speaker embeddings are essential to perform speaker recognition and achieve a good performance. On the other hand, domain adaptation in the embedding feature space or at the back-end level can be integrated into different steps or they can be combined together. The back-end processing focuses more on the speaker-irrelevant features. In this chapter, we present a comprehensive study of domain adaptation at the back-end level. We introduce new approaches for domain mismatch compensation. We apply them in each step alone or in combination with each other. In addition, we extend the domain adaptation to t-Vector and x-Vector speaker embeddings as well for both NIST SRE-16 and NIST SRE-18 tasks.

## 5.1 Introduction

Figure 5.1 shows the flow diagram of our back-ends with incorporating domain-adaptation methods. Although we carry out experiments on various combinations of domain adaptation techniques (together or separately), this figure summarizes the two different pipelines we find out to be successful and used for our submission to the challenge as well.

Based on our preliminary experiments, we propose performing domain mismatch compensation using either of the pipelines shown in Figure 5.1. In other words, domain mismatch is either compensated with SVDA or with adapted PLDA model (which can be supervised adapted PLDA or unsupervised adapted PLDA). In the latter case, scoring can be replaced

with s-norm scoring as well. Mean centralization, length normalization and LDA are shared between the two pipelines.

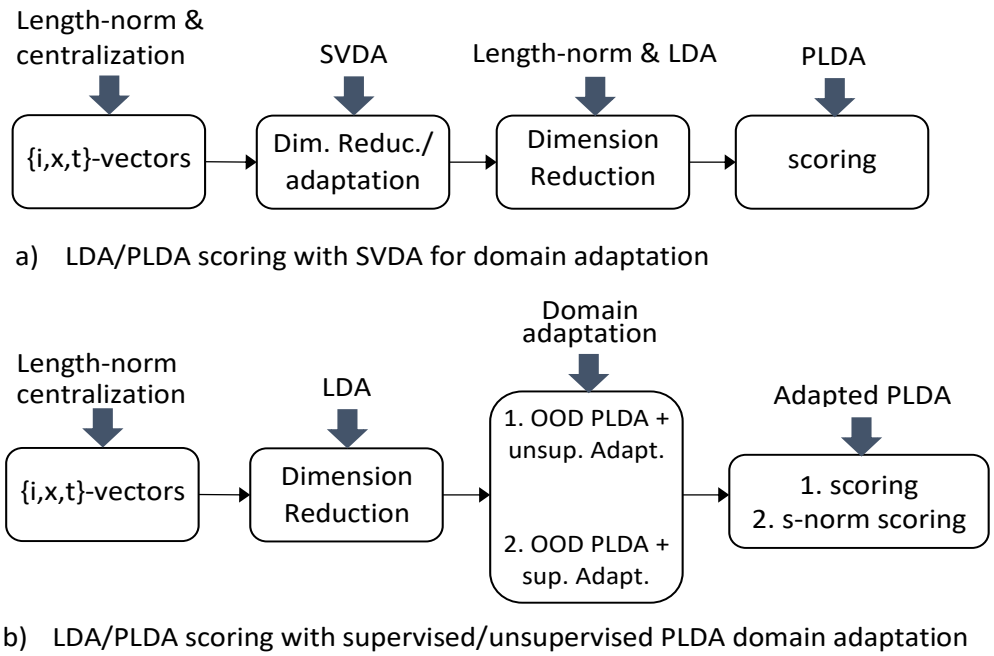


Figure 5.1: Flow diagrams of CRSS back-end classifiers [2].

## 5.2 Method

This section in detail describes applicable domain adaptation techniques at the back-end level processing of speaker embeddings. Mean-centralization with target-domain data is a simple yet very effective approach to transfer the speaker embeddings from the source domain to the target domain. The other steps, where domain adaptation can be integrated in are SVDA, LDA, PLDA, score normalization, calibration and fusion. Here, as we mainly focus on developing a single system for the target-domain, we provide solutions for adaptation in SVDA, LDA and PLDA steps, which are described in detail, in continue.

### 5.2.1 Adaptation with SVDA

In this subsection, we briefly describe discriminant analysis via support vectors [6] (which is studied before in Section 3.4 for dimension reduction purpose) and modify the SVDA framework for adaptation to the domain of interest.

SVDA is a variation of LDA, where both can be used for discriminant analysis, and optimize the Fisher criterion [59]. LDA uses all samples of all classes to calculate the between and within class covariance matrices, as:

$$S_b = \sum_{c=1}^{C} n_c(\mu_c - \mu)(\mu_c - \mu)^T \tag{5.1}$$

$$S_w = \sum_{c=1}^{C} \sum_{k \in c} (x_k - \mu_c)(x_k - \mu_c)^T, \tag{5.2}$$

However, SVDA only uses the support vectors to calculate the between and within class covariance matrices. More specifically, if we define $w_{c_1c_2} = \sum_{i=1}^{l} y_i \alpha_i x_i$ as the optimal direction to classify two classes $c_1$ and $c_2$ by a linear SVM ($y_i$ represents target value (+1 for first class, -1 for second class) of learning pattern $x_i$, $\alpha_i$ is its coefficient), then the between class covariance matrix will be updated as,

$$V_b = \sum_{1 \le c_1 \le c_2 \le C} w_{c_1c_2} w_{c_1c_2}^T. \tag{5.3}$$

Also, let $\hat{X} = [\hat{x}_1, \hat{x}_2, ..., \hat{x}_{\hat{N}}]$ be all the support vectors and $\hat{N}$ represents their total number. Next, the within class covariance matrix will be formulated as,

$$V_w = \sum_{c=1}^{C} \sum_{i \in \hat{I}_c} (\hat{x}_i - \hat{\mu}_c)(\hat{x}_i - \hat{\mu}_c)^T \tag{5.4}$$

where $\hat{I}_c$ includes the index of support vectors in class $c$, and $\hat{\mu}_c$ denotes the mean of them. Finally, similar to LDA, the optimum transformation $\hat{A}$ will contain the $k$ eigenvectors corresponding to the $k$ largest eigenvalues of $V_w^{-1} V_b$.

For training the SVM, two strategies can be adopted; (i.e., 1-VS-1 and 1-VS-Rest [6]). Data for the domain of interest can be easily integrated into this framework, both supervised and unsupervised. In the supervised adapted SVDA, first the in-domain data needs to be clustered (if they are unlabeled), then they will be treated similar to other speaker classes; in another experiment we considered all unlabeled data as belonging to only one single class and used it with a 1-VS-1 strategy. On the other hand, unsupervised adapted SVDA does not perform clustering. In every iteration of SVM, unlabeled in-domain data are added to the rest class with no information of their labels. Algorithm 1 summarizes our proposed 1-VS-Rest SVDA. Other advantages of our proposed SVDA includes: SVDA finds the discriminatory directions using the boundary structure of the classes, and also the SVM is a well-known method for small sample size problem [50].

**Algorithm 1** Algorithm for adapted-SVDA 1-VS-Rest.

C $\leftarrow$ Number of speaker classes
X, Y $\leftarrow$ i/t/x-Vectors, and their labels
N $\leftarrow$ Number of all support vectors
$\gamma \leftarrow$ Regularizer parameter $0 \leq \gamma \leq 1$, and here is set to 0.05.
**for** $i = 0$ to $C$ **do**
    $X_{cu} = X_i$ concatenate $X_{unlabeled}$
    $Y_{cu} = Y_i$ concatenate $Zeros(0, len(unlabeled))$
    $model = svmtrain(Y_{cu}, X_{cu})$
    $Ii =$ index of SVs for class $i$
    $Ij =$ index of SVs for unlabeled data
    $w = SVs(Ii) - mean(SVs(Ii))$
    $Vw = Vw + w^T * w$
    $w = SVCoef(Ii) * SVs(Ii) + SVCoef(Ij) * SVs(Ij)$
    $Vb = Vb + w^T * w$
**end for**
Vw $= (1-\gamma)V_w + \gamma\frac{trace(V_w)}{N-C}$
**return** eignenvectors corresponding to the k largest eigenvalues of $V_b a = \gamma V_w a$

As stated earlier, here, in addition, we perform the adapted SVDA with a 1-VS-1 strategy, which is summarized in Algorithm 2. We experimented with two different situations, (1) all

in-domain data are counted as belonging to one speaker class, and (2) we use the pseudo labels estimated from our clustering approach, which is described in the next subsection.

**Algorithm 2** Algorithm for adapted-SVDA 1-VS-1.

C ← Number of speaker classes
X, Y ← i/t/x-Vectors, and their labels
N ← Number of all support vectors
$\gamma$ ← Regularizer parameter $0 \leq \gamma \leq 1$, and here is set to 0.05.
$model = svmtrain(Y, X)$
$Vw$ ← Initialize to Zero
**for** $i = 0$ to $C$ **do**
    $Ii$ = index of SVs for class $i$
    $w = SVs(Ii) - mean(SVs(Ii))$
    $Vw = Vw + w' * w$
**end for**
Vw $= (1-\gamma)V_w + \gamma\frac{trace(V_w)}{N-C}$
$Vb$ ← Initialize to Zero
**for** $i = 0$ to $C-1$ **do**
    **for** $j = i+1$ to $C$ **do**
        $X_{cu} = X_i$ concatenate $X_j$
        $Y_{cu} = Y_i$ concatenate $Y_j$
        $model = svmtrain(Y_{cu}, X_{cu})$
        $Ii$ = index of SVs for class $i$
        $Ij$ = index of SVs for class $j$
        $w = SVCoef(Ii) * SVs(Ii) + SVCoef(Ij) * SVs(Ij)$
        $Vb = Vb + w^T * w$
    **end for**
**end for**
**return** eignenvectors corresponding to the k largest eigenvalues of $V_b a = \gamma V_w a$

### 5.2.2 Adaptation with LDA

Providing the data from domain of interest in order to find a discriminating transformation matrix can help with achieving better performance at the application step. LDA is a supervised dimension reduction tecnique; therefore adding the target-domain data into the training of LDA requires to provide labels for them as well. In this study, we use a simple

clustering technique to assign psedo labels to the unlabeled data and then integrated them in training of the LDA.

Generally, as mentioned before, for compensating the domain mismatch, the use of unlabeled in-domain data becomes very important. There are several stages where we can use unlabeled data, such as, LDA/PLDA training, and calibration; however, most of them require labeled data. Therefore, performing a speaker clustering of the unlabeled data is required there. After clustering unlabeled data, we can simply use the "estimated" speaker labels for each utterance with supervised methods. The clustering approach we applied here is similar to the method used by CRSS in 2015 NIST LRE i-Vector challenge. With these labels, we incorporate the in-domain information from unlabeled data to train both LDA and PLDA. In fact, in the experiments, this simple operation improves the LDA/PLDA baseline performance for the development set. In practice, we train a gender identification sub-task using previous SRE data before speaker clustering, and then apply a simple K-means algorithm over the gender dependent subsets, finally, we pool these two subsets together. Throughout our experiments, we found out this can provide more accurate speaker clustering and greater benefit to the subsequent LDA and PLDA training.

### 5.2.3 Adaptation with PLDA

Here, we perform PLDA adaptation with two different methods, (i.e., supervised and unsupervised adapted PLDA [17, 60]), details are provided in the following description.

For both supervised and unsupervised PLDA adaptation, $\Gamma$ and $\Lambda$ parameters, representing the between class and within class covariance matrices respectively [60] of PLDA model, need to be updated using the in-domain data. In the supervised adapted PLDA approach, the in-domain data are first clustered (the unlabeled data for SRE-16 & SRE-18 are in-domain data) and when their pseudo labels are estimated, we can perform the tradi-

tional PLDA on them. The $\Lambda$ and $\Gamma$ parameters of the supervised adapted PLDA are then interpolated as,

$$\begin{aligned} \Gamma_{adapt} &= \alpha\Gamma_{in} + (1-\alpha)\Gamma_{out}, \\ \Lambda_{adapt} &= \alpha\Lambda_{in} + (1-\alpha)\Lambda_{out}. \end{aligned} \tag{5.5}$$

Here, $\Gamma_{in}$ and $\Lambda_{in}$ are the between class and within class covariance matrices for the in-domain data, $\Gamma_{out}$ and $\Lambda_{out}$ are the same covariance matrices but calculated from out-domain data. In our experiments, we used $(1-\alpha) = 0.85$.

For unsupervised adapted PLDA, the in-domain data are not clustered first (if they are unlabeled; or their actual labels will not be used if they are labeled). Here, mean and variance of all in-domain data are calculated and used for adapting the PLDA covariance matrices as,

$$\begin{aligned} \Gamma_{adapt} &= \Gamma_{out} + \beta_b S, \\ \Lambda_{adapt} &= \Lambda_{out} + \beta_w S, \end{aligned} \tag{5.6}$$

where $\beta_b$ determines the scale for updating the between class covariance towards the excess variance in a particular direction, and $\beta_w$ is the same but for updating within class covariance matrix. In our experiments, we set $\beta_b = 0.2$ and $\beta_w = 0.6$. In addition, $S$ corresponds to the eigenvalues of adaptation-data (in-domain data) total-covariance in PLDA space [12].

### 5.3 Experimental Setup

For the UBM/i-Vector framework, we extract 60 dimensional features (20-D MFCC and $\Delta+\Delta\Delta$) on a 25ms window, with a shift size of 10ms. Non-speech frames are discarded using an energy-based speech activity detection (SAD). In addition, cepstral mean normalization is applied with a 3-second sliding window. 2048-mixture full covariance UBM and total variability matrix are trained for 600 dimensional i-Vector extraction. Next, LDA is used to reduce the dimension of the i-Vectors to 400-D.

In our study, we used the standard Kaldi x-Vector recipe to train our baseline x-Vector based system. The input feature vector is a 24 dimensional filter-bank from a 25-ms frame length analysis window, these features are then mean-normalized over a 3-s sliding window. Non-speech segments are removed using an energy-based SAD, though other more advanced SAD methods such as Combo-SAD [61] or TO-Combo-SAD [61, 62] could also be used for noisy data. The DNN configuration is described in detail in [12]. The resulting x-Vectors are 512 dimensional, which are then reduced to 150-D with LDA.

In t-Vector framework, high resolution filter bank features are adopted for system development. At the frequency axis, 96 dimensional log mel filter bank features are extracted from a 32-ms analysis speech frame, with a 50% overlap between neighboring frames. Non-speech portions of the utterance are removed using an energy based SAD. To deal with long duration samples in the training data, we uniformly segment the speech utterances into 12-second chunks without overlap, which is equivalent to 750-D feature set along the time axis as the input to the network. To estimate the embedding at the utterance level, we perform segment level embedding averaged in a sequential order, in order to obtain the t-Vector. Here, we extract 128 dimensional t-Vectors, which are then reduced to 80-D with LDA.

Table 5.1 summarizes the data used for training each of our developed speaker embedding systems for both SRE-16 and SRE-18 tasks.

Table 5.1: Corpora used in the speaker embedding system training.

| Dataset | Copora | Min-Utt/Spk | System |
|---|---|---|---|
| D1 | SRE04-08, SWB | 1 | i-vec |
| D2 | D1+Mixer 6 | 8 | t-vec |
| D3 | D2 + SRE-10 + VoxCeleb | 8 | x-vec |

Here, SWB includes all Switchboard II phase 2 & 3 and Switchboard Cellular Part 1 & 2 corpora. D2 and D3 listed in Table 5.1 are augmented by 3-folds after convolving with far-field Room Impulse Responses (RIRs), or by adding noise from the MUSAN corpus

Table 5.2: Number of speakers/segments used for training front-end and back-end processing within our speaker recognition system for this study.

| System | Front-end | Back-end |
|---|---|---|
| i-Vector | 5756/57273 | 3794/36410 |
| x-Vector | 13437/169135 | 3794/36422 |
| t-Vector | 5969/132777 | 3794/36422 |

[63]. The Kaldi x-Vector recipe is adopted for this portion of our processing. A speaker filtering criterion is applied to the training dataset as well for t-Vector and x-Vector feature extraction. For example, 8 min-utt/spk stands for the filtering process that all speakers with less than 8 utterances and less than 500 frames per utterance were excluded for training.

For training the back-end, no augmentation has been applied, our preliminary experiments showed that no gain can be obtained by including augmented data at the back-end training. The out-of-domain PLDA is also trained on only previous SRE data. SVDA, LDA and PLDA all share the same data. In the experiments where unlabeled data are included in training of SVDA, LDA and PLDA, it is explicitly mentioned in the dissertation. Statistics of the data used for training front-end and back-end stages are summarized in Table 5.2. For SRE-16, we report both equalized/unequalized scores.

## 5.4 Results and Analysis

In this section, we first perform experiments for evaluating the effectiveness of SVDA in domain adaptation. Table 5.3 summarizes results for i-Vector/PLDA solution for both SRE-16 and SRE-18. Three different SVDA variations have been applied: (1) 1-VS-1 strategy where all unlabeled in-domain data are considered to belong to one cluster; (2) 1-VS-1 where unlabeled data has been clustered first and their clustering (CL) labels used there; and (3) 1-VS-Rest where all unlabeled data are added to the rest class.

Results show that for SRE-16, SVDA achieves +15% and +14% improvement in terms of min-Cprimary and EER respectively. For SRE-18 as well, +8% and +16% improvement

were achieved with SVDA in terms of min-Cprimary and EER, respectively. For both SRE-16 and SRE-18, SVDA has contributed to domain adaptation. Based on these results, for the following experiment, 1-VS-Rest is used for domain adaptation in SRE-16; and 1-VS-1 (where all unlabeled data considered to belong to only 1 class) is applied for SRE-18.

Table 5.3: SVDA domain adaptation with i-Vector/PLDA for SRE-16 and SRE-18 tasks.

| **SVDA** | **DEV** | | **EVAL** | |
|---|---|---|---|---|
| | **EER** | **min-C** | **EER** | **min-C** |
| | | **SRE-16** | | |
| No SVDA | 15.59 / 16.08 | 0.7 / 0.67 | 12.33 / 12.55 | 0.79 / 0.8 |
| 1-VS-1 (all 1 class) | 15.77 / 16.05 | 0.7 / 0.65 | 10.75 / 11.04 | 0.7 / 0.69 |
| 1-VS-1 (CL labels) | 15.89 / 16.32 | 0.71 / 0.67 | 12.33 / 12.53 | 0.8 / 0.8 |
| 1-VS-Rest | **15.57 / 15.95** | **0.66 / 0.62** | **10.56 / 10.91** | **0.69 / 0.68** |
| | | **SRE-18** | | |
| No SVDA | 12.17 | 0.73 | 12.89 | 0.78 |
| 1-VS-1 (all 1 class) | **10.23** | **0.7** | **11.66** | **0.72** |
| 1-VS-1 (CL labels) | 12.07 | 0.74 | 12.85 | 0.77 |
| 1-VS-Rest | 12.01 | 0.72 | 12.92 | 0.78 |

In another experiment, we compare supervised and unsupervised PLDA adaptation methods for SRE-18 and SRE-16 tasks with i-Vector, t-Vector and x-Vector embeddings. Results are summarized in Table 5.4. Here, SVDA is not applied, in order to measure only the effectiveness of adapted PLDA. For x-Vector and t-Vector embeddings unsupervised adapted PLDA achieves consistent improvement over supervised adapted PLDA and original PLDA. However, for i-Vector, adapted PLDA does not provide a gain. For training t-Vector and x-Vector, we used augmented data but not for training the i-Vectors; which might affect the adapted PLDA.

Finally, to comprehensively examine domain adaptation at the back-end level, we use in-domain data along with alternate back-end blocks; LDA, SVDA, and PLDA. Results are summarized in Table 5.5. All systems use in-domain data first for centralization: major data is used for SRE-16 and unlabeled data for SRE-18. For t-Vector and x-Vector, unsupervised PLDA is used where ✓is set for PLDA. LDA needs labeled data for training; therefore, when ✓is on for LDA, the clustered unlabeled data is added to the training set. For SRE-16,

Table 5.4: Supervised VS Unsupervised PLDA, for SRE-16 and SRE-18.

| **Adapted PLDA** | **SRE-18** | | | | **SRE-16** | |
|---|---|---|---|---|---|---|
| | **DEV** | | **EVAL** | | **DEV** | **EVAL** |
| | **EER** | **min-C** | **EER** | **min-C** | **EER** | **min-C** |
| | | | **i-Vector** | | | |
| × | **12.17** | **0.73** | **12.89** | 0.78 | **12.33 / 12.55** | **0.79 / 0.8** |
| Supervised | 15.28 | 0.78 | 15.56 | 0.8 | 13.93/13.98 | 0.85/0.84 |
| Unsupervised | 14.86 | 0.73 | 16.04 | **0.76** | 13.96/14.23 | 0.8/0.8 |
| | | | **x-Vector** | | | |
| × | 11.4 | 0.78 | 11.23 | 0.77 | 15.32/15.56 | 0.99/0.99 |
| Supervised | 10.34 | 0.64 | 11.05 | 0.65 | 14.96/15.74 | 0.97/0.98 |
| Unsupervised | **8.82** | **0.54** | **9.64** | **0.56** | **8.37/8.29** | **0.6/0.61** |
| | | | **t-Vector** | | | |
| × | 13.34 | 0.88 | 13.87 | 0.87 | 17.2/16.15 | 0.99/0.99 |
| Supervised | 11.04 | 0.76 | 12.57 | 0.78 | 13.16/12.76 | 0.92/0.93 |
| Unsupervised | **9.5** | **0.53** | **9.62** | **0.67** | **9.17/9.32** | **0.7/0.72** |

1-VS-Rest SVDA is used and unlabeled data are added to the rest class; for SRE-18 1-VS-1 SVDA (where all unlabeled data are considered to belong to only one class) is used.

The scores for all the experiments confirm that domain adaptation at the back-end level is promissing, specially for x-Vectors and t-Vectors the improvement is more obvious. For i-Vector as well, SVDA is shown to be effective, specially for the SRE-16 where mean centralization is not adequate for domain adaptation. In i-Vector framework, discriminant analysis and dimension reduction techniques such as SVDA and LDA are shown to be more effective in compensating the domain mismatch rather than the PLDA. However, for x-Vectors and t-Vectors more gains are achieved with adapting the PLDA; however, SVDA is still resulting in better scores. For x-Vector embedding in SRE-18 task, with SVDA domain adaptation EER on EVAL set is 8.67% and with adapted PLDA is 9.63% which confirms that SVDA is a promissing approach to compensate for the domain mismatch.

## 5.5 Summary

In this chapter, we discussed multiple domain adaptation methods for speaker recognition for the NIST SRE-16 and SRE-18 tasks. We developed three alternate speaker embeddings

Table 5.5: Using data of interest, in-domain data in LDA, SVDA, and PLDA for x-Vector, i-Vector and t-Vector, evaluated on both SRE-16 and SRE-18.

| SVDA | LDA | PLDA | SRE-18 DEV EER | SRE-18 DEV min-C | SRE-18 EVAL EER | SRE-18 EVAL min-C | SRE-16 EVAL EER | SRE-16 EVAL min-C |
|---|---|---|---|---|---|---|---|---|
| **i-Vector** | | | | | | | | |
| × | × | × | 12.17 | 0.73 | 12.89 | 0.78 | 12.42 / 12.68 | 0.79 / 0.81 |
| ✓ | × | × | 12.01 | 0.71 | 12.92 | 0.78 | 10.66 / 10.95 | 0.69 / 0.69 |
| ✓ | ✓ | × | 10.76 | 0.7 | 12.34 | 0.76 | 10.69 / 11.02 | 0.69 / 0.69 |
| ✓ | × | ✓ | 12.27 | 0.71 | 13.05 | 0.78 | 12.58 / 12.77 | 0.82 / 0.83 |
| ✓ | ✓ | ✓ | 10.91 | 0.69 | 12.41 | 0.76 | 10.69 / 10.97 | 0.7 / 0.7 |
| × | ✓ | × | 11.15 | 0.73 | 12.35 | 0.75 | 12.32 / 12.56 | 0.77 / 0.78 |
| × | ✓ | ✓ | 11.41 | 0.72 | 12.47 | 0.76 | 12.4 / 12.51 | 0.79 / 0.79 |
| × | × | ✓ | 12.54 | 0.72 | 13.06 | 0.78 | 12.79 / 12.95 | 0.82 / 0.83 |
| **t-Vector** | | | | | | | | |
| × | × | × | 13.34 | 0.88 | 13.87 | 0.87 | 17.2 / 16.15 | 0.99 / 0.99 |
| ✓ | × | × | 11.93 | 0.7 | 10.45 | 0.74 | 13.12 / 12.86 | 0.89 / 0.94 |
| ✓ | ✓ | × | 11.71 | 0.7 | 10.4 | 0.73 | 12.98 / 12.91 | 0.94 / 0.97 |
| ✓ | × | ✓ | 9.79 | 0.57 | 9.96 | 0.66 | 10.01 / 10.29 | 0.71 / 0.72 |
| ✓ | ✓ | ✓ | 9.84 | 0.57 | 9.99 | 0.66 | 10 / 10.23 | 0.7 / 0.72 |
| × | ✓ | × | 13.7 | 0.89 | 14.55 | 0.9 | 22.43 / 21.43 | 0.99 / 0.99 |
| × | ✓ | ✓ | 9.52 | 0.54 | 9.65 | 0.67 | 9.23 / 9.36 | 0.71 / 0.73 |
| × | × | ✓ | 9.49 | 0.52 | 9.61 | 0.67 | 9.23 / 9.33 | 0.7 / 0.72 |
| **x-Vector** | | | | | | | | |
| × | × | × | 11.4 | 0.78 | 11.23 | 0.77 | 15.32 / 15.56 | 0.99 / 0.99 |
| ✓ | × | × | 8.86 | 0.61 | 8.67 | 0.58 | 11.45 / 11.14 | 0.86 / 0.89 |
| ✓ | ✓ | × | 8.89 | 0.65 | 8.72 | 0.59 | 13.36 / 12.91 | 0.99 / 0.99 |
| ✓ | × | ✓ | 8.7 | 0.54 | 9.96 | 0.57 | 8.71 / 8.46 | 0.58 / 0.59 |
| ✓ | ✓ | ✓ | 8.66 | 0.55 | 9.88 | 0.56 | 8.63 / 8.36 | 0.58 / 0.59 |
| × | ✓ | × | 16.97 | 0.87 | 13.93 | 0.86 | 29.36 / 27.24 | 1 / 1 |
| × | ✓ | ✓ | 8.55 | 0.54 | 9.37 | 0.56 | 8.45 / 8.23 | 0.62 / 0.63 |
| × | × | ✓ | 8.8 | 0.54 | 9.63 | 0.56 | 8.42 / 8.32 | 0.6 / 0.61 |

here, i-Vector, t-Vector and x-Vector. We explored the use of discriminant analysis with support vectors (SVDA), with new advancements from our previous methods. We evaluated the 1-VS-Rest SVDA strategy for domain adaptation. In addition, a new version of SVDA studied for speaker recognition using unlabeled data; 1-VS-1 where all unlabeled data is considered to belong to one cluster, and 1-VS-1 where unlabeled in-domain data was clustered. Results confirmed that SVDA can improve speaker recognition for SRE-16 and SRE-18 by +15% and +8% in terms on min-Cprimary respectively; and in terms of EER +14% and +16% respectively, with i-Vector speaker embeddings. Mean centralization, SVDA, LDA, and PLDA are phases which we incorporated in-domain data. We developed an effective con-

figuration for each of these steps to properly use the in-domain data. These results suggest effective steps towards improving domain adaptation for robust speaker recognition.

In the next chapter, we introduce the speaker de-identification task and explain our proposed solution.

# CHAPTER 6

# SPEAKER DE-IDENTIFICATION

Everyday a large amount data is uploaded on the internet and needs to be protected, for medical data to be processed privacy of patients needs to be preserved. These are only two examples that require a speaker de-identification technique. By definition, speaker de-identification means concealing any information that reveals the identity of the speaker. This information includes both linguistic and para-linguistic features. In this chapter, we focus on concealing the voice characteristics of the speaker for the purpose of protecting her/his identity. In addition, the performance of speaker recognition systems on automatic de-identification methods is examined here, to verify their robustness. If we are protecting the identity of a speaker, we need to achieve non-recognizable objective performance from state-of-the-art solutions for speaker recognition.

To perform speaker de-identification, we can simply adjust features of the speech, such as fundamental frequency, intonation, rate of the speech, etc. This engineering based solutions can provide effective performance in some scenarios. However, we need to have a prior knowledge on what aspects of the speakers voice is going to be captured and processed by the speaker recognition system. A more powerful solution is using voice conversion based techniques, which provides non-linear transformation from a source speaker to its de-identified version. With voice-conversion based techniques, the voice characteristics of the target de-identified speaker can also be adjusted easily. Here, for every source speaker we generate a new identity which is not present in the training set. Therefore, conveniently, we can apply our proposed speaker de-identification method to those applications that require multiple voices at the transformed subspace.

Speaker de-identification, on the other hand can be used as an augmentation method. When there is a need to increase the number of speakers in the dataset or add more variabilities, we can produce more samples with speaker de-identification and integrate the resulting

samples into system development; either for speaker recognition task or other applications of speech processing.

## 6.1 Introduction

Speaker de-identification is the task of concealing speaker identity, which may be revealed in linguistic (content of speakers speech) [64, 65] and paralinguistic (spectral and excitation features of the speech signal uttered by the speaker) [66, 67, 68] features. In this chapter, we focus on the latter one. Our goal is to map voice characteristics of a given speaker to a new identity, while preserving the naturalness and intelligibility. Speaker de-identification has many applications; for instance, protecting privacy of subjects speaking in a recording (e.g, witness or victim in courtroom/legal scenarios, voices played in some radio or television programs, and medical records), secure transmission of speech data (e.g., hiding speaker identity while transmission of speech data gathered from online banking services), prevention of unauthorized access, data augmentation, etc.

Previous studies in this research area are very limited. In [66] the authors proposed a method for protecting the privacy of speakers by adding masking sound based on white noise (considering different SNRs) or adding noise using band-pass filters. The authors showed that overall intelligibility decreases as the accuracy of protecting privacy increases. Generally, noise masking speech signals can unintentionally degrade intelligibility. In addition, masking just may decrease performance of speaker recognition systems, but subjectively listeners may still recognize the identity of the speaker. On the other hand, the authors in [68] used GMM-based and phonetic based speaker recognition systems for their evaluations. They transformed a source voice to a synthetic target voice called kal-diphone. Using synthetic voice as the target data degrades the performance of the de-identification system. In addition, authors in [69] manually defined piece-wise linear functions to transform the spectral parameters and achieved 4.4% - 98.6% accuracy with different settings; and

no subjective test has been reported. Authors in [70] adopted an available transformation method, i.e., the weighted frequency warping. They proposed a new method for selection of a speaker from the database. Transformation applies on the source speaker toward this selected speaker. The selection method is designed to meet three different criteria to achieve an overall promising performance.

Here, in contrast to other related works (to the best of our knowledge, they all used an already available voice transformation method), we propose a new convolution encoder-decoder based voice mapping and incorporated that into our speaker de-identification system. We use the publicly available database of voice conversion challenge 2016 (VCC-2016) [71, 72] to develop our voice mapping system. The convolution neural network (CNN) voice mapping architecture is specifically designed to consider details of the database and has the ability to suppress the errors and noises that might occur during the preparation of data for the voice mapping step. Finally, the speaker de-identification system employs the voice mapping module to transform the voice characteristics of a given speaker to all target speakers in the database. Average or gender-dependent average of mapped voices leads to the de-identified voice.

As a brief summary, the main contribution of this study is the development of a new voice mapping system using convolutional encoder-decoder neural networks. Other key aspect of this study is that we evaluate the proposed architecture with an i-Vector/probabilistic linear discriminant analysis (PLDA) [14] speaker recognizer. In addition, the de-identification approach proposed here is designed to gain good performance with both human listeners and machines while preserving quality and naturalness.

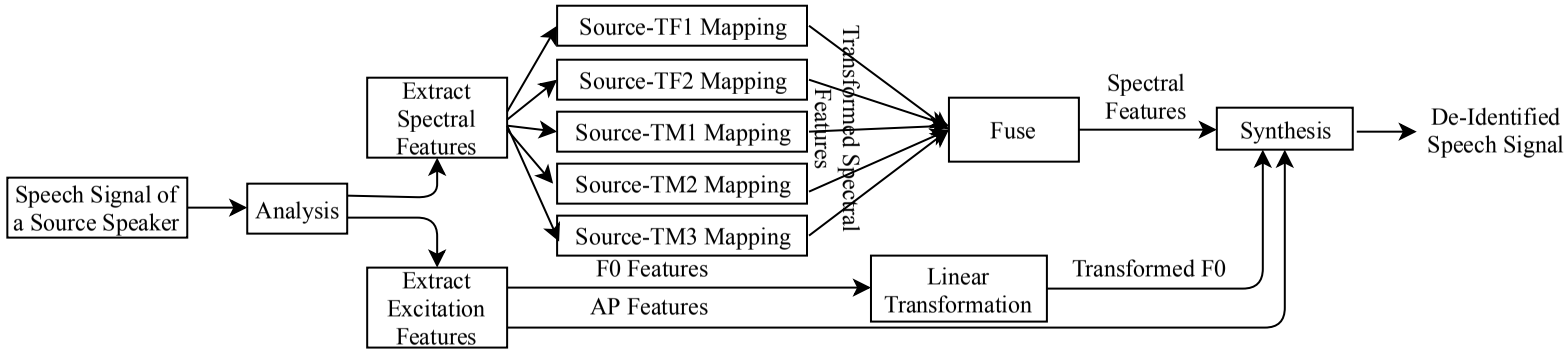


Figure 6.1: The overall block-diagram of proposed speaker de-identification.

## 6.2 Method

The details of the proposed speaker de-identification architecture are described in this section. We first introduce the main idea and the overall architecture of the proposed system, and then explain each individual subsystem in detail.

Figure 6.1 shows the overall block-diagram of the proposed system. Based on this figure, the proposed system performs the speaker de-identification in the following 4 steps:

- ***Feature analysis:*** In Subsection 6.3.2, the features are introduced which contain spectral (MCEP) and excitation (AP, Log-F0) features. They are extracted using the STRAIGHT vocoder.

- ***Feature mapping:*** The VCC-2016 database has 5 source and 5 target speakers. 25 (i.e., every potential mapping from any source to any target) mapping functions from MCEP features of the source to the MCEP features of all target speakers are trained based on a new convolutional encoder-decoder neural network architecture (which is described in detail in Subsection 6.2.1). To preserve the variance of training data and partially resolve the over-smoothing problem, we simply scale the variance of the generated MCEP features to that of the same speaker in the training data. For Log-F0 a simple linear transformation is applied. In addition, AP is moved directly from the source speaker to the de-identified speaker without any modification.

- ***Fusion:*** For a given source speaker, we map the MCEP and Log-F0 features to all target speakers in the database (based on the technique explained in the previous step). Next, mapped features are fused together with two different approaches: (i) average, and (ii) gender-dependent average. It is clear that we can also apply weighted averaging to obtain different voices for each source speaker, but in this study for the sake of simplicity we use equally weighted averaging.

- ***Synthesis:*** In this step, transformed MCEP features are converted back to SP using SPTK toolkit. We stack the SP, F0 as well as the AP (obtained from the previous steps) and use STRAIGHT synthesis module to generate the de-identified speech samples.

### 6.2.1 Convolution Encoder-Decoder Mapping

This subsection introduces a new neural network architecture for mapping acoustic features from a source speaker to a target speaker. Similar to all neural networks, our mapping network has train and test (de-identification) phases.

In the training phase, source and target utterances are first aligned using the dynamic time warping (DTW) algorithm. Next, we prepare the data for our training procedure. The input and output of the network are stacks of 15 consecutive frames of MCEP features. We can interpret these 15 frames as one frame that is appended with 7 previous and 7 next frames. Finally, the mapping network is trained to model the non-linear transformation from the input sequence to the output sequence.

In the de-identification phase, we slide a 15-frame-length window over the input sequence and feed each window as the input to the trained network. The network transforms the input into the same dimensional output; however, we only keep the middle one.

In this study, we introduce a new convolutional neural network (CNN)-based structure to perform the spectral mapping. CNNs represent a variation of neural networks [73] which have a unique structure with a cascade of convolution and pooling layers. Three key CNN aspects benefit our task: local connectivity, weight sharing, and pooling [74]. Local connectivity makes the system more robust to noise. In addition, while static features are sufficient for the network, the benefits of using dynamic features are captured by CNN filtering. Also, weight sharing reduces the number of parameters which partially addresses the issue of over-fitting. Specially here, the VCC-2016 database is small (this can be valid for most of paralleled databases) and can cause an over-fitting problem. Pooling as well can help suppress potential errors of dynamic time warping (DTW) for aligning the two feature sets.

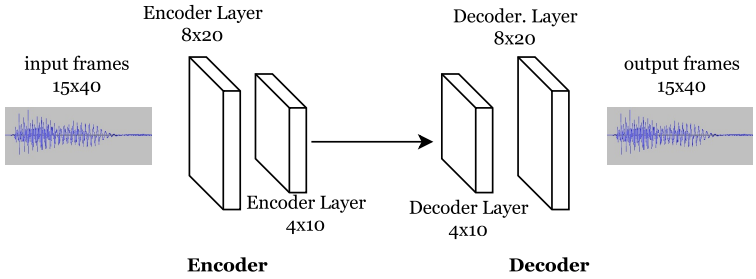


Figure 6.2: Convolutional encoder-decoder architecture.

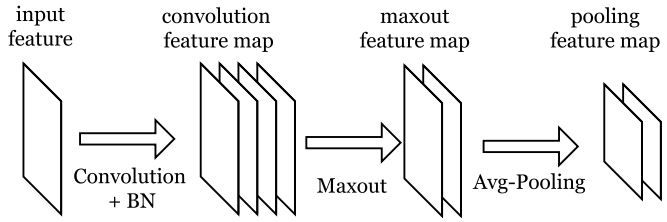


Figure 6.3: Encoding layer: encodes input into a lower dimensional representation. BN is batch-normalization. Each convolution layer uses maxout and is followed by average pooling.

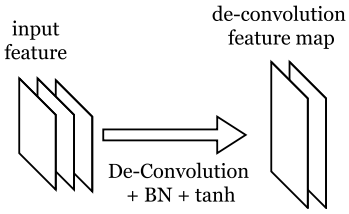


Figure 6.4: Decoding layer: decodes input. The activation function is tanh, and BN is batch normalization.

Various approaches have been introduced to convert spectral features. Examples include, joint density Gaussian mixture model (JDGMM) [75] with parameter generation algorithm [76] (to incorporate dynamic features) which are traditional methods for converting spectral features. LSTM-RNN [77], stacked joint-autoencoder [78], generative training of DNN [79], exemplar-based conversion [80] are among more recent trends in voice conversion. In addition, [81] proposed (combining different techniques including) applying direct waveform modification using spectral differential filtering (DIFFVC) with GMM-based VC and ranked one of the top systems in the VCC-2016 [72].

Table 6.1: EER (%) for original source (Female: SF1, SF2, SF3; Male: SM1, SM2) and target (Female: TF1, TF2; Male: TM1, TM2, TM3) speakers.

| | SF1 | SF2 | SF3 | SM1 | SM2 | TF1 | TF2 | TM1 | TM2 | TM3 |
|---|---|---|---|---|---|---|---|---|---|---|
| EER (%) | 2.516 | 0.559 | 0.4892 | 0.1747 | 0.7687 | 1.747 | 0.3494 | 0.4542 | 0.2096 | 0.2795 |

CNN-based mapping has multiple advantages over other voice mapping methods which include: (1) Compared to approaches that use delta features to capture time-dependencies (such as, JDGMM), our convolutional encoder-decoder network is able to automatically capture the dependencies between adjacent acoustic feature frames without including dynamic features. As a result, our method does not need a parameter generation algorithm which is prone to the over-smooting problem. (2) Compared to LSTM-RNN approaches, our network is faster and easier to train. Additionally, due to the recurrent nature of LSTM-RNN, it cannot fully exploit the GPU capabilities.

Figure 6.2 shows the overall structure of the proposed convolutional encoder-decoder. As it is shown in the figure, the structure contains an encoder followed by a decoder. Encoder is a stack of convolution and pooling layers (Figure 6.3) and decoder is a stack of convolution-transpose[1] (Figure 6.4) layers. The convolution and pooling layers encode the input into low resolution representations and convolution-transpose layers up-sample the data to its original high resolution space. Applying convolution-transpose after the convolutional layers has shown to be effective in other applications; such as image segmentation [83], emotion recognition [82], etc.

[1]Also known as de-convolution, up-convolution, backward strided convolution and fractionally strided convolution [82]

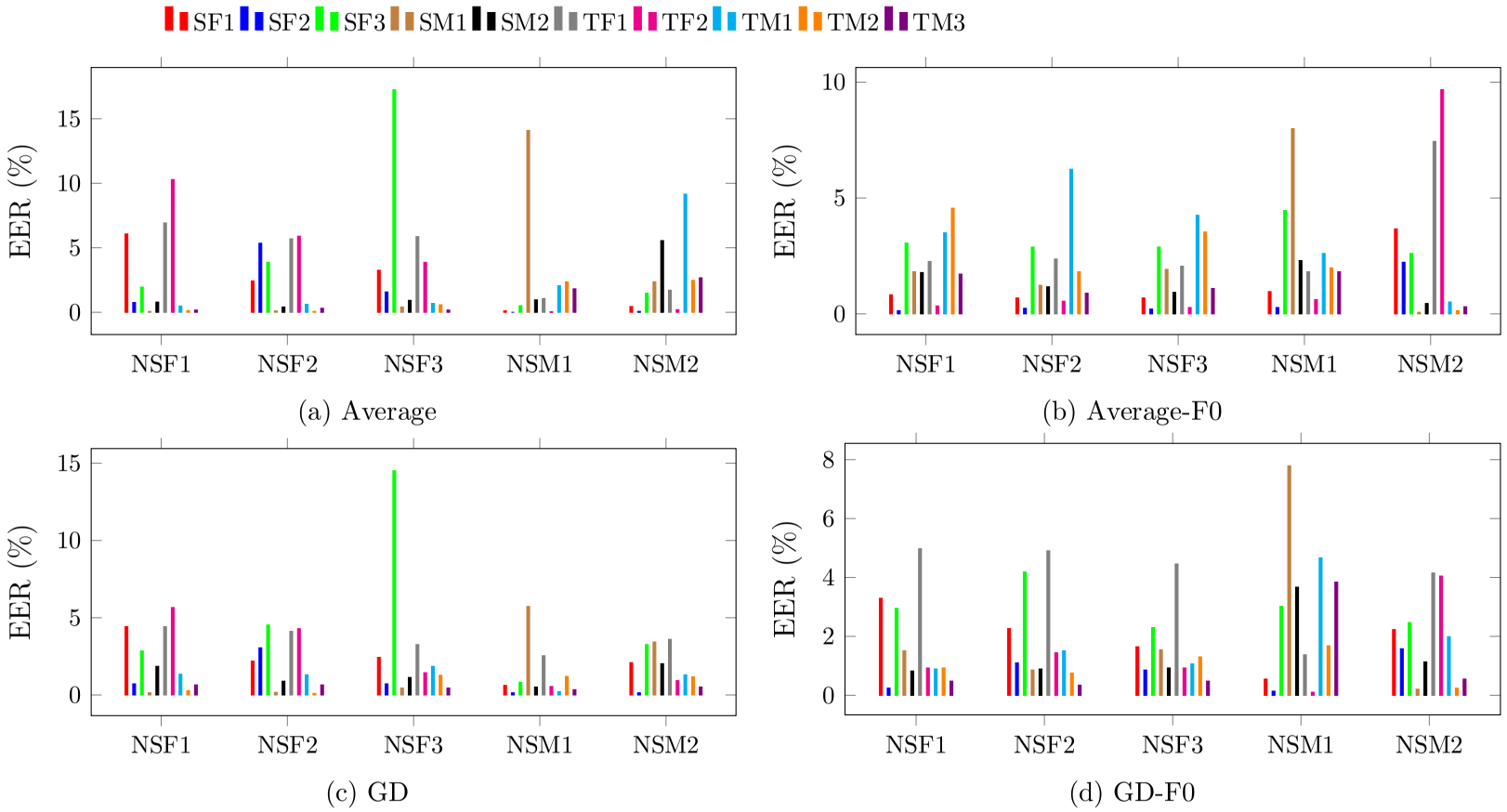


Figure 6.5: EER(%) results for four different systems: a) Average, b) Average-F0, c) GD, d) GD-F0. For every newly generated speaker, the equal error rate against available 10 speakers in database is reported.

## 6.3 Experimental Setup

### 6.3.1 Data

We use publicly available database of voice conversion challenge 2016 (VCC-2016) [71, 72] here. This database is specifically designed for the voice conversion application. In voice conversion, we map a source speaker to a target speaker. This database contains speech data of 10 speakers, 5 source speakers (SF1, SF2, SF3, SM1, SM2) and 5 target speakers (TF1, TF2, TM1, TM2, TM3). S and T represent source and target speakers respectively; in addition, M and F refer to male and female. This database is parallel; i.e., all speakers read the same set of sentences. Each speaker has 162 training and 54 test utterances. For developing our systems, we use 150 training utterances for modeling, and the remaining 12 utterances as development data.

The key points that lead us to choose this database include: (1) to preserve linguistic information we need a parallel database [71] (however, there are also growing studies on non-parallel data [84] as well). (2) to the best of our knowledge, there is not any other publicly available parallel database designed specifically for speech synthesis or voice conversion rather than CMU-ARCTIC [85] (which only has 7 speakers); therefore, we chose VCC-2016 as it has more speakers.

Table 6.2: Summary of results. The EER(%) in figure 6.5 are averaged here for each newly generated speaker.

| Voice De-ID | NSF1 | NSF2 | NSF3 | NSM1 | NSM2 |
|---|---|---|---|---|---|
| Average | 2.764 | 2.476 | 3.46 | 2.301 | 2.613 |
| Average-F0 | 1.999 | 1.807 | 1.783 | 2.483 | 2.709 |
| GD | 2.232 | 2.129 | 2.751 | 1.265 | 1.845 |
| GD-F0 | 1.701 | 1.824 | 1.550 | 2.682 | 1.859 |

### 6.3.2 Features

In the proposed speaker de-identification system, we first compress speech into a set of acoustic features, we then de-identify speaker information in the extracted feature space, and finally we synthesize de-identified speech from the acoustic feature space. The STRAIGHT vocoder [86] is used for analysis and synthesis of utterances. STRAIGHT is a high quality vocoder that introduces around 0.5 MOS degradation in the naturalness of the speech signal [87]. STRAIGHT extracts 513-D spectral envelope (SP), 513-D aperiodicity (AP) features as well as 1-D fundamental frequency (F0). We employ speech signal processing toolkit (SPTK) to convert SP to 40-D Mel-cepstral coefficients (MCEP) [88]. The de-identification is only applied to MCEP and F0 features; the AP features are directly mapped from the source speaker to the de-identified speaker.

### 6.3.3 Evaluation Metrics

In this subsection, the metrics used for evaluation of the proposed speaker de-identification system are explained. Experiments are categorized in objective and subjective tests.

For objective evaluations, we developed an i-Vector/PLDA based speaker recognition [14] system which is explained in 6.3.4. Similar to other speaker recognition evaluations, we report equal error rate (EER) to evaluate and compare the performance of the developed systems [4]. EER measures the error rate of a system at the threshold that miss alarm and false alarm are equal [4].

For subjective evaluations, we conducted an informal subjective test. Details of the experiment are described in 6.3.5.

### 6.3.4 Speaker Recognition Evaluation

Speaker recognition is the task of recognizing whether a given utterance belongs to a target speaker or not. Here we employ an i-Vector/PLDA speaker recognition solution.

As mentioned before, in typical i-Vector/PLDA speaker recognition systems, mel frequency cepstral coefficients (MFCCs) are first extracted as input feature vectors, then speech activity detection (SAD) is applied to remove non-speech segments. Next, a UBM and total variability matrix (TV) are trained, and i-Vectors are extracted. Thereafter, i-Vectors are post-processed with length-normalization and LDA. Eventually, PLDA is trained and final log-likelihood scores are calculated [7].

In detail, the speaker- and channel-dependent GMM supervector in the i-Vector configuration is factorized as [14],

$$M = m + Tw, \tag{6.1}$$

where $m$ is the speaker and channel-independent UBM supervector, $T$ is total variability (TV) matrix that maps the high-dimensional GMM supervector to a lower-dimensional vector $w$; or so-called i-Vector representation [14].

The expectation maximization (EM) algorithm is used to train both UBM and TV matrix. In the E-step, $w$ is considered a latent variable with a normal prior distribution $N(0, I)$. At the end of the optimization, the estimated value for each i-Vector is the mean of the posterior distribution of $w$ [14]. The estimated i-Vector is:

$$\hat{w}(u) = (I + T^T \Sigma^{-1} N(u) T)^{-1} T^T \Sigma^{-1} S(u), \tag{6.2}$$

where $\Sigma$ is the UBM covariance matrix. In addition, $N(u)$ and $S(u)$ are zeroth and centralized first order Baum-Welch statistics for utterance $u$, respectively.

### 6.3.5 Naturalness Evaluation

For subjective evaluation, we conducted mean opinion score (MOS)-naturalness test. 20 listeners participated in the evaluations. Listeners are asked to rank the naturalness of 50 randomly chosen utterances from 1 (bad) to 5 (excellent).

### 6.3.6 Experimental Conditions

This section describes details on the database used for training the i-Vector/PLDA speaker recognition system and CNN configuration we adopted in developing of our system.

For the speaker recognition system, we first extract 19 MFCC features and append them with energy, delta, and delta-delta features using a 25-ms window with sequential 10-ms frame shifts. Next, energy-based SAD is used to remove non-speech segments. A 2048-mixture full covariance UBM and total variability matrix are trained using data collected from SRE2004, 2005, 2006, 2008 and Switchboard II phase 2,3 and Switchboard Cellular Part1 and Part2 [5, 6]. For training both LDA and PLDA, we use training data from VCC-2016 database. The enrollment/test data also includes test utterances from VCC-2016 database.

The CNN encoder-decoder introduced in Sec. 6.2.1 uses 2 convolution and 2 convolution-transpose layers. The first convolution layer converts 15x40-D to 15x40x256, which reduces to 15x40x128 with maxout. Next, average pooling is used to reduce the dimensions to 8x20x128. In the second convolution layer, the 8x20x128 input data is converted to 4x10x512 and again reduces to 4x10x256 with maxout and average pooling. The filter size in CNN for the first and second CNN layers are 9x9 and 3x3. In the decoding layers (convolution-transpose layers) tanh activation function is applied in both layers. The filter size for the first and second convolution-transpose layers are 3x3 and 9x9, respectively. In all CNN and convolution-transpose layers, we used batch normalization.

The minimum mean square error (MSE) has been chosen as the optimization criterion and both L1 and L2 regularization are used here to solve the over-fitting problem. The learning rate starts with 0.01 in initial epochs and decreases gradually. The maximum number of epochs is set to 1000. Adam optimization is also used here for training the model.

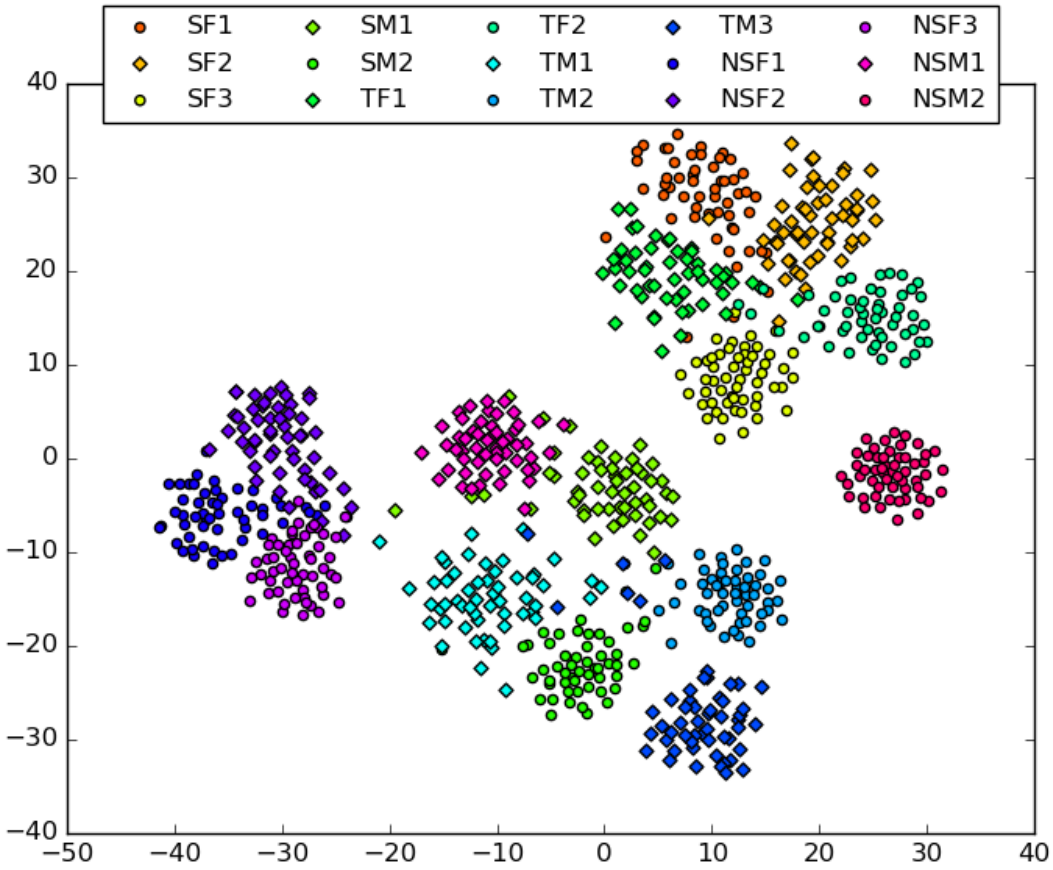


Figure 6.6: 600-D i-Vectors mapped to 2-D representation with t-SNE. Each source speaker is mapped to the average of target speakers and new identity is generated. For example, NSF1 is de-identified version of SF1 which is generated by mapping SF1 to average of all target speakers.

## 6.4 Results and Analysis

### 6.4.1 Objective Test

In this subsection, we evaluate the proposed architecture in terms of equal error rate (EER).

First, we evaluate the performance of the i-Vector/PLDA speaker recognition for VCC-2016 database. The EER for each individual speaker is shown in Table 6.1. The results show the average EER for all speakers is 0.75% which is reasonable.

Next, for each source speaker (SF1, SF2, SF3, SM1, SM2) we generate a new speaker (NSF1, NSF2, NSF3, NSM1, NSM2). For example, NSF1 is created by using the AP and F0 (or linear transformation of F0) of SF1 and average/weighted-average of MCEP features generated by voice mapping systems; specifically, transformation from SF1 to all target speakers (TF1, TF2, TM1, TM2, TM3). We claim that NSF1 is different from all available speakers in the database (all source and target ones). The results are presented in Figure 6.5. We designed trials in a way that smaller EERs represent better de-identification performance.

Figure 6.5 represents four different approaches for fusing spectral features of different target speakers.

- **Average**: AP and F0 of the source speaker are directly (without any change) copied to the new speaker; while, the MCEP features are equally weighted average of transformation to all target speakers.

- **Average-F0**: this is exactly similar to the previous version except that F0 of the new speaker (e.g., NSF1) is a linear transformation of F0 for SF1. Here, if the source speaker is female, we decrease F0 by 10% and if the source speaker is male, we increase F0 by 10%.

- **Gender-dependent (GD)**: in this system F0 and AP are copied from the source speaker to the de-identified speaker. However, the MCEP features are the average of only the cross-gender voice mapping models. For example, for NSF1, we average MCEP features generated by SF1-TM1, SF1-TM2, SF1-TM3 voice mapping systems (in contrast to average between all SF1-TF1, SF1-TF2, SF1-TM1, SF1-TM2, SF1-TM3 voice mapping systems which we had in the first system; i.e. "Average").

- **GD-F0**: this is also exactly the same as "GD" however F0 is linearly increased or decreased by 10% for male and female speakers, respectively.

In Figure 6.5, it is clear that for both "Average" and "GD" when we change F0 we obtain better performance. For example, comparing "Average" and "Average-F0" the EER for NSF3 against SF3 has improved significantly. Therefore, we can conclude that changing F0 even linearly can help. In addition, comparing "Average" and "GD" systems, in all cases, we obtain better performance with "GD" (except, comparing EERs of NSF3 and SF3 that there is not significant improvement). In "GD", we only use cross-gender models; therefore we expect that, as target speakers are more different from the source speaker, the new speaker will be more distinct. The "GD-F0" approximately outperforms the other three systems. Table 6.2 summarizes the average EERs captured by each of the four systems for newly created speakers (NSF1, NSF2, NSF3, NSM1, NSM2). These results also confirm that transforming F0 and using gender information help decrease the EER.

Figure 6.6 uses t-Distributed Stochastic Neighbor Embedding (t-SNE) [89] to map the 600-D i-Vector representation of test data into 2-D space. This figure also confirms that the 10 original speakers of the database and 5 new generated speakers (de-identified speakers) are almost distinct.

### 6.4.2 Subjective Test

For subjective evaluation, we conducted MOS-naturalness test for "GD-F0" speaker de-identification system. We did an informal subjective test at CRSS and obtained 2.8 for pooled utterances of all new generated speakers.

In addition to the MOS-naturalness, an additional subjective test can be designed. We also ask participants "if they can distinguish the new speaker from each available speaker in the database or not". We did an informal subjective test at CRSS and we obtained 100% accuracy for "GD-F0". One of the reasons is that we changed F0, and mapped the source speaker from male to female and vice versa.

## 6.5 Summary

This chapter presented a new solution for the speaker de-identification task. For a given speech signal of a speaker, first, spectral and excitation features are extracted. The spectral features are mapped non-linearly with a novel convolutional encoder-decoder based voice conversion system; and F0 is converted linearly. Transformed features are finally combined together and synthesized to generate the de-identified speech signal. The experiments were carried out on VCC-2016 database and evaluated subjectively and objectively with i-Vector/PLDA speaker recognition system. Each source speaker in the database was mapped to a new speaker; for the best proposed system (i.e., "GD-F0") the EER varies between 1.55%-2.682%, and 2.8 was achieved for the subjective MOS-naturalness test. For similarity as well, new speakers were discriminated from the source speaker with 100% accuracy for "GD-F0" speaker de-identification system.

# CHAPTER 7

# SUMMARY AND CONCLUSIONS

Mismatch between data used to develop a system and data used at the application step is a major problem for data-driven learning based algorithms, in many applications such as speech and image processing. In addition, many of these algorithms benefit from larger amount of training data; therefore, it is valuable to develop solutions to incorporate data from different domains to boost the performance, while being invariant to the domain mismatch problem. Speaker recognition is shown to perform poorly under mismatch conditions. In this dissertation, we studied various methods and algorithms to address the mismatch problem. The mismatch we focused on during this study includes both channel or noise distortions as well as language mismatch. We introduced both supervised and unsupervised techniques to address the mismatch problem and provide a robust speaker recognition system. We evaluated our proposed solutions on different corpora which are focused on various challenges in order to validate the effectiveness of our methods. In addition, we introduced a novel algorithm for speaker de-identification task; which protects the privacy of speakers as well as it can be used as an augmentation method. Here, we summarize the key dissertation contributions and the results, and also highlight several areas for future research.

## 7.1 Key Dissertation Contributions

We presented approaches based on machine learning and deep learning techniques to address the mismatch conditions in speaker recognition. We targeted both very general and common mismatch types, such as channel and noise mismatch, while addressing more challenging scenarios as well including language mismatch. Our solutions are studied separately and in combination with each other to comprehensively evaluate the effectiveness of our proposed methods. In addition, we proposed a new voice conversion based method for speaker de-identification, that they all are highlighted in continue.

### 7.1.1 Speaker Recognition

- **Generalized discriminant analysis for speaker recognition (Chapter 3):** In state-of-the-art framework for speaker recognition systems, linear discriminant analysis (LDA) is usually applied on extracted speaker embeddings to: (i) reduce the dimension, (ii) map the speaker embeddings into a new subspace, where speaker classes are more discriminant from each other. LDA assumes speaker classes have Gaussian distribution and they are linearly separable. However, channel distortions or added noise can make these assumptions to be not valid. Generally, the linear assumption is simplifying and might not be valid even if data are not distorted. Providing a more discriminant speaker embeddings to the score calculation step is also makes the scoring and decision making steps easier as well. Therefore, we introduced non-linear discriminant analysis with incorporating kernel functions. From another view point, when speaker utterances have variable lengths, there are some uncertainty included in the extraction of speaker embeddings, and can lead to a mismatch problem. Therefore, robust dimension reduction and discriminant analysis approach can address the variabilities that occur in the data. We evaluated the effectiveness of our proposed solution on NIST SRE-10 task, which focuses on channel and noise mismatch conditions. In addition, we evaluated the performance on long and short utterances as well. In our experiments, it is shown that in terms of equal error rate (EER) and minimum of detection cost function, GDA not only improves performance for regular test utterances, but it is also useful for short duration test segments. The relative improvement in EER is 20% for the cosine kernel employed with GDA processing.

- **Support vector discriminant analysis both as a dimension reduction approach (Chapter 3) and domain adaptation technique (Chapter 4,5):** The linear discriminant analysis focuses on mean centroid of speaker classes to calculate

the between class covariance matrix. Therefore, it does not consider the structure of speaker classes to calculate the between class covariance matrix. Our proposed solution here, called SVDA, incorporates the boundary structure of speaker classes. It uses support vectors of speaker classes to calculate the scatters in the objective function. Therefore, from one aspect SVDA is a general approach which addresses the variabilities included in speaker classes. We evaluated the performance of that on NIST SRE-10 task for noise and channel distortions and The relative improvement in terms of EER and minDCF with SVDA are about 32% and 9%, respectively.

However, beyond discriminant analysis, SVDA is a domain adaptation technique as well. While training the support vector machine (SVM) to find the support vectors (in a multi-class classification task), two different strategies can be used: (1) one-versus-one, (2) one-versus-rest. Therefore, in SVDA as a domain adaptation technique, unlabeled in-domain data or poorly labeled in-domain data can be integrated effectively within the SVDA framework. Adding knowledge to the discriminator that how data in target-domain is distributed, helps to provide a better transformation matrix. We can either (i) add all the in-domain data to the rest class in one-versus-rest strategy, or (ii) estimate pseudo labels for in-domain data with clustering methods and use them with one-versus-one strategy, or (iii) count them all to belong to one speaker classes and used them with the one-versus-one strategy. We evaluated the effectiveness of our proposed SVDA approach for domain adaptation purpose on both NIST SRE-16 and NIST SRE-18 tasks, with three different speaker embeddings, i-Vector, t-Vector and x-Vector; considering different strategies to include data from the domain of interest. A comprehensive evaluation for all the configurations is provided in Chapter 5. Results confirmed that SVDA can improve speaker recognition for SRE-16 and SRE-18 by +15% and +8% in terms on min-Cprimary respectively; and in terms of EER +14% and +16% respectively, with i-Vector speaker embeddings.

- **Domain normalized/adapted speaker embeddings: a-Vectors (Chapter 4):** A deep learning based technique proposed here to perform domain adaptation within speaker recognition framework. As speaker recognition systems perform poorly for domain mismatch conditions, it means that speaker embeddings not only contain speaker-related information, but also include other directions as well which represent the domain or generally speaker-irrelevant directions. Here, we proposed to append novel auxiliary features to the speaker embeddings for the purpose of compensating the domain-related directions (or speaker-irrelevant information). Therefore, the proposed solution is based on concatenation of domain-adapted auxiliary features and the speaker embeddings to normalize for specific language-dependent directions. In more detail, we modeled a simplified version of the inception-v4 network to map i-Vectors to these new auxiliary features. The concatenated feature vector of i-Vectors and auxiliary features is called a-Vectors. In addition, we also proposed a new loss function called domain-adapted triplet loss function. The proposed loss function is a variation of triplet loss function introduced before for image and speaker recognition/verification. However, here the loss function provides a transformation to map the out-domain speaker embeddings to those in the target-domain. Therefore, instead of focusing on the distance between speaker classes, the distance between out-domain and in-domain data is taken into consideration. This method as well introduces a new unsupervised adaptation technique. Evaluations based on the NIST SRE-16 confirm the effectiveness of the proposed technique. In terms of minimum Cprimary cost, a-Vector outperforms the i-Vector consistently. Moreover, comparing to previous single systems introduced for SRE-16, we achieved 8.5%-18% improvements in terms of equal error rate.

- **Comprehensive evaluation of domain mismatch compensation solutions at the back-end level (Chapter 5):** In chapter 5, we introduced a supervised and

an unsupervised PLDA domain adaptation techniques; a clustering algorithm to assign pseudo labels to the unlabeled data; LDA adaptation; and designed comprehensive experiments to evaluate various methods for domain adaptation separately and in combination with each other. In addition, we summarized the performance for both DEV and EVAL sets; to emphasize how our proposed solutions address the generalization issue. We incorporated the domain adaptation techniques with i-Vector, t-Vector and x-Vector speaker embeddings to verify if they perform well independent of the the front-end processing in a speaker recognition framework. The results confirm SVDA consistently provides a boost in the performance. This simple method addresses the domain adaptation very effectively. In addition, experiments show that the unsupervised adapted PLDA outperforms the supervised PLDA domain adaptation. Since, data provided from the domain of interest is very limited and they are unlabeled, as we expected they can provide more gain when they are included in an unsupervised technique.

### 7.1.2 Speaker De-Identification

- **Convolution encoder-decoder neural network for the voice conversion task (Chapter 6):** A new convolution encoder-decoder based voice conversion system designed in this dissertation. Voice conversion is studied widely, and different methods are proposed to convert voice from a source speaker to a target speaker. Our proposed CNN-based mapping has multiple advantages over other voice mapping methods which include: (1) Compared to approaches that use delta features to capture time-dependencies, our convolutional encoder-decoder network is able to automatically capture the dependencies between adjacent acoustic feature frames without including dynamic features. As a result, our method does not need a parameter generation algorithm which is prone to the over-smooting problem. (2) Compared to LSTM-RNN

approaches, our network is faster and easier to train. Additionally, due to the recurrent nature of LSTM-RNN, it cannot fully exploit the GPU capabilities.

- **Mapping source speakers to new unique identities for speaker de- identification task (Chapter 6):** In chapter 6, we proposed a new approach for speaker de-identification, which effectively conceals the voice characteristics of a given speaker as well as provides de-identified samples with high quality and intelligibility. In other words, our proposed de-identification solution optimizes both objective and subjective measurements. Here, in contrast to other related works (to the best of our knowledge, they all used an already available voice transformation method), we proposed a new convolution encoder-decoder based voice mapping and incorporated that into our speaker de-identification system. The convolution neural network (CNN) voice mapping architecture is specifically designed to consider details of the database and has the ability to suppress the errors and noises that might occur during the preparation of data for the voice mapping step. Finally, the speaker de-identification system employs the voice mapping module to transform the voice characteristics of a given speaker to all target speakers in the database. Average or gender-dependent average of mapped voices leads to the de-identified voice. In addition, each source speaker in our dataset is mapped to a new identity; which is not included in our dataset and it is different from other de-identified voices (achieved from different source speakers). We evaluated the proposed architecture with an i-vector/probabilistic linear discriminant analysis (PLDA) speaker recognizer. In addition, the de-identification approach proposed here is designed to achieve good performance evaluated by both human listeners and machines while preserving quality and naturalness. In our experiments, the EER varies between 1.55%-2.682%, and 2.8 was achieved for the subjective MOS-naturalness test. For similarity as well, new speakers were discriminated from the source speaker with 100% accuracy with our best performing speaker de-identification system.

## 7.2 Future Work

In this dissertation, we proposed a number of domain adaptation solutions for speaker recognition to decrease the vulnerability of the system towards the mismatch between train and enrollment/test data. However, there is still a gap to reduce the error rate further. Here, we highlight several related directions for future research inspired by thesis contributions already reported in this dissertation.

- **Generalized SVDA with incorporating kernel functions:** Generalized discriminant analysis is shown to outperform the linear discriminant analysis in this dissertation for speaker recognition; since, it relaxes the linear separable assumption about speaker classes. Likewise, SVDA can be extended to a non-linear version with incorporating the kernel functions. Since, SVDA both provides a robust dimension reduction approach as well as domain adaptation technique; we suggest extending that to the non-linear SVDA; and it can boost the performance on speaker recognition systems under mismatch conditions further.

- **End-to-end adversarial domain adaptation:** Adversarial domain adaptation is shown to be effective in image recognition task. In addition, in chapter 4, we proposed a-Vectors which confirmed that providing auxiliary features from target-domain for the purpose of domain adaptation is an effective approach. Therefore, we suggest training a new speaker embedding extarction framework while compensating for the domain variabilities with adversarial training can provide more robust speaker embeddings.

- **Speaker de-identification trained on unparalleled data:** In development of our speaker de-identification system, we used a parallel dataset to train our voice conversion model. Parallel dataset means for every source utterance there is a sample from the target speaker that utters the same sentence. Therefore, for each mapping the linguistic

information is preserved with providing the same utterance from a source speaker and a target speaker. However, there are more unparalleled datasets than the parallel ones. We suggest training the speaker de-identification algorithm using unparalleled data, which provides more robust performance, and introducing more data helps with increasing the quality and intelligibility of the de-identified samples as well.

## REFERENCES


[1] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning.," in *AAAI*, vol. 4, p. 12, 2017.

[2] C. Zhang, F. Bahmaninezhad, S. Ranjan, H. Dubey, W. Xia, and J. H. L. Hansen, "UTD-CRSS systems for 2018 NIST speaker recognition evaluation," *IEEE ICASSP, 2019*, 2019.

[3] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. Greenberg, E. S. Reynolds, L. Mason, and J. Hernandez-Cordero, "The 2016 NIST speaker recognition evaluation," *ISCA INTERSPEECH*, pp. pp. 1353–1357.

[4] J. H. L. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[5] F. Bahmaninezhad and J. H. L. Hansen, "Generalized discriminant analysis (GDA) for improved i-vector based speaker recognition.," in *ISCA INTERSPEECH*, pp. 3643–3647, 2016.

[6] F. Bahmaninezhad and J. H. L. Hansen, "i-vector/PLDA speaker recognition using support vectors with discriminant analysis," in *IEEE ICASSP*, pp. 5410–5414, 2017.

[7] C. Zhang, F. Bahmaninezhad, S. Ranjan, C. Yu, N. Shokouhi, and J. H. L. Hansen, "UTD-CRSS systems for 2016 NIST speaker recognition evaluation," *ISCA INTERSPEECH*, pp. pp. 1343–1347.

[8] F. Bahmaninezhad and J. H. Hansen, "Compensation for domain mismatch in text-independent speaker recognition.," in *Interspeech*, pp. 1071–1075, 2018.

[9] F. Bahmaninezhad, C. Zhang, and J. H. Hansen, "Convolutional neural network based speaker de-identification.," in *Odyssey*, pp. 255–260, 2018.

[10] 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, 2000.

[11] C. Zhang, K. Koishida, and J. H. L. Hansen, "Text-independent speaker verification based on triplet convolutional neural network embeddings," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 26, no. 9, pp. 1633–1644, 2018.

[12] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," *IEEE ICASSP, 2018*.

[13] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[14] 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.

[15] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*, pp. 531–542, Springer, 2006.

[16] A. Misra and J. H. L. Hansen, "Modelling and compensation for language mismatch in speaker verification," *Speech Communication*, vol. 96, pp. 58–66, 2018.

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

[18] A. Misra and J. H. L. Hansen, "Spoken language mismatch in speaker verification: An investigation with NIST-SRE and CRSS Bi-Ling corpora," in *IEEE SLT: Spoken Language Technology Workshop*, pp. 372–377, 2014.

[19] S. H. Shum, D. A. Reynolds, D. Garcia-Romero, and A. McCree, "Unsupervised clustering approaches for domain adaptation in speaker recognition systems," 2014.

[20] S. Shon, S. Mun, W. Kim, and H. Ko, "Autoencoder based domain adaptation for speaker recognition under insufficient channel information," *arXiv preprint arXiv:1708.01227*, 2017.

[21] V. C. Colibro, Daniele, E. Dalmasso, K. Farrell, C. S. Karvitsky, Gennady, and P. Laface, "Nuance-politecnico di torino's 2016 NIST speaker recognition evaluation system," *ISCA INTERSPEECH*, pp. 1338–1342, 2017.

[22] O. Plchot, P. Matejka, A. Silnova, O. Novotnỳ, M. Diez, J. Rohdin, O. Glembek, N. Brümmer, A. Swart, J. Jorrın-Prieto, *et al.*, "Analysis and description of ABC submission to NIST SRE 2016," *ISCA INTERSPEECH*, pp. 1348–1352, 2017.

[23] B. J. Borgstrom, D. A. Reynolds, E. Singer, and O. Sadjadi, "Improving the effectiveness of speaker verification domain adaptation with inadequate in-domain data," tech. rep., MIT Lincoln Laboratory Lexington United States, 2017.

[24] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," *ISCA INTERSPEECH*, pp. 999–1003, 2017.

[25] P. A. Torres-Carrasquillo, F. Richardson, S. Nercessian, D. Sturim, W. Campbell, Y. Gwon, S. Vattam, N. Dehak, H. Mallidi, P. S. Nidadavolu, *et al.*, "The MIT-LL, JHU and LRDE NIST 2016 speaker recognition evaluation system," *ISCA INTERSPEECH*, pp. 1333–1337, 2017.

[26] K. A. Lee, V. Hautamäki, T. Kinnunen, A. Larcher, C. Zhang, A. Nautsch, T. Stafylakis, G. Liu, M. Rouvier, W. Rao, *et al.*, "The I4U Mega fusion and collaboration for NIST speaker recognition evaluation 2016," *ISCA INTERSPEECH*, pp. 1328–1332, 2017.

[27] S. Madikeri, S. Dey, M. Ferras, P. Motlicek, and I. Himawan, "Idiap submission to the NIST SRE 2016 speaker recognition evaluation," tech. rep., Idiap, 2016.

[28] M. Rouvier, P.-M. Bousquet, M. Ajili, W. B. Kheder, D. Matrouf, and J.-F. Bonastre, "LIA system description for NIST SRE 2016," *arXiv preprint arXiv:1612.05168*, 2016.

[29] K. A. Lee, V. Hautamaki, T. Kinnunen, H. Yamamoto, K. Okabe, V. Vestman, J. Huang, G. Ding, H. Sun, A. Larcher, *et al.*, "I4u submission to NIST SRE 2018: Leveraging from a decade of shared experiences," *arXiv preprint arXiv:1904.07386*, 2019.

[30] G. Bhattacharya, J. Monteiro, J. Alam, and P. Kenny, "Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification," in *IEEE ICASSP*, pp. 6226–6230, IEEE, 2019.

[31] A. E. Bulut, Q. Zhang, C. Zhang, F. Bahmaninezhad, and J. H. Hansen, "UTD-CRSS submission for MGB-3 Arabic dialect identification: Front-end and back-end advancements on broadcast speech," in *IEEE ASRU Workshop*, pp. 360–367, IEEE, 2017.

[32] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 92–97, IEEE, 2015.

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

[34] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

[35] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *ISCA INTERSPEECH*, 2017.

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

[37] N. Brümmer and E. De Villiers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new dcf," *arXiv preprint arXiv:1304.2865*, 2013.

[38] D. E. Sturim and D. A. Reynolds, "Speaker adaptive cohort selection for tnorm in text-independent speaker verification," in *IEEE ICASSP*, vol. 1, pp. I–741, 2005.

[39] "The NIST year 2010 speaker recognition evaluation plan." `https://www.nist.gov/sites/default/files/documents/itl/iad/mig/NIST_SRE10_evalplan-r6.pdf`, 2010.

[40] "NIST 2016 speaker recognition evaluation plan." `https://www.nist.gov/sites/default/files/documents/itl/iad/mig/SRE16_Eval_Plan_V1-0.pdf`, 2016.

[41] "NIST 2018 speaker recognition evaluation plan." `https://www.nist.gov/sites/default/files/documents/2018/08/17/sre18_eval_plan_2018-05-31_v6.pdf`, 2018.

[42] B. Scholkopft and K.-R. Mullert, "Fisher discriminant analysis with kernels," *Neural Networks for Signal Processing IX*, vol. 1, no. 1, p. 1, 1999.

[43] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, vol. 12, no. 10, pp. 2385–2404, 2000.

[44] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank hmms for improved speech recognition," *Speech Communication*, vol. 26, no. 4, pp. 283–297, 1998.

[45] T. Hastie and R. Tibshirani, "Discriminant analysis by gaussian mixtures," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 155–176, 1996.

[46] S. Sadjadi, J. W. Pelecanos, and W. Zhu, "Nearest neighbor discriminant analysis for robust speaker recognition.," *ISCA INTERSPEECH*, pp. 1860–1864, 2014.

[47] S. Sadjadi, J. Pelecanos, and S. Ganapathy, "The ibm speaker recognition system: Recent advances and error analysis," *ISCA INTERSPEECH*, 2016.

[48] M. McLaren and D. Van Leeuwen, "Source-normalized lda for robust speaker recognition using i-vectors from multiple speech sources," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 755–766, 2011.

[49] A. Kanagasundaram, D. Dean, R. Vogt, M. McLaren, S. Sridharan, and M. Mason, "Weighted lda techniques for i-vector based speaker verification," in *IEEE ICASSP*, pp. 4781–4784, IEEE, 2012.

[50] S. Gu, Y. Tan, and X. He, "Discriminant analysis via support vectors," *Neurocomputing*, vol. 73, no. 10, pp. 1669–1675, 2010.

[51] C. Yu, G. Liu, S. Hahm, and J. H. L. Hansen, "Uncertainty propagation in front end factor analysis for noise robust speaker recognition," in *IEEE ICASSP*, pp. 4017–4021, 2014.

[52] T. Hasan, R. Saeidi, J. H. L. Hansen, and D. A. van Leeuwen, "Duration mismatch compensation for i-vector based speaker recognition systems," in *IEEE ICASSP*, pp. 7663–7667, 2013.

[53] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *IEEE ICASSP*, pp. 7649–7653, 2013.

[54] G. Liu and J. H. L. Hansen, "An investigation into back-end advancements for speaker recognition in multi-session and noisy enrollment scenarios," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1978–1992, 2014.

[55] C. Chang and C. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[56] F. Bahmaninezhad, J. Wu, R. Gu, S.-X. Zhang, Y. Xu, M. Yu, and D. Yu, "A comprehensive study of speech separation: spectrogram vs waveform separation," *arXiv preprint arXiv:1905.07497*, 2019.

[57] A. Senior and I. Lopez-Moreno, "Improving DNN speaker independence with i-vector inputs," in *IEEE ICASSP*, pp. 225–229, 2014.

[58] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," *ISCA INTERSPEECH*.

[59] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[60] D. Garcia-Romero and A. McCree, "Supervised domain adaptation for i-vector based speaker recognition," in *IEEE ICASSP*, pp. 4047–4051, IEEE, 2014.

[61] S. O. Sadjadi and J. H. Hansen, "Unsupervised speech activity detection using voicing measures and perceptual spectral flux," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 197–200, 2013.

[62] A. Ziaei, L. Kaushik, A. Sangwan, J. H. Hansen, and D. W. Oard, "Speech activity detection for nasa apollo space missions: Challenges and solutions," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[63] D. Snyder, G. Chen, and D. Povey, "MUSAN: a music, speech, and noise corpus," 2015.

[64] S. H. K. Parthasarathi, M. M. Doss, H. Bourlard, and D. Gatica-Perez, "Evaluating the robustness of privacy-sensitive audio features for speech detection in personal audio log scenarios," *IEEE ICASSP*, pp. 4474–4477, 2010.

[65] S. H. K. Parthasarathi, H. Bourlard, and D. Gatica-Perez, "Wordless sounds: robust speaker diarization using privacy-preserving audio representations," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 1, pp. 85–98, 2013.

[66] K. Hashimoto, J. Yamagishi, and I. Echizen, "Privacy-preserving sound to degrade automatic speaker verification performance," *IEEE ICASSP*, pp. 5500–5504, 2016.

[67] T. Justin, V. Štruc, S. Dobrišek, B. Vesnicer, I. Ipšić, and F. Mihelič, "Speaker de-identification using diphone recognition and speech synthesis," *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, vol. 4, pp. 1–7, 2015.

[68] Q. Jin, A. R. Toth, T. Schultz, and A. W. Black, "Speaker de-identification via voice transformation," *IEEE ASRU*, pp. 529–533, 2009.

[69] C. Magariños, P. Lopez-Otero, L. Docio-Fernandez, E. R. Banga, C. Garcia-Mateo, and D. Erro, "Piecewise linear definition of transformation functions for speaker de-identification," *Sensing, Processing and Learning for Intelligent Machines (SPLINE), 2016 First International Workshop on*, pp. 1–5, 2016.

[70] M. Pobar and I. Ipsic, "Online speaker de-identification using voice transformation," *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*, pp. 1264–1267, 2014.

[71] T. Toda, L. Chen, D. Saito, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, "The voice conversion challenge 2016," *ISCA INTERSPEECH*, 2016.

[72] M. Wester, Z. Wu, and J. Yamagishi, "Analysis of the voice conversion challenge 2016 evaluation results," *ISCA INTERSPEECH*, 2016.

[73] Y. LeCun, Y. Bengio, *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[74] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[75] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," *IEEE ICASSP*, vol. 1, pp. 285–288, 1998.

[76] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[77] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," *IEEE ICASSP*, pp. 4869–4873, 2015.

[78] S. H. Mohammadi and A. Kain, "A voice conversion mapping function based on a stacked joint-autoencoder," *ISCA INTERSPEECH*, pp. 1647–1651, 2016.

[79] L.-H. Chen, L.-J. Liu, Z.-H. Ling, Y. Jiang, and L.-R. Dai, "The USTC system for voice conversion challenge 2016: Neural network based approaches for spectrum, aperiodicity and f 0 conversion," *ISCA INTERSPEECH*, pp. 1642–1646, 2016.

[80] Y.-C. Wu, H.-T. Hwang, C.-C. Hsu, Y. Tsao, and H.-M. Wang, "Locally linear embedding for exemplar-based spectral conversion," *ISCA INTERSPEECH*, 2016.

[81] K. Kobayashi, S. Takamichi, S. Nakamura, and T. Toda, "The NU-NAIST voice conversion system for the voice conversion challenge 2016," *ISCA INTERSPEECH*, pp. 1667–1671, 2016.

[82] S. Khorram, Z. Aldeneh, D. Dimitriadis, M. McInnis, and E. M. Provost, "Capturing long-term temporal dependencies with convolutional networks for continuous emotion recognition," *ISCA INTERSPEECH*, 2017.

[83] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[84] T. Nakashika, T. Takiguchi, and Y. Minami, "Non-parallel training in voice conversion using an adaptive restricted boltzmann machine," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2032–2045, 2016.

[85] J. Kominek and A. W. Black, "The CMU arctic speech databases," *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[86] H. Kawahara, I. Masuda-Katsuse, and A. De Cheveigne, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, no. 3, pp. 187–207, 1999.

[87] H. Banno, H. Hata, M. Morise, T. Takahashi, T. Irino, and H. Kawahara, "Implementation of realtime straight speech manipulation system: Report on its first implementation," *Acoustical science and technology*, vol. 28, no. 3, pp. 140–146, 2007.

[88] S. Khorram, H. Sameti, F. Bahmaninezhad, S. King, and T. Drugman, "Context-dependent acoustic modeling based on hidden maximum entropy model for statistical parametric speech synthesis," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2014, no. 1, p. 12, 2014.

[89] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

## BIOGRAPHICAL SKETCH

Fahimeh Bahmaninezhad received her BSc degree in Computer Engineering-Software Engineering from Shadid Chamran University of Ahvaz in 2010 and MSc degree in Computer Engineering-Artificial Intelligence from Sharif University of Technology in 2012. In January 2015, she joined the Center for Robust Speech Systems (CRSS) at The University of Texas at Dallas (UTD) as a PhD student and research assistant under the supervision of professor John H.L. Hansen. In summer 2018, she interned with Microsoft (Redmond, WA) as a research intern at the Applied Science group, focusing on single-channel speech separation. In winter 2019, she joined Tencent AI Lab (Bellevue, WA) as a research intern, focusing on multi-channel speech separation and speech extraction. Her research interests include machine learning in signal, speech and language processing, speech and speaker recognition and deep learning.

**CURRICULUM VITAE**

# Fahimeh Bahmaninezhad

November 2019

## Contact Information:

Department of Electrical Engineering
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080-3021, U.S.A.

Email: fahimeh.bahmaninezhad@utdallas.edu

## Educational History:

B.S., Computer Engineering, Shahid Chamran University of Ahvaz, 2010
M.S., Computer Engineering, Sharif University of Technology, 2012
Ph.D., Electrical Engineering, the University of Texas at Dallas, 2019

*Advancements in domain adaptation for speaker recognition and effective speaker de- identification*
Ph.D. Dissertation
Electrical Engineering Department,, Cornell University
Advisors: Dr. John H.L. Hansen

*Speaker adaptation in HMM-based Persian speech synthesis*
Master's Thesis
Computer Engineering Department, Sharif Univeristy of Technology
Advisor: Dr. Hossein Sameti

## Employment History:

Research Intern, Tencent, January 2019 – May 2019
Research Intern, Microsoft, May 2018 – August 2019

## RESEARCH PUBLICATIONS:

- **F. Bahmaninezhad**, C. Zhang, and J. H.L. Hansen, "An Investigation of Domain Adaptation in Speaker Embedding Space for Speaker Recognition", (submitted to) Speech Communication, 2019.
- **F. Bahmaninezhad**, S.X. Zhang, Y. Xu, M. Yu, John H.L. Hansen, and D. Yu, "A Unified Framework for Speech Separation", (submitted to) IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 2019.

- **F. Bahmaninezhad**, J. Wu, R. Gu, S.X. Zhang, Y. Xu, M. Yu, and D. Yu, "A Comprehensive Study of Speech Separation: Spectrogram vs Waveform Separation", ISCA INTERSPEECH, 2019.

- K.A. Lee, et al., "I4U Submission to NIST SRE 2018: Leveraging from a Decade of Shared Experiences", ISCA INTERSPEECH, 2019.

- C. Zhang, **F. Bahmaninezhad**, S. Ranjan, H. Dubey, W. Xia and J. H.L. Hansen, "UTD-CRSS Systems for 2018 NIST Speaker Recognition Evaluation", IEEE ICASSP, 2019.

- **F. Bahmaninezhad**, and J. H.L. Hansen, "Compensation for Domain Mismatch in Text-Independent Speaker Recognition", ISCA INTERSPEECH, 2018.

- **F. Bahmaninezhad**, C. Zhang, and J. H.L. Hansen, "Convolutional Neural Network Based Speaker De-Identification", ISCA ODYSSEY, 2018.

- A.E. Bulut, Q. Zhang, C. Zhang, **F. Bahmaninezhad**, and J. H.L. Hansen, "UTD-CRSS Submission for MGB-3 Arabic Dialect Identification: Front-End and Back-End Advancements on Broadcast Speech", IEEE ASRU, 2017.

- C. Zhang, **F. Bahmaninezhad**, S. Ranjan, C. Yu, N. Shokouhi, and J. H.L. Hansen, "UTD-CRSS Systems for 2016 NIST Speaker Recognition Evaluation", ISCA INTERSPEECH, 2017.

- K.A. Lee, et al., "The I4U Mega Fusion and Collaboration for NIST Speaker Recognition Evaluation 2016", ISCA INTERSPEECH, 2017.

- **F. Bahmaninezhad**, and J. H.L. Hansen, "i-Vector/PLDA Speaker Recognition Using Support Vectors with Discriminant Analysis", IEEE ICASSP, 2017.

- C. Zhang, **F. Bahmaninezhad**, S. Ranjan, C. Yu, N. Shokouhi, and J. H.L. Hansen, "UTD-CRSS Systems for 2016 NIST Speaker Recognition Evaluation", NIST SRE Workshop, 2016.

- K.A. Lee, et al., "The I4U Submission to the 2016 NIST Speaker Recognition Evaluation", NIST SRE Workshop, 2016.

- **F. Bahmaninezhad**, and J. H.L. Hansen, "Generalized Discriminant Analysis (GDA) for Improved i-Vector Based Speaker Recognition", ISCA INTERSPEECH, 2016.

- S. Khorram, H. Sameti, **F. Bahmaninezhad**, S. King, and T. Drugman, "Context-Dependent Acoustic Modeling Based on Hidden Maximum Entropy Model for Statistical Parametric Speech Synthesis", EURASIP Journal on Audio, Speech, and Music Processing, 2014.

- S. Khorram, H. Sameti, and **F. Bahmaninezhad**, "Context-dependent Deterministic Plus Stochastic Model", International Conference on Signal Processing (ICSP), 2014.
- S. Khorram, **F. Bahmaninezhad**, and H. Sameti, "Speech Synthesis Based on Gaussian Conditional Random Fields", Artificial Intelligence and Signal Processing (AISP), 2013.
- **F. Bahmaninezhad**, S. Khorram, and H. Sameti, "Average Voice Modeling Based on Unbiased Decision Trees", Advances in Nonlinear Speech Processing (NOLISP), 2013.
- **F. Bahmaninezhad**, H. Sameti, and S. Khorram, "HMM-based Persian Speech Synthesis Using Limited Adaptation Data", IEEE International Conference on Signal Processing (ICSP), 2012.