INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Curriculum Learning based Probabilistic Linear Discriminant Analysis for Noise Robust Speaker Recognition

Shivesh Ranjan, Abhinav Misra, John H. L. Hansen

Center for Robust Speech Systems (CRSS)
The University of Texas at Dallas, Richardson, TX, USA
{Shivesh.Ranjan, Abhinav.Misra, John.Hansen}@utdallas.edu

Abstract

This study introduces a novel Curriculum Learning based Probabilistic Linear Discriminant Analysis (CL-PLDA) algorithm for improving speaker recognition in noisy conditions. CL-PLDA operates by initializing the training EM algorithm with cleaner data (*easy* examples), and successively adds noisier data (*difficult* examples) as the training progresses. This curriculum learning based approach guides the parameters of CL-PLDA to better local minima compared to regular PLDA. We test CL-PLDA on speaker verification task of the severely noisy and degraded DARPA RATS data, and show it to significantly outperform regular PLDA across test-sets of varying duration.

Index Terms: curriculum learning, speaker recognition, PLDA, noise robust, DARPA RATS.

1. Introduction

Humans learn usually by building up on the knowledge they have already acquired. The organization of human learning process is based on gradually introducing harder and comparatively more difficult to learn concepts. This learning mechanism of humans has inspired a distinct category of *Curriculum Learning* (CL) based algorithms in Machine Learning [1]. CL based training strategies have been used to train Deep Neural Networks (DNN) with deeper architectures [2], formulate better algorithms for learning latent variable models [3], and improving noise robustness of automatic speech recognition (ASR) [4] among others. Motivated by the CL based Machine Learning algorithms, this study proposes CL based Probabilistic Linear Discriminant Analysis (CL-PLDA) to improve the performance of speaker identification (SID) systems in the presence of severe noise and distortions.

State-of-the-art text independent SID systems utilize i-Vector PLDA based frameworks [5, 6, 7, 8]. Speaker-specific attributes of an utterance can be compactly represented by i-Vectors [5]. I-vectors based techniques have also been used in language identification (LID) [9, 10, 11] and gender-identification [12].

Performance of SID systems can degrade rapidly if training, enrollment, or test data is corrupted by noise [13, 14, 15, 16, 17]. In [18], including noisy data in the estimation of PLDA parameters was found to offer improved SID performance in noisy and reverberant conditions. To improve noise robustness, using a combination of multiple PLDA backends, each trained for a different condition was proposed in [19]. In [20], noise robust Modified Hilbert Envelope Coefficients (MHEC) features were adopted for SID and LID tasks of the severely noisy and degraded DARPA RATS data [21]. Human auditory processing inspired Medium Duration sub-band Speech Amplitudes (MMeDuSA) features were introduced for noise robust SID in [22].

A Signal-to-Noise Ratio (SNR) invariant version of PLDA was proposed in [23], that operated by removing the SNR-specific information from speaker-specific information during training, thereby minimizing the effects of noise on the performance of PLDA back-end for SID. In [24], a mixture of PLDA models was trained by utilizing the SNR of the utterances. A Nearest Neighbor Discriminant Analysis (NDA) based approach for noise robust speaker recognition was proposed in [25], where NDA replaced Linear Discriminant Analysis (LDA) based pre-processing of i-Vectors before training the PLDA back-end.

In addition to using noise robust features, and improved PLDA back-ends that can better handle noisy data, speech enhancement and robust i-Vector extraction based methods have also been proposed to improve the performance of SID systems in noisy conditions. A DNN based speech enhancement method was used to extract clean speech which was then used for training the SID system in [26]. A DNN autoencoder was used for speech enhancement for SID in [27]. A Convolutional Neural Network (CNN) trained for ASR was used to compute the frame posteriors for training the i-Vector extractor matrix in [28]. Long Short Term Memory (LSTM) networks were used for speech enhacement for SID in [29].

Clean i-Vectors were estimated from the noisy i-Vectors using maximum a posteriori (MAP) estimation procedure assuming an additive model for noise in [30]. In [31], the noisy i-Vectors were cleaned by using an Minimum Mean Square Error (MMSE) based estimator.

Our proposed CL-PLDA algorithm operates by initializing the PLDA training EM algorithm with cleaner utterances, which in the CL paradigm are *easier* examples to learn. CL-PLDA operates by gradually adding noisier examples which amount to *difficult* examples for learning in the context of CL based algorithms. Compared to CL-PLDA, the regular PLDA operates by training on all the available data in each iteration of the training EM algorithm. We measure the performance of CL-PLDA on the noisy and severely distorted utterances of the DARPA RATS SID data, and show that CL-PLDA consistently outperforms PLDA for SID in noisy conditions using an evaluation set of approximately 3.2 million trials with enrollment/test sets of varying duration. We hypothesize that CL-PLDA is able to find better local minima during the training which results in improved estimation of parameters than the regular PLDA, resulting in better SID performance in noisy and degraded conditions.

This project was funded by AFRL under contract FA8750-15-1-0205 and partially by the University of Texas at Dallas from the Distinguished University Chair in Telecommunications Engineering held by J.H.L. Hansen.

http://dx.doi.org/10.21437/Interspeech.2017-1199

2. Speaker Identification using i-Vector PLDA based approach

2.1. i-Vector Extraction

I-Vectors offer compact representations of speech utterances while preserving the speaker-specific information [5] . In the i-Vector parlance, a speaker-specific GMM mean supervector M can be represented in terms of speaker and channel independent supervector m, a low rank *total variability matrix* T, and a vector w as

$$M = m + Tw. \tag{1}$$

In (1), w is a random vector with a standard normal distribution $N(0, I)$. The T matrix is learned using large amounts of training data. The i-Vector of an utterance are its coordinates in the *total variability space* (i.e. space spanned by the columns of T), extracted as the maximum a posteriori (MAP) point estimates of w given the utterance [5, 32].

2.2. SID using PLDA back-end

A PLDA model learns the within-class and across-class variabilities of a large labeled training set using an Expectation Maximization (EM) algorithm [6] . In this work, we have used Gaussian PLDA (G-PLDA) of the form described in [8]. Assuming R training utterances of a speaker, an entire collection of i-Vectors may be expressed as $\{\eta_r : r = 1, 2, ..., R\}$. In the G-PLDA formulation, an i-Vector of this collection can be represented as,

$$\eta_r = m + \Phi\beta + \epsilon_r. \tag{2}$$

In (2), m is a global offset, the columns of Φ constitute a basis for speaker subspace , β corresponds to the coordinates in speaker subspace, and ϵ_r is a Gaussian with zero mean and covariance Σ. The G-PLDA model parameters $\{m, \Phi, \Sigma\}$ are estimated using an EM algorithm on a large collection of speaker-labeled i-Vectors. Given a test utterance, the verification score can be computed using a closed form solution with the G-PLDA model as presented in [8]. A single speaker-verification trial needs access to the mean of speaker's enrollment i-Vectors, the test i-Vector, and the G-PLDA model parameters $\{m, \Phi, \Sigma\}$.

3. Curriculum Learning based Probabilistic Linear Discriminant Analysis (CL-PLDA)

CL-PLDA is inspired by CL based Machine Learning algorithms. The key idea in CL is to start a learning algorithm with simpler data and gradually introduce more complex data as the algorithm progresses. CL was originally introduced in [1], and the authors hypothesized that this human learning inspired approach could lead to better local minima for non-convex functions. We hypothesize that adopting a CL based approach leads to better estimation of PLDA model parameters, and can lead to improved SID performance.

3.1. CL-PLDA training

The PLDA training is done iteratively via an EM algorithm [6]. The algorithm begins by randomly initializing model parameters Φ and Σ (from Sec. 2) before the first iteration, and updates their values at the end of each iteration thereafter. Details about the update equations, and the training EM algorithm in general can be found in [6]. Each iteration of the regular PLDA training updates Φ and Σ based on all the available training data.

CL-PLDA operates differently than the regular PLDA in the manner it uses the training data to carry updates of the model parameters. Specifically, CL-PLDA begins with random initialization of Φ and Σ before the first iteration, but does not use all the available training data. Data is only introduced gradually to the training algorithm, beginning with easy data examples. As the number of iterations increases, comparatively more difficult data is included in the training set. When more data is introduced, CL-PLDA updates previous values of Φ and Σ, obtained from the simpler data (observed earlier) to account for the larger dataset that now has both simpler, and comparatively more difficult examples. This gradual inclusion of difficult data, as we shall see in Sec. 4, leads to better estimation of the CL-PLDA model parameters compared to the regular PLDA. Adopting CL-PLDA does introduce an extra hyper-parameter, the number of iterations n it operates on a particular subset of the training data before moving to a larger subset with more difficult examples added. However, the results of Sec. 4 will demonstrate that a very small n is sufficient for CL-PLDA in practice. The key requirement of CL-PLDA is a procedure to define *easy* and *difficult* training examples which effectively influences the whole training paradigm, and is introduced next.

3.2. Determining Easy and Difficult Data for CL-PLDA training

We treat SNR as a good measure of difficulty for a learning example for training CL-PLDA. For ASR applications, a similar strategy was suggested in [4]. We hypothesize that for a labeled factor analysis scheme such as PLDA, easy examples with high SNRs are more informative for the learning algorithm to estimate the latent variables underlying the examples. However, the model should also be able to account for the difficult, noisier data as well, and gradual introduction of noisier data (with lower SNRs) in the training leads to improved estimation of CL-PLDA model parameters that can explain both clean and noisy data better.

For the DARPA RATS data used in this paper, a clean source utterance was retransmitted through eight radio channels with different transmitter and receiver equipments. In addition to these retransmitted utterances being severely noisy, they are also highly degraded with shifts in pitch, non-linear intermittent fading, long time scale amplitude variation, ring modulation etc [21]. In Table 1 of [21], a measure of the intended signal to non-signal component of each radio channel is presented. This is obtained as the difference of Signal to Noise Distortion (SND) and Noise Distortion (ND). The SND value represents the transferred signal from transmitter to receiver together with the effects of noise and distortions induced by the particular channel. On the other hand, the ND value is representative of noise and distortions a particular radio channel. Thus, SND-ND values from Table 1 of [21], offer a good measure of the signal to non-signal component of each radio channel, and are used instead of SNR values in this work. Details of the evaluation protocols for SND and ND can be found in [21]. The following Alg. 1 presents the steps of our proposed CL-PLDA algorithm for noise robust speaker recognition.

4. Experiments, Results and Discussion

4.1. Training, and Evaluation Data

4.1.1. Training Data

For the experiments reported in this study, we used data from the DARPA RATS corpora. Specifically, the training set had approximately 56,000 utterance from 5913 speakers. Both training and evaluation sets were highly multilingual with utterances from Arabic, Dari, Farsi, Pashto, and Urdu. This included utter-

Algorithm 1: CL-PLDA for noise robust speaker recognition.

```
1  Partition the training data $d_{all}$ into $K$ progressively
     difficult and distinct subsets $d_1$, $d_2$, ..., $d_K$ based on a
     suitable difficulty criterion such as SNR or SND-ND.
2  Initialize current CL-PLDA parameters $\Phi_{curr}$ and
     $\Sigma_{curr}$ randomly.
3  Initialize an empty current dataset $d_{cur} \leftarrow \emptyset$
4  for k=1:K do
5      $d_{cur} \leftarrow d_{cur} \cup d_k$
6      for l=1:n do
7          Run 1 itertion of PLDA training with current
             values $\Phi_{curr}$, $\Sigma_{curr}$, and $d_{cur}$ to obtain
             updated values $\Phi_{upd}$, $\Sigma_{upd}$.
8          $\Phi_{cur} \leftarrow \Phi_{upd}$
9          $\Sigma_{cur} \leftarrow \Sigma_{upd}$
10     end
11 end
12 Use $\Phi_{cur}$, $\Sigma_{cur}$ as the CL-PLDA model parameters.
```

ances from 8 different channels (A-H) and clean src (source) utterances. Specifically, the training data comprised of utterances from the *train* set of the following Linguistic Data Consortium (LDC) releases: LDC2012E117, LDC2012E49, LDC2012E63, LDC2012E69 and LDC2012E85. The training set was used to train the UBM, i-Vector extractor matrix, PLDA and CL-PLDA based speaker verification back-ends.

4.1.2. Evaluation Data

To compare the performance of CL-PLDA against the regular PLDA, we used an enrollment set of approximately 16,000 utterances from 305 speakers. The enrollment and test sets were set aside from *dev* set of the same LDC releases as used for the training set. The test set had 10,617 utterances. We constructed the evaluation trials as the Cartesian product of the enrollment speakers (i.e. 305) times the number of test utterances (i.e. 10,617). Overall, the evaluation list had around 3.2 million trials which included 10,617 target trials.

4.2. PLDA and CL-PLDA based SID Systems

A standard Kaldi based recipe was used to train the SID system to extract the i-Vectors for both PLDA and CL-PLDA based SID systems (available with Kaldi examples, under sre10/v1) [33, 34]. We used 20-dim MFCC features with delta, and delta-delta for the SID system. A standard Voice Activity Detection (VAD) approach used in the original Kaldi SID recipe, was also used here. A full covariance UBM with 2048 components was trained using 4 iterations of the EM training. To train the i-Vector extractor matrix 5 iterations of the EM algorithm was used. 600 dimensional i-Vectors were extracted for the entire training, enrollment and test data.

4.2.1. PLDA baseline SID System

For the baseline PLDA SID back-end, the entire training set with utterances from all the 8 channels and clean source was used in every iteration of the training. Linear Disciriminant Analysis (LDA) was used to reduce the i-Vector dimensions from 600 to 200, before applying either PLDA or CL-PLDA. The baseline PLDA model was trained with 5 iterations of the EM algorithm.

4.2.2. CL-PLDA based SID System

As mentioned in Sec. 3, we used the SND-ND values from Table 1 of [21] for grading the difficulty of the channels. Since there are 8 different channels plus clean source utterances in training data, K (from Alg. 1) assumes a value of 9. Specifically, we arranged data according to SND-ND values for the channels. The order in which the data was included in CL-PLDA was src (source) utterances, followed by the utterances from channels B, E, A, F, C, D, G and H respectively. To train the CL-PLDA model, we used 5 iterations ($n = 5$) each time after adding a new channel to d_{cur} (from Alg. 1) for all the channels except the last channel.

We observed that after adding the data from the last channel (H) to d_{cur}, we just needed a single iteration of the EM training algorithm to obtain the final CL-PLDA model parameters that gave the most competitive SID performance, and more iterations caused a slight degradation compared to when using only a single iteration at the end. Thus, the *for* loop of Alg. 1 (line 6) was executed only for a single iteration after adding the last data-set from channel H for all experiments reported in this study. A possible reason for only a single iteration needed after adding the data from the last channel could be that CL-PLDA has already trained gradually on most of the training data, and so only a single update after adding more noisier and degraded data at the end is sufficient.

4.3. Results

Table 1 shows the results in terms of EER for the baseline PLDA system compared against our proposed CL-PLDA approach. The relative reduction in EER (%) obtained by our proposed CL-PLDA compared to the regular PLDA are also shown. We have shown the results for the enrollment-test sets of duration 3s-3s, 10s-10s, 30s-30s, 120s-120s. Single PLDA and CL-PLDA models, trained on the i-Vectors extracted from the entire duration of the utterances were used for evaluating the different duration enrollment-test sets.

Our proposed CL-PLDA outperforms PLDA across all the enrollment-test sets. For the smaller duration enrollment-test (3s-3s and 10s-10s) sets, CL-PLDA offers less reduction in EER compared to the larger duration enrollment-test evaluation sets (30s-30s and 120s-120s). For the 30s-30s and 120s-120s SID tasks, compared to PLDA, CL-PLDA reduces the EER by 10.20% and 13.95% respectively. The DARPA RATS SID task also used other performance metrics such as: Miss @ 4% False Alarm (FA), Miss @ 2.5% FA, FA @ 10% Miss among others [20, 22, 25, 28]. For brevity, we do not report all these metrics in this study. Instead, Fig. 1 shows the Detection Error

Table 1: *EER (%) for PLDA compared against CL-PLDA for enrollment-test sets of durations 3s-3s, 10s-10s, 30s-30s, 120s-120s. Single CL-PLDA, and PLDA back-end were trained on the i-Vectors extracted from the entire duration of the training utterances and used for all sets.*

Enrollment-test duration	EER (%)		Rel. reduction in EER (%) CL-PLDA vs PLDA
	PLDA	CL-PLDA	
3s-3s	37.89	37.28	+1.60
10s-10s	18.72	17.94	+4.16
30s-30s	8.13	7.30	+10.20
120s-120s	4.37	3.76	+13.95

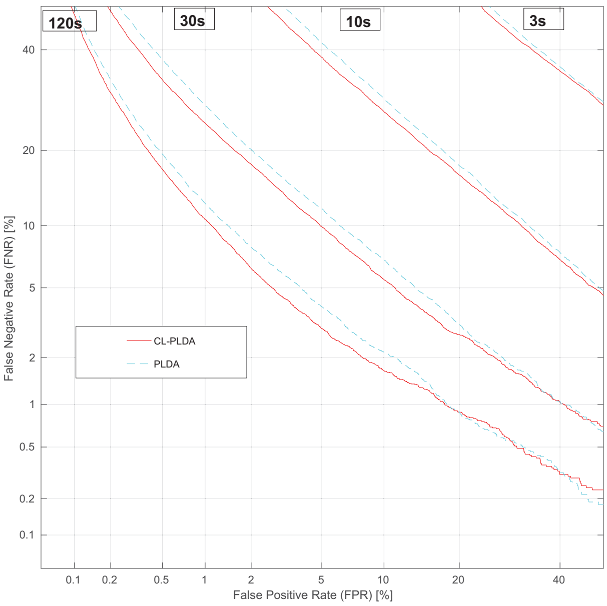

Figure 1: *DET plots for CL-PLDA vs PLDA for enrollment-test of duration (from left to right) 120s-120s, 30s-30s, 10s-10s, 3s-3s.*

Tradeoff (DET) plots for CL-PLDA vs PLDA SID systems for the different enrollment-test sets (from left to right) of duration 120s-120s, 30s-30s, 10s-10s, and 3s-3s. Clearly, CL-PLDA outperforms PLDA for all the 4 enrollment-test sets considered as can be inferred from the DET plots.

4.4. Discussion

From the results of Table 1, and Fig. 1, CL-PLDA offers significant improvements over PLDA for SID in severely noisy conditions. We hypothesize that CL-PLDA is able to find better local minima due to the gradual introduction of noisy and degraded data. This causes CL-PLDA to find more robust model parameters compared to the regular PLDA. While CL-PLDA consistently outperforms PLDA for all the enrollment-test cases considered, the improvements are comparatively smaller when the duration of enrollment-test sets is reduced. Here, it should be noted that the i-Vectors used for training the PLDA and CL-PLDA were extracted from the entire utterances (approximately 15 min. long). Duration mismatch between the data for PLDA training, and enrollment-test can lead to severe degradation in SID performance, which can be overcome to an extent by including a mixed training set of both long and short duration segments in the PLDA training [35]. Using such a training strategy for PLDA and CL-PLDA to test the shorter duration enrollment-test sets can lead to improved SID performance for both the approaches, and may cause CL-PLDA to offer comparatively similar reduction in EER as for the duration matched cases.

We also examined how the inclusion of data from the various channels impacted the performance of CL-PLDA. To this end, Table 2 show the EERs obtained for the various enrollment-test sets by including data from the different channels in CL-PLDA. For the results shown in each row, 5 iterations ($n = 5$) each were run for all the channels up to the corresponding channel of the row. For instance, for the results of the 3rd row (SRC + B + E), 5 iterations were run each for the training set with SRC, SRC ∪ B, SRC ∪ B ∪ E respectively. For each row, a single iteration using the entire data set was run at the end. Interestingly, by gradually introducing the data from only the source and 3 channels, B, E and A, CL-PLDA outperforms PLDA (from Table 1). It can also be observed that the best results are obtained by gradually introducing the training data for all the channels across all the enrollment-test sets except the 3s-3s condition. The abnormal behavior of 3s-3s is most likely due to the duration mismatch mentioned previously.

Table 2: *EER (%) obtained by including the different channels in CL-PLDA for enrollment-test sets of duration 120s-120s, 30s-30s, 10s-10s and 3s-3s.*

Distinct Channels in CL-PLDA	EER (%)			
	120s-120s	30s-30s	10s-10s	3s-3s
SRC	24.79	29.53	35.00	44.29
+B	9.06	14.31	24.85	40.76
+E	4.56	8.76	19.65	**37.08**
+A	3.92	7.79	18.39	37.84
+F	3.78	7.33	18.06	37.45
+C	3.86	7.38	18.03	37.27
+D	3.83	7.37	17.75	37.32
+G	**3.76**	**7.30**	**17.94**	37.28

Table 3: *Effect of varying n, the number of iterations per channel in CL-PLDA based SID on the enrollment-test sets of duration 120s-120s, 30s-30s, 10s-10s, and 3s-3s.*

n (iter. per channel)	EER (%)			
	120s-120s	30s-30s	10s-10s	3s-3s
1	3.92	7.71	17.91	36.72
2	3.75	7.35	18.04	37.29
3	3.74	7.30	18.02	37.28
4	3.75	7.30	17.98	37.28
5	3.76	7.30	17.94	37.28

Table 3 shows the effect of varying n (i.e number of iterations per channel) on the SID performance for the enrollment-test sets of various durations. As is evident, even by running only a single iteration per channel ($n = 1$), CL-PLDA is able to obtain very competitive results, and outperforms the baseline PLDA results shown in Table 1. There is not much variation in the SID performance for values of n greater than 3. Interestingly, the best EER for 3s-3s trials is obtained for $n = 1$, which may again be due to the duration mismatch between the training and enrollment-test i-Vectors.

5. Conclusions

We presented a novel Curriculum Learning based PLDA (CL-PLDA) algorithm to improve speaker recognition in noisy conditions. CL-PLDA operates by initiating the training with cleaner data and gradually adding noisier data as the training progresses. We tested CL-PLDA for SID on the noisy and severely degraded data from DARPA RATS, where CL-PLDA consistently outperformed PLDA for various enrollment-test sets using an evaluation list with approximately 3.2 million trials. On the enrollment-test sets of duration 30s-30s and 120-120s, our proposed CL-PLDA reduced the EER by 10.20% and 13.95% respectively, compared to PLDA. Future work will explore using CL-PLDA with duration mismatch compensation strategies to improve speaker recognition for very short duration utterances in noisy conditions. Further, the use of SNR as a difficulty criterion will be investigated.

6. References

[1] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," *Proceedings of the 26th annual International Conference on Machine Learning*, pp. 41–48, 2009.

[2] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," *arXiv preprint arXiv:1512.02595*, 2015.

[3] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," *Advances in Neural Information Processing Systems*, pp. 1189–1197, 2010.

[4] S. Braun, D. Neil, and S.-C. Liu, "A curriculum learning method for improved noise robustness in automatic speech recognition," *arXiv preprint arXiv:1606.06864*, 2016.

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

[6] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," *IEEE ICCV-2007*, pp. 1–8, 2007.

[7] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." *Odyssey*, 2010.

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

[9] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language recognition via i-Vectors and dimensionality reduction." *ISCA INTERSPEECH*, pp. 857–860, 2011.

[10] D. Martınez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language recognition in ivectors space," *Proceedings of Interspeech, Firenze, Italy*, pp. 861–864, 2011.

[11] S. Ranjan, C. Yu, C. Zhang, F. Kelly, and J. H. Hansen, "Language recognition using deep neural networks with very limited training data," *IEEE ICASSP 2016*, 2016.

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

[13] M. I. Mandasari, M. McLaren, and D. A. van Leeuwen, "The effect of noise on modern automatic speaker recognition systems," *IEEE ICASSP 2012*, pp. 4249–4252, 2012.

[14] R. Togneri and D. Pullella, "An overview of speaker identification: Accuracy and robustness issues," *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 23–61, 2011.

[15] J. Ming, T. J. Hazen, J. R. Glass, and D. A. Reynolds, "Robust speaker recognition in noisy conditions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1711–1723, 2007.

[16] L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matejka, O. Plchot, and N. Scheffer, "Promoting robustness for speaker modeling in the community: the PRISM evaluation set," *Proceedings of NIST 2011 workshop*, 2011.

[17] 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.

[18] Y. Lei, L. Burget, L. Ferrer, M. Graciarena, and N. Scheffer, "Towards noise-robust speaker recognition using probabilistic linear discriminant analysis," *IEEE ICASSP 2012*, pp. 4253–4256, 2012.

[19] D. Garcia-Romero, X. Zhou, and C. Y. Espy-Wilson, "Multicondition training of Gaussian PLDA models in i-vector space for noise and reverberation robust speaker recognition," *IEEE ICASSP 2012*, pp. 4257–4260, 2012.

[20] S. O. Sadjadi and J. H. L. Hansen, "Mean hilbert envelope coefficients (mhec) for robust speaker and language identification," *speech communication*, vol. 72, pp. 138–148, 2015.

[21] K. Walker and S. Strassel, "The RATS radio traffic collection system," *Odyssey*, 2012.

[22] V. Mitra, M. McLaren, H. Franco, M. Graciarena, and N. Scheffer, "Modulation features for noise robust speaker identification." *ISCA INTERSPEECH*, pp. 3703–3707, 2013.

[23] N. Li and M.-W. Mak, "Snr-invariant PLDA modeling in nonparametric subspace for robust speaker verification," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 10, pp. 1648–1659, 2015.

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

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

[26] J. Chang and D. Wang, "Robust speaker recognition based on DNN/I-Vectors and speech separation," *IEEE ICASSP 2017*, 2017.

[27] O. Plchot, L. Burget, H. Aronowitz, and P. Matějka, "Audio enhancing with DNN autoencoder for speaker recognition," *IEEE ICASSP 2016*, pp. 5090–5094, 2016.

[28] M. McLaren, Y. Lei, N. Scheffer, and L. Ferrer, "Application of convolutional neural networks to speaker recognition in noisy conditions." *ISCA INTERSPEECH*, pp. 686–690, 2014.

[29] M. Kolbœk, Z.-H. Tan, and J. Jensen, "Speech enhancement using long short-term memory based recurrent neural networks for noise robust speaker verification," *Spoken Language Technology Workshop (SLT), 2016 IEEE*, pp. 305–311, 2016.

[30] W. B. Kheder, D. Matrouf, J.-F. Bonastre, M. Ajili, and P.-M. Bousquet, "Additive noise compensation in the i-vector space for speaker recognition," *IEEE ICASSP 2015*, pp. 4190–4194, 2015.

[31] W. B. Kheder, D. Matrouf, M. Ajili, and J.-F. Bonastre, "Probabilistic approach using joint clean and noisy i-vectors modeling for speaker recognition," *Interspeech 2016*, pp. 3638–3642, 2016.

[32] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 3, pp. 345–354, 2005.

[33] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," *IEEE 2011 workshop on automatic speech recognition and understanding*, 2011.

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

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