Odyssey 2012
The Speaker and Language Recognition Workshop
25-28 June 2012, Singapore

Dataset Shift in PLDA based Speaker Verification

Carlos Vaquero

Agnitio, Spain
cvaquero@agnitio.es

Abstract

Dataset shift is a problem widely studied in the field of speaker recognition. Among the different types of dataset shift, covariate shift is the most common one in real scenarios. Traditional solutions for the problem of covariate shift have been developed in the context of channel and session variability, and make use of large datasets to train models for channel/session compensation. However, in real applications, it is not always possible to obtain a large matched dataset to train these techniques.

This work analyzes the stages of an i-vector system that are more vulnerable to covariate shift, and proposes different techniques to mitigate this effect. The proposed techniques operate under the assumption that little matched data is available for development. These techniques are evaluated in a scenario where covariate shift is simulated introducing language shift. Among the proposed techniques, the most promising one is the i-vector adaptation based on the mean centering and length normalization technique.

However, the proposed techniques are not enough to reduce the wide gap in the accuracy that appears in presence of covariate shift.

1. Introduction

In real scenarios, it is usual to find that a state-of-the-art speaker verification system does not obtain an accuracy significantly higher than a simpler solution as a MAP speaker verification system [1]. This is due to the existing mismatch between the data and conditions considered during the development of the system, and those the system faces in a real scenario. The existence of this mismatch and techniques to mitigate the effect it produces have been widely studied in the field of pattern recognition. Recently, this problem has adopted the name of dataset shift [2].

In a classification problem, dataset shift appears when the joint distributions of the inputs and outputs are different for training and testing. The concept of dataset shift has been introduced in the field of pattern recognition since it is a problem common to most areas of the field [3]. In effect, there is not a standard term to refer to this effect: different technologies based on pattern recognition (including speaker verification) refer to this effect using several and different manners. This reduces the visibility of the approaches developed for a given technology, that usually can help in other technologies.

The problem of dataset shift is well known in speaker verification. Traditionally, the research related to dataset shift in speaker verification has been mostly focused in the compensation of channel and session variability. Some of the most recent techniques developed for this purpose include Joint Factor Analysis (JFA) [4], LDA and Within Class Covariance Normalization (WCCN) on i-vectors [5], Probabilistic LDA (PLDA) [6], length normalization for i-vectors [7], and the development of a fully Bayesian approach for PLDA based speaker verification [8]. Most of this techniques require a large amount of labeled development data to operate correctly. However, in real applications, it is not usual to find large datasets matched to the operation conditions. If they exist, they may contain labeling errors, and most users have serious problems to provide these datasets because of confidentiality or privacy reasons. In real cases, it is usual to obtain a small amount of data that is representative of the scenario the system will face, but it is not enough to develop a complete system.

Some of these techniques provide flexibility to incorporate new data. Subspace based techniques as JFA or i-vector techniques enables us to train a small variability subspace that can be stacked with the subspace matrices previously trained with a larger mismatched dataset [9], [10].

However, only a few of the techniques previously mentioned are suitable to deal with dataset shift when the dataset available is small. Among them, only the works presented in [7], [8] enables us to use a small matched dataset directly to mitigate the effect of dataset shift. The work presented in [8] can add a small development dataset matched to the evaluation conditions in the likelihood ratio calculations. The work in [7] set the basis to perform i-vector adaptation. Note that these two techniques operate only in the PLDA back-end. Currently there are no techniques to adapt other stages as the i-vector extractor, or even the PLDA model, given a small dataset matched to the operating conditions of the system.

This lack of flexibility of state-of-the-art techniques for dataset shift compensation makes difficult the deployment of these systems in unseen scenarios. The current architecture does not provide methods to adapt the different stages, so a system cannot learn easily from its environment, except for some normalization or calibration techniques.

This work analyzes the architecture of a state-of-the-art i-vector PLDA system, aiming at the identification of the stages that are critical for its operation in dataset shift conditions. An environment affected by dataset shift is proposed, simulated by language mismatch during development and testing. Several solutions are evaluated to mitigate this effect, considering that a small matched dataset is available. These solutions include stacking total variability matrices, i-vector adaptation using matched i-vector mean and the use of normalization techniques.

This paper is organized as follows: In section 2 we describe the i-vector PLDA system considered in this study and we analyze the most vulnerable stages to dataset shift. In section 3 we describe the techniques considered to mitigate the effect of dataset shift, while in section 4 we describe the evaluation setup considered to simulate dataset shift. In section 5 we evaluate the proposed techniques to summarize the main conclusions in section 6.

2. Dataset shift in i-vector PLDA speaker verification system

2.1. Dataset shift in speaker verification

Dataset shift is defined as the situation that appears when the training and testing joint distributions of the input and output variables are different. In a speaker verification problem, the inputs are sets of pairs (or groups) of feature vectors and the outputs can take two values, depending on which one of the two possible hypotheses every input pair fulfills: H_1, or the pair of feature vectors belong to the same speaker, and H_2 or the pair of feature vector belong to different speakers.

The different kinds of dataset shift that may appear can be classified into three groups [3]:

- Covariate shift: we refer to covariate shift when there are changes in the distribution of the input variables, but the conditional distribution of the output variables remain unchanged. This is the most common type of dataset shift in most pattern recognition technologies, including speaker verification. It appears when the features observed during the operation of the system differ significantly from those considered for development, but the relationship between the features and the decisions is still valid. For example, this is the type of dataset shift we face when the development and evaluation data are acquired in different scenarios (channel mismatch).
- Prior probability shift: it refers to changes in the distribution of the output variable when the relationship between inputs and outputs remains unaltered. This problem is also usual in speaker verification, and it involves calibration problems. In effect, if the prior probability for a given hypothesis is obtained from a development dataset, and the evaluation dataset has a different prior for the same hypothesis, the system will be miscalibrated. Fortunately, calibration methods enables us to plug any prior to recalibrate the system, so if the prior for the evaluation data can be estimated, this type of shift is not an issue.
- Concept shift: concept shift or concept drift appears when the relationship between inputs and outputs changes. This is the hardest type of dataset shift that has been studied in the literature. It is usually related to non-stationary problems, where the relationship between inputs and outputs evolve with time. Typical examples of concept shift are adversarial environments, such as email filtering. In this environments, concept shift appears since there is an adversary that tries to work around the concepts learned by the classifier. In the field of speaker verification, concept shift may appear in scenarios that are subject to spoofing attacks. We do not consider this type of dataset shift in this work.

In this work we focus on covariate shift, which is the most usual type of dataset shift that appears during the deployment of a speaker verification system. Prior probability shift must also be taken into account in real applications, but as explained before, calibration techniques can deal with this shift if a relatively small matched development dataset is provided.

2.2. Dataset shift in i-vector PLDA systems

The i-vector PLDA approach has become the state-of-the art for speaker recognition, for several reasons: First, it achieves, in the framework of NIST SRE, the best results. Second, the i-vector architecture enables to represent a single recording by a low dimensional manageable vector (i-vector). This i-vector is assumed to contain most of the relevant information in the recording (speaker and session). Finally, PLDA provides a probabilistic framework that enables the calculation of likelihoods for sets of i-vectors. Thus, it is possible to evaluate hypotheses and compare them providing results in the form of likelihood ratios, without any restriction of the model-test architecture present in traditional speaker verification systems.

However, the i-vector PLDA system provides little flexibility in order to adapt to or to take advantage of small matched development datasets in scenarios with covariate shift. The main problem of this system is that both the i-vector extractor and the PLDA are statistical models that depend completely on the development data, but they follow different probabilistic frameworks so there is no way to adapt both models jointly. Other approaches, as JFA, would allow some sort of adaptation since the development data is considered in a single probabilistic framework.

In fact, according to the notation used in Section 2.1 it is usual to consider that the inputs are the i-vectors and the outputs the decisions taken (target/non-target). However, the i-vector extractor is trained over development data, so the problem in presence of covariate shift is not only the fact the the i-vector distribution change. There is another problem that comes from the fact that the i-vector extractor may not be able to describe all the desired variability to compare speakers properly in a new scenario.

Figure 1 shows the stages that follow an i-vector PLDA system during its operation, assuming that there are two input sequences of acoustic observations to be compared. Following this figure, we analyze how the presence of covariate shift affects the operation of the system.

2.2.1. Universal Background Model

The Universal Background Model (UBM) is a model of the distribution of the acoustic observations. It defines the space where the system expects to find acoustic observations. In presence of dataset shift, it may happen that the UBM does not fit the acoustic observations. In this situation, we can expect a significant degradation in the accuracy of the system.

Assuming that a small dataset representative of the evaluation data is available, there are techniques as MAP that enables us to adapt the UBM to the data. However, adapting the UBM will require retraining all the background models (total variability subspace, PLDA...) again on the development data, so this solution is hardly ever possible in a real application.

2.2.2. Total Variability subspace

The total variability subspace defines the directions of maximum variability among the different sessions of the development datasets. These directions are obtained maximizing the Likelihood of a Factor Analysis model. Therefore, the development data is very well described considering only the total variability subspace, but it is not guaranteed that the evaluation data will be correctly described in presence of dataset shift. If the evaluation data varies in other directions of the space, we expect significant degradation.

As stated previously, there is not any work exploring techniques to adapt the total variability subspace given a small development dataset. The preferred solutions to include new development data are based on the estimation of a new total variability subspace and to consider that the subspace is the union

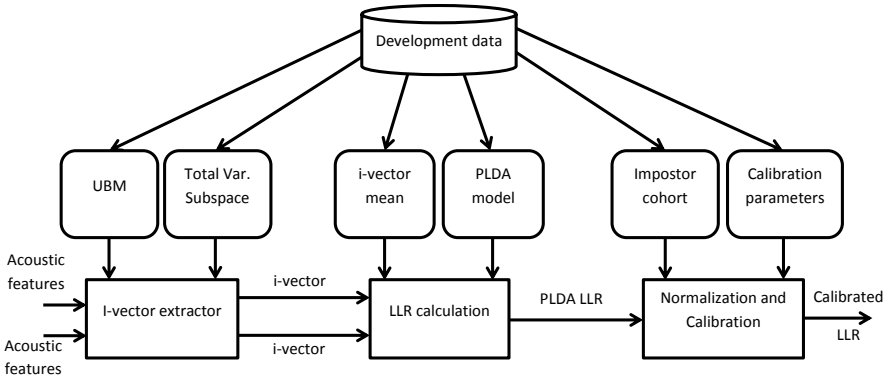

Figure 1: *Stages in the operation of an i-vector PLDA system.*

of this new subspace and the initial one. This is achieved concatenating or stacking total variability matrices.

The UBM and the total variability subspace are usually trained separately, but the JFA paradigm enables us to include the UBM parameters in the FA model. According to the model presented in [11], it is possible to train both the UBM and the i-vector extractor jointly, and thus it should be possible to adapt them jointly. However, this solution will require retraining the subsequent stages, including PLDA or calibration.

2.2.3. I-vector centering and length normalization

Recently, it has been shown that length normalization improves significantly the accuracy of the i-vector PLDA system [7]. Length normalization confines the i-vectors to the hypersphere of unit radius, so that they are closer to fulfill the Gaussian assumption of the PLDA model. Note that length normalization removes any scaling mismatch due to dataset shift. However, a severe mean shift will be critical for length normalization, confining the i-vectors obtained affected by dataset shift into a much smaller region in the space.

In this situation, adaptation can be possible given a small matched dataset. The i-vector mean obtained from the new data can be estimated and considered as center to perform i-vector centering and length normalization.

2.2.4. PLDA

The PLDA model seeks for the between speaker and within speaker covariance matrices in the space of i-vectors, in order to estimate the speaker and session component for every i-vector. The covariance matrices are estimated following a maximum likelihood criterion, on a development dataset. Therefore, if the evaluation data is affected by dataset shift, the covariance matrices may not explain the data properly, and thus the accuracy of the system will be degraded.

There is not any work on adaptation techniques for a PLDA model, but fully bayesian approaches are capable of including a small matched dataset in the computation of the PLDA Log-Likelihood Ratio (LLR) [8]. These techniques have a very high computational cost to be deployed in real systems, so they are not analyzed in this work. However, they are a starting point to face the dataset shift problem.

On the other hand, the problem of performing adaptation in the PLDA model is that the PLDA works under the assumption that the i-vectors contain all the desired speaker information. As we have stated previously, this may not be true. Therefore, the use of adaptation techniques or fully bayesian approaches in the PLDA stage may not solve the dataset shift problem if the i-vector extractor is not adapted.

2.2.5. Score normalization and calibration

A mismatch between the evaluation data and the development data considered for calibration or as normalization cohort will lead to an undesired operating point, due to a misalignment in the score distributions. Therefore, it is mandatory in most applications to obtain and to use a matched development dataset for calibration. Fortunately, calibration can be achieved with a relatively small dataset.

3. Techniques to compensate for dataset shift

We have described how the presence of covariate shift affects the background models considered in an i-vector PLDA speaker verification system. In the following sections we analyze some solutions to mitigate the effect of covariate shift. We assume that a large development dataset Ω_{dev} is available and has been considered to train all the background models. We also assume that a relatively small matched development dataset $\Omega_{matched}$ is provided. The size of this dataset is not enough to develop a complete system, but it is representative of the evaluation data.

3.1. Retraining the system

As first approach, we can pool all available data Ω_{dev} and $\Omega_{matched}$, in order to retrain the complete speaker verification system, from the UBM to the PLDA model.

3.2. Stacking total variability matrices

We can use the given dataset $\Omega_{matched}$ to estimate the total variability subspace that will appear during the evaluation of the system. This subspace will be represented by a total variability matrix $T_{matched}$, that can be stacked or concatenated to the initial matrix T_{dev} as follows:

$$T = [T_{dev} T_{matched}]. \quad (1)$$

Then, the i-vector mean and the PLDA model must be retrained. The technique of stacking or concatenating variability matrices was introduced during the NIST SRE 2008, to mitigate

the covariate shift due to the presence of microphone recordings [9]. It was firstly considered for the JFA framework, but it has also been tested in i-vector systems obtaining satisfactory results in [10]. In this last work, two approaches are considered in order to retrain the back end. The first approach pools all the available data including both the initial development data and the given matched dataset. The second one estimates the back end parameters separately for each dataset, and then obtains the final parameters weighting both models. Although the weighting solution gives better results for and i-vector system with LDA and WCCN, it is not straightforward to combine two different PLDA models by weighting. Therefore, we consider the pooling approach.

3.3. I-vector adaptation

The study recently presented in [7] shows that i-vector system can work significantly better if the i-vectors are centered on their mean and length normalization is performed. The center of the i-vector space is determined as the mean m_{dev} of the i-vectors extracted from Ω_{dev}. In presence of covariate shift, we can expect this center to be displaced. We propose the use of $m_{matched}$, computed over the i-vectors obtained from $\Omega_{matched}$ to perform i-vector centering and length normalization. This can be seen as an i-vector adaptation, where the new i-vector mean $m_{matched}$ will be subtracted from the evaluation i-vectors to move them to the same space where the i-vectors in Ω_{dev} are confined.

Note that this approach can be applied when both sides of a speaker verification trial suffers the same form of covariate shift. In a situation where only one side of a trial is affected by covariate shift, there is no reason to believe that the use of $m_{matched}$ will obtain better results than the use of m_{dev}. It is important to notice that both sides of the trial must be centered considering the same i-vector mean.

3.4. Score normalization and calibration

Normalization techniques aims at ensuring that the non-target score distributions are identical (usually Standard Normal) for all speakers and conditions. This is usually helpful for calibration purposes but also to improve the accuracy of a speaker verification system correcting the misalignment in score distributions that appear for different speakers and recordings. As normalization technique we consider S-norm, which is defined as follows:

$$s' = \frac{1}{\sqrt{2}} \left(\frac{s - \mu_1}{\sigma_1} + \frac{s - \mu_2}{\sigma_2} \right), \qquad (2)$$

where s is the score to normalize, μ_1 and σ_1 are the mean and standard deviation of the scores obtained when comparing one side of the trial to the impostor cohort, and μ_2 and σ_2 are the mean and standard deviation of the scores obtained when comparing the other side of the trial to same the impostor cohort. This normalization is suitable for PLDA systems since it keeps the symmetry of the scores.

Calibration is one of the most critical stages during the development of a speaker verification system, an it is also one of the stages more vulnerable to covariate shift. This stage is mandatory to ensure that the actual operating point matches the required one in a real scenario. When we apply any score calibration technique to the output scores of a speaker verification system we are assuming that the target and non-target score distributions for the output scores are identical to those observed during the development stage. This may not be true in an scenario with covariate shift.

The most straightforward solution is to calibrate the system considering $\Omega_{matched}$. To do so, $\Omega_{matched}$ must be diverse and large enough to build an evaluation containing several target and non-target trials, in order to estimate both target and non-target score distributions properly.

4. Experimental setup

In order to simulate a situation with covariate shift, we consider the NIST SRE data. We assume that the development data Ω_{dev} only contains English speech, but the language in the evaluation scenario is different. Concretely we consider three groups of languages for evaluation: Chinese, containing speech in Mandarin and Yue, Hindi-Urdu, containing speech in Hindi and Urdu, and Russian, containing speech in Russian. For every group of languages an evaluation is built from the short2-short3 condition of the NIST SRE 2008 evaluation. Development data for each language group is extracted from NIST SRE 2004, 2005 and 2006.

Note that we consider the situation where all the evaluation data suffers the same covariate shift (i.e. both sides of a trial are affected by covariate shift in the same form), we will not consider cross-language trials.

4.1. Development data

As development data we consider the NIST SRE 2004, 2005 and 2006 data (Mixer database), the Switchboard II and switchboard cellular databases, and Fisher data. Switchboard and Fisher data are completely in English, so this databases are only used as development data Ω_{dev} to build the UBM, i-vector extractor and PLDA model. Mixer database contains a wide variety of languages. We only consider the three mentioned groups of languages for $\Omega_{mached-lang}$ and English for Ω_{dev}. The number of recordings and speakers considered in from each dataset are shown in Table 1.

Table 1: *Development data.*

Language	sessions	speakers
English (Mixer)	15743	1411
English (Switchboard)	19961	1624
English (Fisher)	22329	13432
Chinese	1544	267
Hindi-Urdu	37	7
Russian	407	73

4.2. Evaluation data

To evaluate the proposed strategies to deal with covariate shift we consider the data extracted from the short2-short3 condition from the NIST SRE 2008. We build four separate evaluations, one for each group of languages and one for English. For each evaluation we consider all possible trials (all possible model-test pairs). The number of models, test segments, target and non-target trials for each evaluation are shown in Table 2.

4.3. System configuration

The speaker verification system considered in this study is a gender independent i-vector PLDA system that makes use of

Table 2: *Evaluation data.*

Language	models (spks)	test ses.	tar	non-tar
English	1260 (863)	1825	1948	2297552
Chinese	102 (70)	149	163	15035
Hindi-Urdu	73 (48)	83	74	5985
Russian	47 (30)	66	64	3038

a mixture of two gender dependent PLDA models, as the one described in [12]. We consider i-vectors with a dimensionality of 400. The PLDA model considers a full covariance matrix to model the session component and a low rank matrix of dimension 120 to span the speaker subspace in the i-vector space. The system performs i-vector centering and length normalization.

5. Experiments and Results

In this section we analyze and compare the techniques proposed to deal with covariate shift in the proposed scenario of language mismatch. Results are presented in terms of EER and minimum of the (unnormalized) Detection Cost Function (DCF) defined by NIST. We consider the *old* DCF, that assumes $c_{fa} = 1$, $c_m = 10$ and $P_{target} = 0.01$. The *new* DCF is not evaluated since in most conditions there are not enough trials to analyze this operating point. The actual value of the DCF is also considered when studying the calibration of the system.

5.1. Baseline

We evaluate the i-vector PLDA system described in [12] in situations of language mismatch. The system is completely trained on English data, including the calibration. The results obtained on every evaluation are shown in Table 3:

Table 3: *Baseline results.*

Language	EER	minDCF
English	1.43%	0.0060
Chinese	4.99%	0.0196
Hindi-Urdu	2.69%	0.0214
Russian	3.04%	0.0193

Note that the evaluation on English data obtain much higher accuracy than in any other language. Part of the difference can be due to the intrinsic difficulty of speaker detection in a given language, which may vary from language to language. Since there is not enough data available to train a complete system for each one of the proposed languages, it is not possible to separate and quantify the impact of covariate shift and intrinsic difficulty for each language. However, there is no reason to believe that some of the languages under evaluation involve a higher difficulty in the speaker detection task. Thus, we work under the reasonable assumption that most of the difference in accuracy is due to the presence of covariate shift.

5.2. Pooling development and adaptation data

A traditional solution considered to solve this mismatch is pooling Ω_{dev} and all available $\Omega_{mached-lang}$ to retrain the complete speaker verification system, as proposed in Section 3.1. The results obtained considering this solution are shown in Table 4.

We observe improvement for Hindi-Urdu, and no improvement or a slight degradation for the other languages. The size of $\Omega_{mached-lang}$ is small compared to the size of Ω_{dev} so its influence in the background models is small. Thus, no further study on this technique is considered. Under the assumption that larger $\Omega_{mached-lang}$ are available for each language, this technique could be considered, or a single system could be retrained for every language separately, pooling Ω_{dev} and $\Omega_{mached-lang}$, or even considering $\Omega_{mached-lang}$ only.

Table 4: *Results pooling Ω_{dev} and $\Omega_{mached-lang}$. Results in bold are better than those obtained for the baseline system.*

Language	EER	minDCF
English	**1.30%**	0.0065
Chinese	5.14%	**0.0191**
Hindi-Urdu	**2.34%**	**0.0149**
Russian	3.90%	0.0213

5.3. Stacking total variability matrices

To ensure that the desired language is modeled in the system, we train a total variability matrix $T_{mached-lang}$ for each language, containing 50 new directions of maximum inter-session variability. This matrix is concatenated to the initial matrix T_{dev} considered in the baseline system, to obtain a low rank total variability matrix T with 450 columns. As explained in Section 3.2, the data in $\Omega_{mached-lang}$ and Ω_{dev} is then pooled to retrain the PLDA model and the mean for i-vector length normalization.

Table 5: *Results obtained stacking English and matched language total variability matrices. Results in bold are better than those obtained for the baseline system.*

Language	EER	minDCF
Chinese	**4.85%**	**0.0185**
Russian	**2.87%**	**0.0188**

Table 5 shows the results obtained considering the concatenated total variability matrix for each language. Note that we do not test this strategy on Hindi-Urdu since $\Omega_{Hindi-Urdu}$ is small to estimate a total variability matrix. It can be observed a slight improvement in all the results. The improvement is particularly high in the EER for Russian language.

5.4. I-vector adaptation

In order to perform i-vector adaptation we extract an i-vector for every matched development dataset $\Omega_{mached-lang}$ and obtain the mean $m_{matched-lang}$ of the set of i-vectors. The verification system is identical to the baseline, but during the evaluation, the adapted mean $m_{matched-lang}$ will be considered for i-vector centering and length normalization, as explained in Section 3.3.

Table 6 shows the results obtained for each language group when using i-vector adaptation. Compared to the results obtained by the baseline in Table 3, we observe improvement in EER and minDCF for Chinese (16% and 9% relative improvement respectively), and Hindi-Urdu (17% and 41% relative improvement respectively). For Russian there is a slight improvement in terms of minDCF (7% relative improvement) but degradation in EER (20% relative degradation).

This results indicate that there is actually a severe misalig-

Table 6: *Results obtained using language dependent i-vector adaptation. Results in bold are better than those obtained for the baseline system.*

Language	EER	minDCF
Chinese	**4.19%**	**0.0178**
Hindi-Urdu	**2.22%**	**0.0125**
Russian	3.65%	**0.0180**

ment between the i-vectors for English and for other languages. To quantify this misalignment we compute the Mahalanobis distance of the i-vector mean obtained on the English development dataset Ω_{dev} from each of the matched-language development datasets $\Omega_{mached-lang}$. This distance is defined as follows:

$$d = \sqrt{(m_{dev} - m_{lang})S_{lang}^{-1}(m_{dev} - m_{lang})}, \qquad (3)$$

where $m_{lang} = m_{matched-lang}$ and S_{lang} is the i-vector covariance matrix estimated on $\Omega_{mached-lang}$. For Hindi-Urdu we add the identity matrix to the covariance estimation for regularization purposes, since there are not enough Hindi-Urdu development recordings to estimate a full rank covariance matrix.

Table 7: *Mahalanobis distances between English and other languages.*

Language	Mahalanobis Distance
Chinese	3.94
Hindi-Urdu	6.25
Russian	4.19

Table 7 shows the Mahalanobis distances between the language dependent development datasets $\Omega_{mached-lang}$ and m_{dev}. These values are very high: note that the i-vectors from Ω_{dev} are distributed around m_{dev} (which is close to the origin) with covariance matrix close to the identity. The language dependent covariances show much higher variability, but the Mahalanobis distances are so high to assume that there is little or no overlap between English and any other language i-vector clouds.

This implies a severe misalignment during the process of mean subtraction and normalization when m_{dev} is considered. Without i-vector adaptation, the mean subtraction and length normalization will concentrate the i-vectors from a non-English language on a reduced region of the hypersphere of unit radius. Thus, we expect the i-vectors belonging to a language different from English to be closer and thus to introduce a shift in the scores.

Therefore, it is reasonable to use an estimate of the mean of the evaluation dataset using $\Omega_{mached-lang}$ instead of the mean m_{dev}.

5.5. Score Normalization and Calibration

To test score normalization, we consider the baseline system and we use the recordings from $\Omega_{mached-lang}$ as impostor cohort. We select a subset ensuring that there are no more than two sessions for a single speaker. Because of the architecture of the system, score normalization must be gender dependent. We use S-norm as normalization technique. For comparison, we also consider the case of using the mismatched development dataset Ω_{dev} for score normalization. This dataset is also considered for normalization in the English evaluation set.

Table 8: *Results for S-norm. Results in bold are better than those obtained for the baseline system.*

Normalization	English S-norm		Matched S-norm	
Language	EER	minDCF	EER	minDCF
English	**1.26%**	**0.0058**	**1.26%**	**0.0058**
Chinese	**3.83%**	**0.0194**	4.23%	0.0210
Hindi-Urdu	**2.40%**	**0.0179**	4.65%	0.0296
Russian	4.22%	**0.0155**	**2.79%**	0.0161

Table 8 shows the results obtained for language matched and mismatched S-norm. It is interesting to observe that, in general, using English data for normalization enables us to obtain better results than using matched S-norm. We have carried out preliminary experiments that indicate that S-norm and other normalization techniques require a large amount of recordings from different speakers. The matched datasets $\Omega_{mached-lang}$ considered in this study are very small for some languages (especially for Hindi-Urdu), therefore, a large mismatched dataset is preferred. Note also that the available recordings are divided into two independent groups since normalization is gender dependent.

Comparing the results to those obtained for the baseline system, we can observe that gender dependent S-norm is actually helpful for this system, using Ω_{dev} to obtain the impostor cohort.

Table 9: *Matched and mismatched calibration. Best results are in bold.*

Calibration data	English		Matched	
Language	minDCF	actDCF	minDCF	actDCF
English	0.0060	**0.0062**	0.0060	**0.0062**
Chinese	0.0196	0.0839	0.0196	**0.0201**
Hindi-Urdu	0.0214	0.0583	0.0214	**0.0244**
Russian	0.0193	0.1569	0.0193	**0.0201**

Finally, we analyze the importance of using matched data for training the calibration. Table 9 shows the results in terms of minDCF and actDCF (actual DCF)otained for the baseline system when considering Ω_{dev} and $\Omega_{mached-lang}$ for score calibration. From the results in the table we can extract two main conclusions. First, matched calibration is mandatory to the correct operation of the system. Second, a small matched dataset is enough to obtain a good calibration.

The significant degradation in terms of actDCF obtained when considering mismatched data for calibration indicates that the score distributions are different for different languages. Figure 2 compares the target and non-target score distributions for obtained English and Chinese. Note that Chinese scores are in general higher than English scores. This is consistent with the results observed in Section 5.4, where we show that the i-vector clouds from different languages are far from the English i-vector cloud, and thus the length normalization using m_{dev} for centering will confine the i-vectors from a different language in a small region of the hypersphere of unit radius. Since the i-vectors are closer in the space, we expect the scores to be higher.

This problem can be solved using the i-vector adaptation technique evaluated in Section 5.4. However, given the good re-

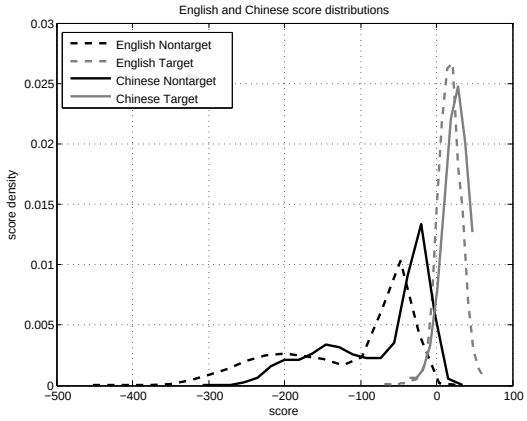

Figure 2: *Score distributions for English and Chinese.*

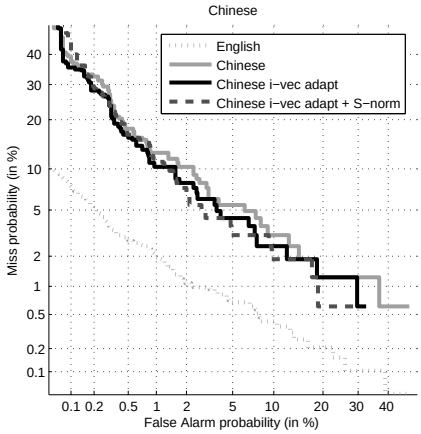

Figure 3: *DET curves for Chinese considering no compensation, i-vector adaptation and i-vector adaptation + S-norm for compensate covariate shift due to language mismatch.*

sults obtained when training the calibration using matched data, the best solution to ensure good calibration is to train the calibration with matched developed data.

5.6. Comparison and summary

Tables 10 and 11 compare the techniques analyzed to mitigate covariate shift in terms of EER and minDCF. They also include some results combining the proposed techniques. We observe that most of the proposed techniques for covariate shift compensation and their combination enables us to improve the accuracy of the speaker verification system. Most of the techniques introduce a slight improvement.

The i-vector adaptation itself and in combination with other techniques, specially with S-norm (using English impostor cohort), improves significantly the results. Among the proposed approaches, the i-vector adaptation and S-norm are very convenient since they do not involve the retraining of any background model of the speaker verification system. They only require the recalibration of the system using matched data, that is needed in any case. On the other hand, Stacking total variability matrices does not generally provide any additional improvement over these techniques, and it implies the retraining of the PLDA model.

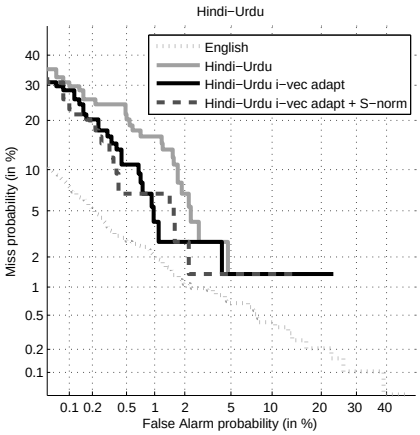

Figure 4: *DET curves for Hindi-Urdu considering no compensation, i-vector adaptation and i-vector adaptation + S-norm for compensate covariate shift due to language mismatch.*

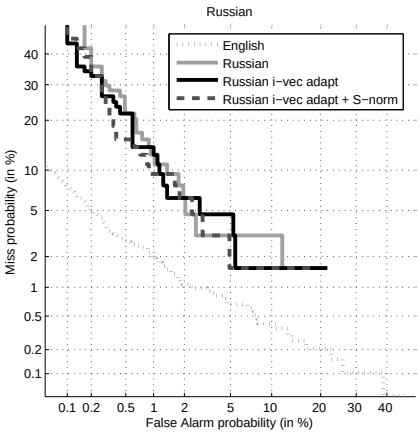

Figure 5: *DET curves for Russian considering no compensation, i-vector adaptation and i-vector adaptation + S-norm for compensate covariate shift due to language mismatch.*

Therefore, among the proposed techniques we select the i-vector adaptation and the i-vector adaptation with S-norm. Figures 3, 4 and 5 compare the DET curves obtained considering these techniques, for every group of languages under test. The DET curves represented are obtained considering no compensation for covariate shift, i-vector adaptation, and i-vector adaptation + S-norm. For comparison, the DET curve obtained for English matched data is also shown.

Note that we obtain improvement for all languages in most regions of the DET. However, there is still a wide gap between the best DET curves for each language and the DET curve for English, so there is still a lot of work to do in the line of compensating for covariate shift when a small matched dataset is available for development.

6. Conclusions

In this work we have analyzed a state-of-the-art i-vector PLDA system in order to determine the components that are most vulnerable to dataset shift, specifically to covariate shift. From this analysis we can conclude that an i-vector PLDA system is not

Table 10: *Comparison and combination of the proposed techniques in terms of EER. Best results are in bold.*

EER	Technique						
Language	Baseline	Stacking	ivec-adapt	Snorm	Stack.+ivec-adapt	ivec-adapt+Snorm	Stack.+ivec-adapt+Snorm
Chinese	4.99%	4.85%	4.19%	3.83%	5.12%	**3.78%**	4.64%
Hindi-Urdu	2.69%	N/A	2.22%	2.40%	N/A	**1.93%**	N/A
Russian	3.04%	**2.87%**	3.65%	4.22%	3.62%	3.04%	3.44%

Table 11: *Comparison and combination of the proposed techniques in terms minDCF. Best results are in bold.*

minDCF	Technique						
Language	Baseline	Stacking	ivec-adapt	Snorm	Stack.+ivec-adapt	ivec-adapt+Snorm	Stack.+ivec-adapt+Snorm
Chinese	0.0196	0.0185	**0.0178**	0.0194	**0.0178**	0.0192	0.0196
Hindi-Urdu	0.0214	N/A	0.0125	0.0179	0.0131	**0.0098**	N/A
Russian	0.0193	0.0188	0.0180	**0.0155**	0.0160	0.0167	0.0176

very flexible to take into account a small development dataset to compensate for covariate shift. This is mainly because it involves two separate stages that need large amounts of development data to be trained, the i-vector extractor and the PLDA model. Thus, it is not possible to compensate for covariate shift adapting only in a single stage. In addition, modifying the i-vector extractor implies the retraining of the PLDA model, and this is not possible in many real cases.

We have proposed several techniques to compensate for covariate shift given a small matched development dataset, some of them based on recent advances in speaker verification. These techniques include pooling all available data, stacking total variability matrices, i-vector adaptation using i-vector centering and length normalization, and S-norm. They have been evaluated on a scenario with covariate shift. This scenario has been built using NIST SRE data and considering language mismatch as covariate shift. This scenario has shown to be a valid framework to evaluate covariate shift compensation strategies, although the number of trials per evaluation is not large. Note that language mismatch is just an example of an scenario with covariate shift. The proposed techniques are not developed specifically to solve the language mismatch problem, and they could be used in scenarios with other sources of covariate shift, as channel mismatch or noisy environments.

Among the evaluated measures, i-vector adaptation and S-norm have been the most successful, although all have shown a slight improvement in terms of EER and minDCF for all languages in most cases. However, the results are still far from those obtained considering an scenario without covariate shift. Therefore, there is still much work to do towards adaptation techniques for state-of-the-art speaker verification systems, in order to take advantage of a small matched development dataset in scenarios with covariate shift.

7. Acknowledgments

The author would like to thank Luis Buera and Niko Brümmer for fruitfully discussions.

8. References

[1] Vaquero, C., Scheffer, N. and Kajarekar, S., "Impact of Prior Channel Information for Speaker Identification", International Conference on Biometrics (ICB), 443–453, Alghero, Italy, 2009.

[2] Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D., "Dataset Shift in Machine Learning", The MIT Press, 2008.

[3] Moreno-Torres, J., Raeder, T., Alaiz-Rodriguez, R., Chawla, N. V. and Herrera, F., "A Unifying View on Dataset Shift in Classification", Pattern Recognition, 45, 521–530, 2011.

[4] Kenny, P., Boulianne, G., Ouellet, P., and Dumouchel, P., "Speaker and Session Variability in GMM-based Speaker Verification". IEEE Transactions on Audio, Speech and Language Processing, 15(4):1448–1460, 2007.

[5] Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., and Ouellet, P.,"Front-End Factor Analysis For Speaker Verification". IEEE Transactions on Audio, Speech and Language Processing, 2010.

[6] Kenny, P., "Bayesian Speaker Verification with Heavy-Tailed Priors". In Proc. Odyssey, Brno, Czech Republic, 2010.

[7] Garcia-Romero, D. and Espy-Wilson, C. Y., "Analysis of I-vector length Normalization in Speaker Recognition Systems". In Proc. Interspeech, 249–252, Florence, Italy, 2011.

[8] Villalba, J. and Brümmer, N., "Towards Fully Bayesian Speaker Recognition: Integrating Out the Between Speaker Covariance". In Proc. Interspeech, 505–508, Florence, Italy, 2011.

[9] Burget, L. et al, "BUT system for NIST 2008 speaker recognition evaluation", In Proc. Interspeech, 2335–2338, Brighton, 2009.

[10] Senoussaoui, M., Kenny, P., Dehak, N. and Dumouchel, P., "An i-vector Extractor Suitable for Speaker Recognition with both Microphone and Telephone Speech", In Proc. Odyssey, Brno, Czech Republic, 2010.

[11] Kenny, P., "Joint Factor Analysis of Speaker and Session Variability: Theory and Algorithms". Technical report CRIM, Montreal, Canada, 2005.

[12] Sennoussaoui, M., Kenny, P., Brummer, N., de Villiers, E., and Dumouchel, P., "Mixture of PLDA Models in I-Vector Space for Gender-Independent Speaker Recognition", In Proc. Interspeech, Florence, Italy, 2011.