On robustness of unsupervised domain adaptation for speaker recognition

Pierre-Michel Bousquet, Mickael Rouvier

On robustness of unsupervised domain adaptation for speaker recognition

Pierre-Michel Bousquet, Mickael Rouvier

University of Avignon - LIA, France

{pierre-michel.bousquet, mickael.rouvier}@univ-avignon.fr

Abstract

Current speaker recognition systems, that are learned by using wide training datasets and include sophisticated modelings, turn out to be very specific, providing sometimes disappointing results in real-life applications. Any shift between training and test data, in terms of device, language, duration, noise or other tends to degrade accuracy of speaker detection. This study investigates unsupervised domain adaptation,when only a scarce and unlabeled "in-domain" development dataset is available. Details and relevance of different approaches are described and commented, leading to a new robust method that we call feature-Distribution Adaptor. Efficiency of the proposed technique is experimentally validated on the recent NIST 2016 and 2018 Speaker Recognition Evaluation datasets.

Index Terms: Speaker recognition, speaker embeddings, x-vectors, unsupervised, domain adaptation

1. Introduction

As any application of machine learning, effectiveness of the automatic speaker recognition relies on extensive model training datasets. The term *big data* often used in machine learning implicitly means that these datasets are comprised of huge amounts of labeled observations but, also, span a wide variety of real settings (for utterances: channel, device, duration, language, type and level of noise, reverberation, etc.). Actually, these requirements about the "scope of domain" are not fulfilled and this fact can explain the disappointing results of some real-life speaker recognition applications.

We focus here on the challenge of unsupervised domain adaptation, when the methods have to transmit information about domain shift from a small unlabeled in-domain development dataset to the wide out-of-domain training dataset used for modeling.

Methods can be model-based, adapting parameters of a model by a mix of techniques as interpolation [1, 2], nuisance attribute projection [3], Bayesian maximum likelihood [4] or / and eigenvectors and eigenvalue-spectrum regularization [2, 1]. The methods can also be feature-based, transforming out-of-domain data to better fit the in-domain distribution [5, 6].

For model-based methods, the unsupervised domain adaptation can also be dealt with by carrying out a clustering of the in-domain development dataset, identifying the clusters with speaker-classes then performing supervised domain adaptation with these new speakers labels [7, 8]. On the one hand, it is shown in [5] that these model-based methods perform better when they are preceded by a feature-based unsupervised adaptation (CORAL in [5]). On the other hand, scores provided by a PLDA remain the best metric for clustering [9, 7, 10, 11] and unsupervised methods can be useful to better estimate the PLDA parameters. These remarks show the usefulness of unsupervised domain adaptation.

Our contributions are as follows: we describe the most efficient unsupervised domain adaptation methods in section 2, in particular the algorithm implemented in Kaldi [1]. This code has been used by many participants during NIST-SRE18 [12] and no documentation is available. In section 3, details about the methods are reviewed and discussed, leading to propose a new method, intended to enhance accuracy and increase robustness of the adaptation. We call this new method "feature-Distribution Adaptor". Section 4 reports experimental results and we conclude in section 5.

In what follows, the terms in-domain and out-of-domain are abbreviated by inD and ooD.

2. Domain adaptation methods

Figure 1 details the steps of different speaker recognition backend processes with embeddings (i-vector or x-vector) and feature- or model-based domain adaptation. For comparison, the first row describes a system without domain adaptation. We call it standard as it is the most commonly implemented: after LDA dimensionality reduction, the vectors are whitened by centering (subtracting the mean vector of the training dataset) and standardization by the within-class covariance matrix (W-norm [13, 14]), then length-normalized. Gaussian-PLDA model is estimated, following [15], usually with full-rank speaker and nuisance covariance. Vectors are scored by using this Gaussian model.

The second system in Figure 1 describes the example recipe in Kaldi. As no documentation of its backend process is available, we detail it here. The process uses many specific steps: first, the version of PLDA is the one proposed by Ioffe [16]. This PLDA includes a post-modeling normalization: given the estimated between and within class covariance matrices $(\mathbf{B}, \mathbf{W})$, W-normalization is applied then rotation by the $\mathbf{B}$-eigenvectors. This additional step makes diagonal both matrices but is is worth noting that it modifies the within-class covariance after modeling. Then, a first step of domain adaptation is carried out ("by-domain mean adapt"): each vector is centered around the mean of its specific domain (its own mean for ooD, the mean computed from unlabeled development data for inD).

The Kaldi recipe contains an algorithm of covariance adaptation, referred to as PLDA unsupervised adaptor, that we detail in Algorithm 1. In step (i), eigenvalue decomposition is carried out, $\mathbf{P}$ is the eigenvector matrix and $\mathbf{\Delta}$ the diagonal matrix of corresponding eigenvalues. Then parameters are transformed as shown in step (ii). Let us note that applying the transformation of step (ii) to inD and ooD total covariance matrices would make them simultaneously diagonal, equal to Δ and the identity matrix respectively. Therefore, in this space, the dimensional variances (eigenvalues) of the ooD distribution are all equal to 1. The method assumes that the inD variances higher than 1 are specific to this domain. During step (iii), only diagonal values of the model parameters are updated and the result is interpo-

[1] https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16/v2

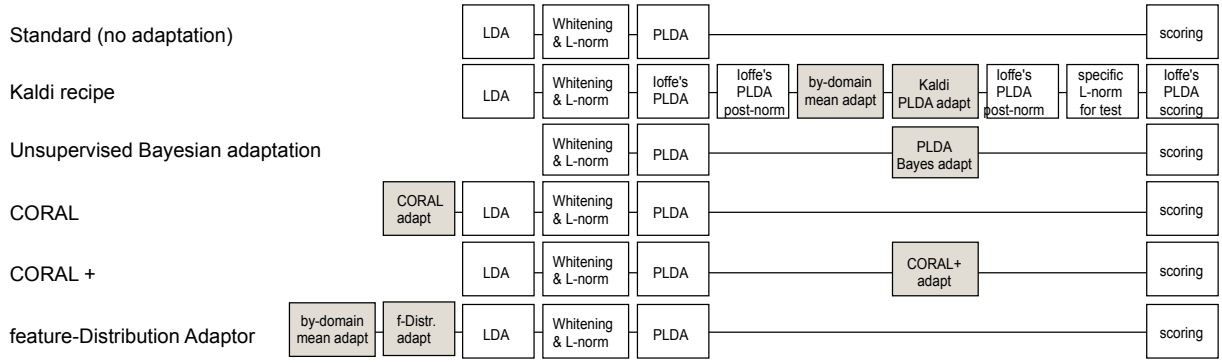

Figure 1: *Details of some systems with unsupervised domain adaptation. For better comparison, the first one corresponds to a baseline without domain adaptation.*

Algorithm 1 Kaldi PLDA unsupervised adaptor

Given PLDA matrices $(\mathbf{B}, \mathbf{W})$, compute covariance matrices $\mathbf{\Sigma}_i$ and $\mathbf{\Sigma}_o = \mathbf{B} + \mathbf{W}$ of inD and ooD data.
(i)
Compute SVD of $\mathbf{\Sigma}_o^{-\frac{1}{2}} \mathbf{\Sigma}_i \mathbf{\Sigma}_o^{-\frac{1}{2}} = \mathbf{P} \mathbf{\Delta} \mathbf{P}^t$
(ii)
$\mathbf{W} \leftarrow \mathbf{P}^t \mathbf{\Sigma}_o^{-\frac{1}{2}} \mathbf{W} \mathbf{\Sigma}_o^{-\frac{1}{2}} \mathbf{P}$
$\mathbf{B} \leftarrow \mathbf{P}^t \mathbf{\Sigma}_o^{-\frac{1}{2}} \mathbf{B} \mathbf{\Sigma}_o^{-\frac{1}{2}} \mathbf{P}$
(iii)
for $i = 1$ to r **do**
  if $\mathbf{\Delta}_{i,i} > 1$ **then**
    $\mathbf{B}_{i,i} = \mathbf{B}_{i,i} + \alpha_b \left(\mathbf{\Delta}_{i,i} - 1\right)$
    $\mathbf{W}_{i,i} = \mathbf{W}_{i,i} + \alpha_w \left(\mathbf{\Delta}_{i,i} - 1\right)$
(iv)
$\mathbf{W} \leftarrow \mathbf{\Sigma}_o^{+\frac{1}{2}} \mathbf{P} \mathbf{W} \mathbf{P}^t \mathbf{\Sigma}_o^{+\frac{1}{2}} \mathbf{P}$
$\mathbf{B} \leftarrow \mathbf{\Sigma}_o^{+\frac{1}{2}} \mathbf{P} \mathbf{B} \mathbf{P}^t \mathbf{\Sigma}_o^{+\frac{1}{2}} \mathbf{P}$

lating between initial and transformed parameter. The positive parameters α_b, α_w are such that $\alpha_b + \alpha_w = 1$ and allow to freely weight the initial ooD and pseudo-inD covariances. The last step brings back the transformed parameters in the initial space.

After adaptation, Ioffe's PLDA post-normalization must be applied for a second time, that further modifies features and PLDA parameters. Lastly, a supplementary normalization is performed ("specific L-norm for test") before scoring. The test-vectors are modified as follows :

$$x \longleftarrow \frac{x}{\left\| x^t \widehat{\mathbf{\Sigma}}_o^{-1} x \right\|} \tag{1}$$

where $\widehat{\mathbf{\Sigma}}_o = \mathbf{B} + \mathbf{W}$ is the total covariance matrix of the pseudo-inD data after adaptation. Experiments show that this transformation is required to achieve competitive performance. We did not find theoretical justification of this post-modeling normalization. The system ends up by the specific scoring of Ioffe's PLDA model.

The third system is the unsupervised Bayesian adaptation proposed in [4]. The authors apply the supervised Bayesian adaptation of PLDA [8] to unlabelled inD datasets. The inD development data do not need clustering: the observations are assumed to be speaker-independent, ie. that each sample is provided by a unique speaker. This assumption eliminates the requirement of speaker labeling. The interesting results of this approach, competitive to those of the supervised version, may be due to the fact that the inD development dataset of the experiments NIST SRE16 [17] contains few samples per speaker (about 2).

The fourth system in Figure 1 includes the CORAL domain adaptation presented in [18] and introduced for speaker recognition in [5]. This feature-based method is inserted as initial step of the process. Given the covariance matrices $\mathbf{\Sigma}_i$ and $\mathbf{\Sigma}_o$ of the inD and ooD domain, the authors aim at finding a matrix $\mathbf{A}$ that minimizes the distance between $\mathbf{\Sigma}_i$ and $\mathbf{A}^t \mathbf{\Sigma_o} \mathbf{A}$, using the Frobenius norm metric. The analytical solution is equal to :

$$\mathbf{A}^t = \mathbf{\Sigma}_i^{+\frac{1}{2}} \mathbf{\Sigma}_o^{-\frac{1}{2}} \tag{2}$$

Actually, the authors apply a regularization term to this solution:

$$\mathbf{A}^t = \left(\lambda \mathbf{I} + \mathbf{\Sigma}_i\right)^{+\frac{1}{2}} \left(\lambda \mathbf{I} + \mathbf{\Sigma}_o\right)^{-\frac{1}{2}} \tag{3}$$

where $\mathbf{I}$ is the identity matrix and λ a positive scalar. Each ooD vector x becomes $x \leftarrow \mathbf{A}^t x$. In [18], the regularization factor λ is introduced in order to avoid complications when matrices are not full-rank, that is for the sake of efficiency and stability. The authors arbitrarily set this parameter to 1. This value is retained in [5] for speaker recognition.

A new domain adaptation method has been recently introduced and tested for SRE18, referred to as CORAL+ [1]. Unlike CORAL and like Kaldi adaptor, this method (fifth system of Figure 1) is model-based. In the same way as step (ii) of Kaldi PLDA unsupervised adaptor (but separately for each PLDA parameter $\mathbf{B}$ and $\mathbf{W}$), a simultaneous diagonalization of the parameter and of its pseudo-inD covariance version is carried out. The result is a combination of the two matrices (the latter is beforehand thresholded to 1 in terms of eigenvalues). The parameter-dependent scalars for interpolation are experimentally tuned, to optimize performance.

3. Feature-Distribution adaptor

We tested during and after NIST-SRE18 evaluation some variants of the previous methods. They are based on the following arguments.

First, speaker embeddings as i-vector or x-vector have proven to be very efficient for speaker detection but these low-rank representations are not "ready" for modeling and scoring. They can be used provided that they are first whitened and

normalized. Transformations like discriminant dimensionality reduction (LDA), whitening and length-normalization are required to make these representation consistent with the usual and well controlled Gaussian probabilistic framework. All these techniques rely on parameters learned by using ooD data. Hence, InD data follow transformations driven by ooD parameters. Specific information of the target domain can be degraded during these stages. It could be more relevant to adapt ooD data to the target domain before any of these techniques.

Second, it is known that the mean shift due to the mismatch between training and test data can be partially solved by centralizing the datasets to a common origin [19]. Also in [19], a severe misalignment is observed between the i-vectors for English and for other languages. The Kaldi recipe contains a by-domain mean adaptation (step 5 in Figure 1), performed after normalization and PLDA. We propose to apply this adaptation at the same time as LDA. Indeed, length-normalization, which is done before the mean adaptation in Kaldi code, is critically depending on the location of the origin.

Third, as explained above, the practical implementation of CORAL does not use the analytical solution of the minimization problem of eq. (2) but a regularized version (eq. (3) with $\lambda = 1$), in which the identity is added to the covariances. The resulting matrices have the same eigenvectors that the initial covariances matrices and the same eigenvalues increased by 1. By this way, the impact of the residual components is mitigated, avoiding to take into account inaccurate information during adaptation. This precaution is especially useful for inD covariance, which is estimated by using a small development dataset (in our experiments, about 2000 observations for estimating a 512×512 matrix).

As noticed in Section 2, after step (ii) of Kaldi adaptor Algorithm 1, inD and ooD total covariances matrices would be simultaneously diagonal, equal to Δ and the identity matrix respectively. In this intermediate space, it can be presumed that the inD variances (eigenvalues) higher than 1 are specific to this domain and, also, that low eigenvalues are unreliable -keeping in mind the low amount of the inD development data-. The best way for adaptation is to carry out an eigenvalue-spectrum regularization of the ooD matrices.

By synthesizing all these remarks, it can be of interest to propose the following Algorithm 2. As shown in Figure 1, this algorithm has just to be added as first step to a standard system without adaptation. We refer to this method as "feature-

Algorithm 2 feature-Distribution Adaptor

Apply by-domain mean adaptation to inD and ooD vectors.
Compute covariance matrices $\mathbf{\Sigma}_i$, $\mathbf{\Sigma}_o$ of inD and ooD data.
Compute SVD of $\mathbf{\Sigma}_o^{-\frac{1}{2}} \mathbf{\Sigma}_i \mathbf{\Sigma}_o^{-\frac{1}{2}} = \mathbf{P} \mathbf{\Delta} \mathbf{P}^t$
Compute matrix $\widehat{\Delta}$ such that $\widehat{\Delta}_{i,i} = \max(1, \Delta_{i,i})$
For each ooD vector x **do** $x \leftarrow \left(\mathbf{\Sigma}_o^{+\frac{1}{2}} P \widehat{\Delta}^{\frac{1}{2}} P^t \mathbf{\Sigma}_o^{-\frac{1}{2}} \right) x$

Distribution Adaptor" by analogy with Kaldi PLDA unsupervised adaptor. The transformation avoids to adapt axes of variability higher than those of inD data, by applying a flooring of 1 to the estimated inD-eigenvalues in a whitened space. Let us note that, if $\widehat{\Delta} = \Delta$, the resulting ooD covariance becomes $\mathbf{\Sigma}_i$.

Unlike Kaldi, the method is feature-based, and works on the whole covariance matrix (not only its diagonal values). No specific SVD per within- or between- covariance is carried out, as done in model-based CORAL+. We presume that the limited size of the inD development dataset may make their estimation inaccurate. There is no interpolation between the original and the resulting matrix, as in [7, 4, 2, 1]. The latter is considered as the best pseudo-inD estimation of the covariance, given the available inD and ooD data.

To better assess the relevance of the previous approach and, also, the gain of accuracy involved by the "first step" feature-based adaptation compared to a post-normalization model-based adaptation, we also propose a modified version of the PLDA unsupervised adaptor of Kaldi. This new version is described in Algorithm 3. Following the same arguments as those

Algorithm 3 Modified-Kaldi

Apply by-domain mean adaptation to inD and ooD vectors.
(...)
Apply step (i) of Algo. 1
Replace steps (ii) to (iv) of Algo. 1 by:
Compute matrix $\widehat{\Delta}$ such that $\widehat{\Delta}_{i,i} = \max(1, \Delta_{i,i})$
$\mathbf{B} = \left(\mathbf{\Sigma}_o^{+\frac{1}{2}} P \widehat{\Delta}^{\frac{1}{2}} P^t \mathbf{\Sigma}_o^{-\frac{1}{2}} \right) \mathbf{B} \left(\mathbf{\Sigma}_o^{-\frac{1}{2}} P \widehat{\Delta}^{\frac{1}{2}} P^t \mathbf{\Sigma}_o^{+\frac{1}{2}} \right)$
$\mathbf{W} = \left(\mathbf{\Sigma}_o^{+\frac{1}{2}} P \widehat{\Delta}^{\frac{1}{2}} P^t \mathbf{\Sigma}_o^{-\frac{1}{2}} \right) \mathbf{W} \left(\mathbf{\Sigma}_o^{-\frac{1}{2}} P \widehat{\Delta}^{\frac{1}{2}} P^t \mathbf{\Sigma}_o^{+\frac{1}{2}} \right)$

set out above, about the flooring of eigenvalues to 1 in a specific space, the Kaldi adaptor can be improved by replacing steps (ii) to (iv) of Algorithm 1 as done in Algorithm 3. As for the initial PLDA unsupervised adaptor, the intermediate inD and ooD covariance matrices are simultaneously diagonal but, here, the adaptation is performed on the whole matrix and not only on the diagonal values. Let us note that this modification can be implemented in Kaldi with very small changes.

4. Experiments

4.1. Configuration

X-vector and i-vector have been trained using data collected from NIST SRE2004, 2005, 2006, 2008 and from Switchboard II phase 2,3 and Switchboard Cellular Part1 and Part2.

For x-vectors, we use MFCC feature withe 23 cepstral coefficients. The window length and shift size are 25-ms and 10-ms respectively. A cepstral mean normalization is applied with a window size of 3 seconds. Next, non-speech frames are discarded using energy-based voice activity detection. The x-vector extractor is a variant of Kaldi toolkit [2] in which we implemented attentive statistics pooling layer [20] . The attentive statistics pooling layer calculates weighted means and weighted standard deviations over frame-level features scaled by an attention model. This enables x-vector to focus only on important frames. The setting of x-vector network follows the example recipe in kaldi : *sre16/v2* except the number of epochs that is fixed to 6 and the size of mini-batch that is fixed to 128. The extracted x-vectors are 512-dimension. The system employs the data augmentation included in Kaldi.

For i-vectors, we use MFCC feature with 20 cepstral coefficients appended with the first and second order. A 4096-mixture full covariance UBM has been trained. The extracted i-vectors are 400-dimension.

Experiments were conducted on the NIST SRE16 [17] and SRE18 datasets. The training set consists primarily English speech, the enrollment and test segments are in Tagalog and Cantonese for SRE16, in Tunisian Arabic (*cmn2*) for SRE18. The unlabeled development sets comprises 2272 and 2332 seg-

Table 1: *Comparison of x-vector systems with distinct domain adaptation methods.*

system	*By-domain mean adapt.*	**eval SRE18** cmn2 eer	dcf	**eval SRE16** tagalog eer	dcf	cantonese eer	dcf
standard (no adapt.)	-	10.67	0.669	20.96	0.996	6.89	0.561
Kaldi	✓	7.61	0.544	11.04	0.734	3.48	0.359
Kaldi (without by-D mean adapt)	-	9.76	0.587	15.52	0.835	5.12	0.451
Unsupervised Bayesian adaptation	-	10.15	0.628	19.97	0.837	7.14	0.529
CORAL	-	11.94	0.618	17.52	0.859	8.55	0.544
CORAL+	-	9.88	0.587	17.43	0.842	5.62	0.472
Unsupervised Bayesian adaptation	✓	7.73	0.600	13.19	0.806	3.89	0.404
CORAL	✓	8.12	0.581	11.24	0.710	4.16	0.412
CORAL+	✓	8.47	0.565	13.16	0.791	4.49	0.401
CORAL+ with specific L-norm.	✓	7.32	0.549	10.94	0.735	3.33	0.359
feature-Distribution Adaptor	✓	**7.22**	**0.508**	**10.31**	**0.688**	3.31	**0.335**
Modified-Kaldi	✓	7.35	0.544	10.92	0.732	**3.28**	0.350

ments respectively. Robust error measures are computed by using the NIST toolkit, that delivers averages of EER and DCF from many partitions.

4.2. Results

Results of the different approaches using x-vector system are reported in Table 1. The first two lines show the benefit of the Kaldi recipe, for all the experiments. Next four lines show results of Kaldi, unsupervised Bayesian adaptation, CORAL and CORAL+ without initial by-domain mean adaptation (with only centering during the standard whitening step). The disappointing results suggest that some of these methods may include unmentioned mean adaptation, at one of their stages.

The last six lines of Table 1 confirm the benefit of the initial by-domain mean adaptation proposed in Section 3. For CORAL+, as the results of this method were disappointing (compared to those obtained by the authors during NIST-SRE18 and in [1]), we tested the method with the last specific length-normalization of eq. (1), performed after modeling, that is included in Kaldi. With this addition, the method provides a significant gain of performance for all the experiments.

The last two lines show results of our contribution. Feature-distribution Adaptor confirms the assumptions of Section 3 and the relevance of the approach in terms of robustness. The method yields better performance than the previous ones, for all the experiments and error measurements. Lastly, the modified-Kaldi approach confirms the effectiveness of the matrix transformation of Algorithm 3 but, as presumed in Section 3, performing domain adaptation after normalization leads to lower accuracy than the previous approach.

Table 2: *Comparison of i-vector and augmented x-vector based systems on NIST-SRE 2016 (standard without adaptation then feature-Distribution Adaptor).*

		tagalog eer	dcf	cantonese eer	dcf
standard	i-vector	23.07	0.972	10.32	0.698
f-D Adaptor	i-vector	16.09	0.823	6.51	0.544
f-D Adaptor	x-vector	10.31	0.688	3.31	0.335

Results reported in Table 2 compare performance of two representations: i-vector vs x-vector, on the same evaluation SRE16. The Table allows to better assess the benefit of the new augmented x-vector approach, but also that of the domain adaptation: for EER, about half of the gain is brought by the embedding and the other half by the adaptation of mean and covariance. This recalls the importance of an optimized backend process in robust speaker recognition.

5. Conclusion

In this study, we focused on unsupervised domain adaptation with recent speaker embeddings (i-vector, augmented x-vector). These low-size representations turn out to be very efficient, but have the major drawback of not being "ready" for scoring. Normalizations are required for preparing data to modeling and scoring. As highlighted in this study, domain adaptation could improve its relevance to be included as initial step of the system, before any type of normalizations (all being learned by using out-of-domain information). The adaptation procedure we propose is relevant, since done on features and as initial step. Dealing with unsupervised domain adaptation merely boils down to adding a preliminary step to a "standard" backend system.

Our experiments highlight the substantial share of performance brought by an appropriate mean adaptation. The by-domain mean adaptation has already been implemented in some systems, but not necessarily as initial step before normalization. We think that this point deserved to be experimentally confirmed. For covariance, the proposed regularization procedure, based on comparison of eigenvalue-spectra in a "whitened" space, appears to be more efficient than all other approaches, especially when performed as first step. As explained in the introduction, the feature-Distribution Adaptor can also be used to refine the prior clustering of supervised adaptation methods.

This method appears to us quite simple and relevant enough to be robust and, thus, to be tested in future work for other mismatch of domain than language.

6. Acknowledgements

This research was supported by the ANR agency (Agence Nationale de la Recherche), VoxCrim project (ANR-17-CE39-0016)

7. References

[1] K. A. Lee, Q. Wang, and T. Koshinaka, "The CORAL+ algorithm for unsupervised domain adaptation of PLDA," *CoRR*, vol. abs/1812.10260, 2018. [Online]. Available: http://arxiv.org/abs/1812.10260

[2] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *icassp*, 2018, pp. 5329–5333.

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

[4] B. J. Borgstrom, E. Singer, D. Reynolds, and O. Sadjadi, "Improving the effectiveness of speaker verification domain adaptation with inadequate in-domain data," in *interspeech*, 2017.

[5] J. Alam, G. Bhattacharya, and P. Kenny, "Speaker verification in mismatched conditions with frustratingly easy domain adaptation," in *Speaker and Language Recognition Workshop (IEEE Odyssey)*, 2018.

[6] J. Alam, P. Kenny, G. Bhattacharya, and M. Kockmann, "Speaker verification under adverse conditions using i-vector adaptation and neural network," in *interspeech*, 2017.

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

[8] E. L. Jesus Villalba, "Bayesian Adaptation of PLDA Based Speaker Recognition to Domains with Scarce Development Data," in *Speaker and Language Recognition Workshop (IEEE Odyssey)*, 2012.

[9] S. Shum, D. Reynolds, D. Garcia-Romero, , and A. McCree, "Unsupervised clustering approaches for domain adaptation in speaker recognition systems," in *Speaker and Language Recognition Workshop (IEEE Odyssey)*, 2014.

[10] J. Villalba and al., "The JHU-MIT System Description for NIST SRE18," in *NIST Speaker Recognition Evaluation Workshop*, 2018.

[11] J. Alam and al., "ABC NIST SRE 18 System description," in *NIST Speaker Recognition Evaluation Workshop*, 2018.

[12] National Institute of Standards and Technology, "Speaker recognition evaluation plan 2018," https://www.nist.gov/document/sre18evalplan2018-05-31v6pdf, 2018.

[13] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-Class Covariance Normalization for SVM-based Speaker Recognition," in *International Conference on Speech Communication and Technology*, 2006, pp. 1471–1474.

[14] P.-M. Bousquet, A. Larcher, D. Matrouf, J.-F. Bonastre, and O. Plchot, "Variance-Spectra based Normalization for I-vector Standard and Probabilistic Linear Discriminant Analysis," in *Speaker and Language Recognition Workshop (IEEE Odyssey)*, 2012.

[15] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[16] S. Ioffe, "Probabilistic linear discriminant analysis," *Computer Vision*, pp. 531–542, 2006.

[17] National Institute of Standards and Technology, "Speaker recognition evaluation plan 2016," https://www.nist.gov/document/sre16evalplanv13pdf, 2016.

[18] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," *CoRR*, vol. abs/1511.05547, 2015. [Online]. Available: http://arxiv.org/abs/1511.05547

[19] C. Vaquero, "Dataset Shift in PLDA based Speaker Verification," in *Speaker and Language Recognition Workshop (IEEE Odyssey)*, 2012.

[20] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *interspeech*, 2018, pp. 2252–2256.