

An Iterative Framework for Unsupervised Learning in the PLDA based Speaker Verification

Wenbo Liu^{1,2}, Zhiding Yu², Ming Li^{1,3}

¹SYSU-CMU Joint Institute of Engineering, Sun Yat-Sen University

²Department of Electrical and Computer Engineering, Carnegie Mellon University

³SYSU-CMU Shunde International Joint Research Institute

{wenboliu, yzhiding}@andrew.cmu.edu, mingli1@cmu.edu

Abstract

We present an iterative and unsupervised learning approach for the speaker verification task. In conventional speaker verification, Probabilistic Linear Discriminant Analysis (PLDA) has been widely used as a supervised backend. However, PLDA requires fully labeled training data, which is often difficult to obtain in reality. To automatically retrieve the speaker labels of unlabeled training data, we propose to use the Affinity Propagation (AP) - a clustering method that takes pairwise data similarity as input - to generate the labels for the PLDA modeling. We further propose an iterative refinement strategy that incrementally updates the similarity input of the AP clustering with the previous iteration's PLDA scoring outputs. Moreover, we evaluate the performance of different PLDA scoring methods for the multiple enrollment task and show that the generalized hypothesis testing achieves the best results. Experiments were conducted on the NIST SRE 2010 and the 2014 i-vector challenge database. The results show that our proposed iterative and unsupervised PLDA model learning approach outperformed the cosine similarity baseline by 35% relatively.

Index Terms: Speaker Verification, I-Vector, Probabilistic Linear Discriminant Analysis, Affinity Propagation

1. Introduction

The goal of speaker verification is to automatically verify the claimed speaker identity given a segment of speech. Recently, total variability i-vector modeling has gained significant attention in speaker verification due to its excellent performance, compact representation and small model size [1]. In this framework, zero-order and first-order Baum-Welch statistics are first calculated by projecting the Mel-frequency cepstral coefficients (MFCC) features onto Gaussian Mixture Model (GMM) components. Then, factor analysis is adopted on the supervectors to generate a low dimensional total variability space which jointly models language, speaker and channel variabilities all together [2]. Finally, within this i-vector space, variability compensation methods, such as Within-Class Covariance Normalization (WC-CN) [3], Linear Discriminative Analysis (LDA) and Nuisance Attribute Projection (NAP) [4] are performed to reduce the variability for backend modelling. The commonly used backend scoring methods are cosine similarity [1], Support Vector Machine (SVM) [5], sparse representation [6], Probabilistic Linear Discriminant Analysis (PLDA) [7, 8], deep belief networks [5],

This research is funded in part by NSFC China, CMU-SYSU Collaborative Innovation Research Center and the SYSU-CMU Shunde International Joint Research Institute.

etc. Among the aforementioned techniques, PLDA is widely adopted and considered as the state-of-the-art backend modeling approach [8, 9, 10, 11, 12, 13, 14].

Like other backend methods, PLDA also requires speaker label input for the supervised learning. In reality however, large quantities of data are often not sufficiently labeled, and the labor cost of labeling can be high. Two class of methods were proposed to address the problem. The first one is domain adaptation. Models originally built from the labeled out-of-domain data are adapted to the target domain with unlabeled in-domain data [15]. The second class of methods is semi-supervised learning, which can be applied on the database with a small amount of labeled data and large amounts of unlabeled data.

But domain adaptation and semi-supervised learning still need some amount of labeled data. The trend today is that we have some very exciting opportunities to collect large amounts of unlabeled speech data, thus giving rise to "data deluge". The problem we address here is to utilize these large scale unlabeled data to train a high quality PLDA model. We present a fully unsupervised algorithm with an iterative framework.

The proposed iterative framework consists two steps: clustering and scoring. We first perform clustering on the unlabeled development data to estimate the labels and train a PLDA model. We then apply the PLDA model to score the development data for pairwise similarity that is incrementally fused with previous similarity to form new clustering input in next round. This process is repeated for several iterations. To some extent, the iterative incremental update framework shares certain analogy to the idea of AdaBoost and self-paced learning, in the sense that each round of PLDA serves as a probabilistic interpretation of the data and a gradual refinement of the similarity. Finally the learned PLDA model generate scores on target/test data.

There exists a variety of clustering methods such as K-means and spectral clustering [16]. We choose affinity propagation (AP) [17] because it takes pairwise similarity as input which fit the PLDA hypothesis testing based scoring. Nevertheless, other clustering methods may also be applied here.

We also consider the case where there are multiple target or test i-vectors in each trial. We extend the conventional hypothesis testing based PLDA scoring to a generalized form that can perform scoring with arbitrary number of targets and tests.

2. The Proposed Method

The overview of the proposed iterative framework for fully unsupervised learning is shown in Fig.1 . This framework mainly consists of an iterative process of AP clustering and PLDA training/scoring. We first describe the i-vector PLDA baseline

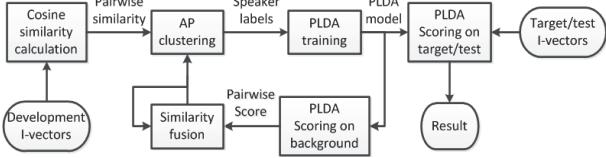


Figure 1: The overview of the proposed framework with iterative and unsupervised PLDA model learning.

and the simplified version we used in Sec 2.1. Then introduce the affinity propagation clustering in Sec 2.2. Our main contribution is presented in Sec 2.3 and Sec 2.4.

2.1. I-vector and PLDA baseline

An i-vector extractor is a system that maps a sequence of vectors (typically cepstral coefficients) from a speech utterance to a fixed-length vector. For the NIST experiment, we adopt the generalized i-vector framework with phonetic tokenizations and tandem features in [18]. This generalized framework extends the choices of tokens and features for statistics calculation while keeps the factor analysis and subsequent backend modeling the same way as the conventional i-vector method. Other types of generalized i-vectors have also been proposed recently in [19, 20]. Phonetic aware generalized i-vector extraction is so far the state-of-the-art front-end system on the NIST speaker verification task [19, 20, 18]. For the i-vector challenge in our experiment, the i-vectors have already been provided.

PLDA is a standard and widely used technique in speaker verification. In this work, we adopt the modified Gaussian PLDA (G-PLDA) proposed in [9]. Instead of decomposing each i-vector into the sum of a global mean component, a speaker subspace component, a channel subspace component and a residual term [7], the method directly models the channel component and the residual with a full covariance matrix for simplification and performance [9].

2.2. The Affinity Propagation clustering

Given data without any label, clustering is probably the most direct and effective way to generate training labels for the backend system. In this paper, we use Affinity Propagation [17] to perform clustering. In statistics, AP is a clustering algorithm based on the concept of “message passing” between data points. Like k-medoids, AP finds “exemplars” - members of the input set that are representative of clusters. But unlike many clustering algorithms, it does not require the number of clusters to be determined or estimated before running the algorithm. This brings much convenience in our tasks where the number of true clusters can be different and are often not easy to be determined. The only parameter we need to set is the clustering iteration number, which is also carefully studied in this work.

2.3. Iterative PLDA on clustering

AP takes pairwise similarity as clustering input, which motivates our iterative learning framework for fully unsupervised speaker verification. This section describes the framework.

- Step 1: Initialize the pairwise similarity input for AP using the cosine similarity. Given two normalized i-vectors η_i and η_j , the cosine similarity is defined as:

$$\text{similarity}(i, j) = \frac{\eta_i \cdot \eta_j}{\|\eta_i\| \|\eta_j\|} \quad (1)$$

- Step 2: Given pairwise similarity, perform AP clustering on development data to generate cluster labels.
- Step 3: Train the PLDA based on the development data and the unsupervised cluster labels.
- Step 4: perform one-to-one hypothesis testing for pairwise development data. This results in a score that measures the pairwise development data similarity in somewhat more refined way. Normalize the score:

$$\text{score}(i, j) = \frac{(\text{score}(i, j) - \min(\text{score}))}{(\max(\text{score}) - \min(\text{score}))} \quad (2)$$

- Step 5: The obtained score in Step 4 is used to update the data similarity. We formulate the similarity update as an incremental process with a updating factor β :

$$\begin{aligned} \text{similarity}(i, j)^t &= \beta * \text{similarity}(i, j)^{t-1} \\ &\quad + (1 - \beta) * \text{score}(i, j)^{t-1}, \end{aligned} \quad (3)$$

where $\text{similarity}(i, j)^t$ is the similarity between the i th and j th development data at the t th round. $\text{similarity}(i, j)^0$ is the cosine similarity from Step 1.

- Repeat Step 2 to Step 5 with a limited number. Generate the final scores on test data after a number of iterations.

The incremental update shares certain analogy to Adaboost in which classification is boosted with a number of weighted “weak learners”. If one takes a very aggressive step with $\beta = 0$, the performance of the algorithm would be much worse.

2.4. Multiple-Enrollment PLDA

The PLDA model assumes that the samples with the same identity are generated by the same distribution, while the samples with different identities are generated by different independent distributions. Consider the case with only one target η_i and one test η_j given in a trial, one could perform two alternative hypothesis testings by assuming the samples are either generated from the same distribution, or from independent ones:

$$\text{score} = \log \frac{p(\eta_i, \eta_j | \mathcal{H}_s)}{p(\eta_i | \mathcal{H}_d)p(\eta_j | \mathcal{H}_d)}. \quad (4)$$

Multiple-enrollment of targets and tests can greatly improve the verification accuracy. Many conventional literatures often either heuristically take the average of PLDA output scores on single-enrollment tests, or take the means of both target and test samples to perform one single-enrollment test. Here we generalize the hypothesis testing into multiple-enrollment using a fully probabilistic approach. We happened to notice that very recently [14, 11] proposed similar methods for multiple-enrollment. Here we show an alternative, simpler derivation with matrix block-wise operations.

Let N_1 and N_2 denote the number of targets and tests. $N = N_1 + N_2$. Also let η_i denote any target or test sample i-vector, we have the following definition of the score:

$$\text{score} = \log \frac{p(\eta_1, \dots, \eta_{N_1}, \eta_{N_1+1}, \dots, \eta_N | \mathcal{H}_s)}{p(\eta_1, \dots, \eta_{N_1} | \mathcal{H}_d)p(\eta_{N_1+1}, \dots, \eta_N | \mathcal{H}_d)}, \quad (5)$$

where $\{\eta_1, \dots, \eta_{N_1}\}$ are targets and $\{\eta_{N_1+1}, \dots, \eta_N\}$ are test i-vectors. Since we adopt the modified G-PLDA, each i-vector can be represented as:

$$\eta_i = m + \Phi\beta + \epsilon_i. \quad (6)$$

Without loss of generality, assume $m = 0$. And note that β and ϵ are independent [9]. Since $E[A][B] = E[A]E[B]$ with A and B being independent, one has:

$$\Sigma_{tot} \triangleq E[\eta_i \eta_i^\top] = \Phi E[\beta \beta^\top] \Phi^\top + E[\epsilon_i \epsilon_i^\top] = \Phi \Phi^\top + \Sigma. \quad (7)$$

When $\eta_i \eta_j$ are generated from the same distribution, one has:

$$\Sigma_{ac} \triangleq E[\eta_i \eta_j^\top] = \Phi \Phi^\top. \quad (8)$$

When $\eta_i \eta_j$ are generated from different distributions, the covariance is simply 0. The score can be further computed as:

$$\begin{aligned} score &= \log \mathcal{N} \left(\begin{bmatrix} \eta_1 \\ \vdots \\ \eta_N \end{bmatrix}; \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \right) \\ &\quad - \log \mathcal{N} \left(\begin{bmatrix} \eta_1 \\ \vdots \\ \eta_N \end{bmatrix}; \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} \right) \end{aligned} \quad (9)$$

where A_{11} has $N_1 \times N_1$ blocks and A_{12} has $N_1 \times N_2$ blocks:

$$A_{11} \triangleq \begin{bmatrix} \Sigma_{tot} & \Sigma_{ac} & \dots & \Sigma_{ac} \\ \Sigma_{ac} & \Sigma_{tot} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \Sigma_{ac} \\ \Sigma_{ac} & \dots & \Sigma_{ac} & \Sigma_{tot} \end{bmatrix} \quad A_{12} \triangleq \begin{bmatrix} \Sigma_{ac} & \dots & \Sigma_{ac} \\ \vdots & \ddots & \vdots \\ \Sigma_{ac} & \dots & \Sigma_{ac} \end{bmatrix}$$

A_{22} is defined identical to A_{11} with $N_2 \times N_2$ blocks and A_{21} is defined as A_{12}^\top . With the Schur complement one can obtain the precision matrices in Eq. (9) with blockwise inversion. The final multiple-enrollment score can therefore be obtained as:

$$score = \hat{\eta}^\top Q_1 \hat{\eta} + \tilde{\eta}^\top Q_2 \tilde{\eta} + \hat{\eta}^\top P_1 \hat{\eta} + \tilde{\eta}^\top P_2 \tilde{\eta}, \quad (10)$$

where $\hat{\eta}$ and $\tilde{\eta}$ represent the concatenated target samples and test samples respectively:

$$\left\{ \begin{array}{l} Q_1 = A_{11}^{-1} - (A_{11} - A_{12} A_{22}^{-1} A_{12}^\top)^{-1} \\ Q_2 = A_{22}^{-1} - (A_{22} - A_{12}^\top A_{11}^{-1} A_{12})^{-1} \\ P_1 = A_{11}^{-1} A_{12} (A_{22} - A_{12}^\top A_{11}^{-1} A_{12})^{-1} \\ P_2 = A_{22}^{-1} A_{12}^\top (A_{11} - A_{12} A_{22}^{-1} A_{12}^\top)^{-1} \end{array} \right. \quad (11)$$

3. Experiment result

3.1. DATA DESCRIPTION

We use two databases in our experiment, namely the NIST speaker recognition evaluation (SRE) 2010 database and the I-vector challenge database. We first performed experiments on the NIST SRE 2010 corpus. Our focus is the female part of the common condition 5 (a subset of tel-tel) in the core task. We used equal error rate (EER), the normalized old minimum decision cost value (norm old minDCF) and the norm new minDCF as the metrics for evaluation. For cepstral feature extraction, a 25ms Hamming window with 10ms shifts was adopted. Each utterance was converted into a sequence of 36-dimensional feature vectors, each consisting of 18 MFCC coefficients and their first derivatives. We employed the Czech phoneme recognizer [21] to perform the voice activity detection (VAD) by simply dropping all frames that are decoded as silence or speaker noises. Feature warping is applied to mitigate variabilities. Our generalized i-vector framework is the same as [18] and we used the hybrid-GMM-hybrid setup which concatenates the MFCC and the phonetic tandem feature at the feature level. The training data for NIST 2010 task include Switchboard II part1 to part3,

NIST SRE 2004, 2005, 2006 and 2008 corpora on the telephone channel. The gender-dependent GMM UBMs consist of 1024 mixture components and the English tandem feature dimension is 52. The i-vector dimension and speaker subspace rank in the PLDA model are 500 and 150.

The second database we used is the i-vector challenge database [22]. I-vector machine learning challenge is a special speaker recognition challenge coordinated by NIST. The challenge is based on the i-vector paradigm widely used by state-of-the-art speaker recognition systems. Only i-vectors are provided. This database consists of three subsets, namely development set, validation set and test set. Only validation set provides the speaker labels. Therefore, we perform our iterative unsupervised PLDA model learning on the development set and evaluate the performance on the validation set. Since there are 5 i-vectors associating with each speaker, we construct our own evaluation trials with both single enrollment and multiple enrollment conditions. For the single enrollment task, each i-vector is scored on every i-vector in the validation set except itself. For the multiple enrollment task, the first 4 i-vectors in the alphabetical order for each speaker is considered as the target i-vectors, and the remaining 5th i-vector for each speaker is used for testing. The total trials number for the single enrollment and multiple enrollment tasks are 21,317,185 and 8,522,956.

We report our performance in terms of EER, norm old minDCF (08cost) and norm old minDCF (10cost), respectively. For all experiments, β is set to 0.8.

3.2. Result on the NIST dataset

We first compare the performance of the supervised PLDA baseline, the cosine similarity scoring baseline and our iterative unsupervised system with standard AP clustering.

Table 1: Result on the NIST SRE 2010 core condition 5 female part task, the AP clustering stopped at iteration 400

	EER	08 cost	10 cost
PLDA (supervised)	1.69%	0.1045	0.198
Cosine similarity	7.35%	0.317	0.672
AP + PLDA (round 1)	4.79%	0.229	0.53
AP + PLDA (round 2)	4.52%	0.226	0.518
AP + PLDA (round 3)	4.51%	0.220	0.492
AP + PLDA (round 4)	4.51%	0.215	0.464
AP + PLDA (round 5)	5.06%	0.216	0.453

In Table 1, we can see that the PLDA supervised training outperformed any other unsupervised method which is reasonable due to the additional label information. The AP + PLDA iterative system (round 4) outperformed the Cosine similarity baseline ($7.35\% \rightarrow 4.51\%$ EER) significantly which supports our claim that clustering and PLDA learning could be integrated together as an unsupervised PLDA learning method. Results also shows that the EER and cost values keep decreasing with the increase of round number, which demonstrates the effectiveness of the proposed iterative unsupervised learning approach.

Table 2 shows the results of round 4 AP+PLDA iterative unsupervised learning with different AP clustering iteration numbers. From table 2 we can observe that the AP clustering iteration number also plays an important role. Larger amount of iterations (400) did not improve the performance which might be due to over clustering where the clustering tends to create more clusters than needed. One speaker may be clustered into mul-

Table 2: Result on the NIST SRE 2010 core condition 5 female part task with AP+PLDA iterative unsupervised learning round 4. The AP clustering iterations impact the performance.

	EER	08 cost	10 cost
Baseline (supervised)	1.69%	0.1045	0.198
AP(32 iterations) + PLDA	4.52%	0.217	0.495
AP(38 iterations) + PLDA	4.54%	0.213	0.54
AP(64 iterations) + PLDA	4.26%	0.202	0.485
AP(100 iterations) + PLDA	4.24%	0.218	0.498
AP(150 iterations) + PLDA	3.92%	0.177	0.452
AP(200 iterations) + PLDA	4.20%	0.183	0.420
AP(250 iterations) + PLDA	3.78%	0.184	0.470
AP(400 iterations) + PLDA	4.51%	0.215	0.464

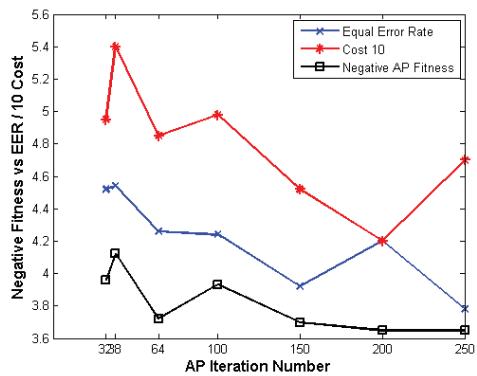


Figure 2: Relationship of EER and Fitness.

multiple cluster centers due to the channel variabilities. Therefore early stop helps. From Table 2, we can see that 200 or 250 iterations gives the best performance. Comparing the EER / Cost and the negative AP clustering fitness ($Const - Fitness$) curve in Fig3¹, we can find that the performance positively correlates to negative fitness. Higher fitness gives better performance until converge. This shows the fitness curve in AP clustering is a good indicator of how to set the iteration number for each round.

3.3. Result on I-vector Challenge database

We also evaluate our iterative unsupervised learning framework on i-vector challenge database as described in Sec 3.1. We first show the results of the single enrollment task in which the validation data itself (without the labels) is used for AP clustering and PLDA model learning. From Table 3, we can find out that the proposed iterative learning improves the performance by more than 20% relatively. the performance becomes better with more iterations ($5.67\% \rightarrow 4.18\% EER$). Fig. (3) shows the improvement of the error curve as iteration accumulates.

Table 4 shows the results of different PLDA scoring methods on the multiple enrollment task in i-vector Challenge database. The AP clustering and PLDA model learning are based on the unlabeled development data and results are evaluated on the validation set. Both system 1 and 3 average all the target i-vectors from a single trial into a single mean i-vector and then perform scoring the same way as in the single enrollment task. Instead, system 2 calculates the scores for each test

¹Note we have scaled each input to approximately the same range

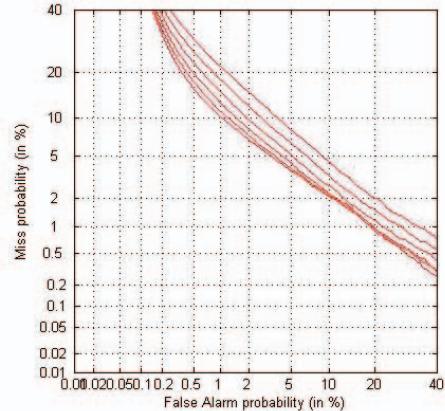


Figure 3: Improvement of the error curve over iterations.

Table 3: Result on the i-vector Challenge database single enrollment task, the AP clustering stopped at iteration 60

	EER
Cosine similarity	6.52%
AP + PLDA (round 1)	5.67%
AP + PLDA (round 2)	5.02%
AP + PLDA (round 3)	4.60%
AP + PLDA (round 4)	4.28%
AP + PLDA (round 5)	4.18%

i-vector on all the target i-vectors in the trial and then averages the scores as the final output for this trial. In this work, we propose a generalized hypothesis testing framework (system 4) for multiple enrollment or testing tasks. The results in Table 4 demonstrates the effectiveness of our extended PLDA multiple enrollment scoring approach described in Sec 2.4.

4. Conclusions

In this paper, we proposed an iterative learning framework for fully unsupervised Speaker Verification. Our framework is not limited to AP, but applicable to any method that takes pairwise similarity measure as input. We also gave an alternative, simpler derivation with matrix block operations on the joint hypothesis testing score of multiple enrollment. Experimental results on both NIST speaker verification and the I-Vector Challenge showed that our proposed framework can indeed improve the speaker verification systems based on fully unsupervised learning. Note that our system outperforms the cosine distance baseline considerably while significantly reducing the gap to supervised PLDA, showing the value of this work.

Table 4: Result on the i-vector Challenge database multiple enrollment task with different PLDA scoring methods. Round 1 AP+PLDA, the AP clustering stopped at iteration 60

ID	Methods	EER
1	Cosine similarity baseline	2.68%
2	AP + PLDA, score averaging	2.37%
3	AP + PLDA, i-vector averaging	2.22%
4	AP + PLDA, generalized hypothesis testing	2.07%

5. References

- [1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.
- [2] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, “Language recognition via i-vectors and dimensionality reduction,” in *Proc. INTERSPEECH*, 2011, pp. 857–860.
- [3] A. Hatch, S. Kajarekar, and A. Stolcke, “Within-class covariance normalization for SVM-based speaker recognition,” in *Proc. INTERSPEECH*, vol. 4, 2006, pp. 1471–1474.
- [4] W. Campbell, D. Sturim, and D. Reynolds, “Support vector machines using gmm supervectors for speaker verification,” *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, 2006.
- [5] S. Cumani, N. Brummer, L. Burget, and P. Laface, “Fast discriminative speaker verification in the i-vector space,” in *Proc. ICASSP*. IEEE, 2011, pp. 4852–4855.
- [6] M. Li, X. Zhang, Y. Yan, and S. Narayanan, “Speaker verification using sparse representations on total variability i-vectors,” in *Proc. INTERSPEECH*, 2011, pp. 4548–4551.
- [7] S. Prince and J. Elder, “Probabilistic linear discriminant analysis for inferences about identity,” in *Proc. ICCV*, 2007, pp. 1–8.
- [8] P. Matejka, O. Glemek, F. Castaldo, M. Alam, O. Plchot, P. Kenny, L. Burget, and J. Cernocky, “Full-covariance ubm and heavy-tailed plda in i-vector speaker verification,” in *Proc. ICASSP*, 2011, pp. 4828–4831.
- [9] D. Garcia-Romero and C. Y. Espy-Wilson, “Analysis of i-vector length normalization in speaker recognition systems,” in *Proc. INTERSPEECH*, 2011, pp. 249–252.
- [10] R. Saeidi, K.-A. Lee, T. Kinnunen, T. Hasan, B. Fauve, P.-M. Bousquet, E. Khouri, P. Sordo Martinez, J. M. K. Kua, C. You *et al.*, “I4u submission to nist sre 2012: A large-scale collaborative effort for noise-robust speaker verification,” 2013.
- [11] P. Kenny, T. Staflakis, P. Ouellet, M. Alam, P. Dumouchel *et al.*, “Plda for speaker verification with utterances of arbitrary duration,” in *Proc. ICASSP*. IEEE, 2013, pp. 7649–7653.
- [12] Y. Lei, L. Burget, L. Ferrer, M. Graciarena, and N. Scheffer, “Towards noise-robust speaker recognition using probabilistic linear discriminant analysis,” in *Proc. ICASSP*. IEEE, 2012, pp. 4253–4256.
- [13] G. Liu, T. Hasan, H. Boril, and J. H. Hansen, “An investigation on back-end for speaker recognition in multi-session enrollment,” in *Proc. ICASSP*. IEEE, 2013, pp. 7755–7759.
- [14] P. Rajan, A. Afanasyev, V. Hautamäki, and T. Kinnunen, “From single to multiple enrollment i-vectors: Practical plda scoring variants for speaker verification,” *Digital Signal Processing*, 2014.
- [15] D. Garcia-Romero and A. McCree, “Supervised domain adaptation for i-vector based speaker recognition,” in *Proc. ICASSP*. IEEE, 2014, pp. 4075–4079.
- [16] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, “On spectral clustering: Analysis and an algorithm,” *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.
- [17] B. J. Frey and D. Dueck, “Clustering by passing messages between data points,” *science*, vol. 315, no. 5814, pp. 972–976, 2007.
- [18] M. Li and W. Liu, “Speaker verification and spoken language identification using a generalized i-vector framework with phonetic tokenization and tandem features,” in *Proc. INTERSPEECH*, 2014.
- [19] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, “A novel scheme for speaker recognition using a phonetically-aware deep neural network,” in *Proc. ICASSP*, 2014.
- [20] L. F. D’Haro, R. Cordoba, C. Salamea, and J. D. Echeverry, “Extended phone log-likelihood ratio features and acoustic-based i-vectors for language recognition,” in *Proc. ICASSP*. IEEE, 2014, pp. 5379–5383.
- [21] P. Schwarz, P. Matejka, and J. Cernocky, “Hierarchical structures of neural networks for phoneme,” in *Proc. ICASSP*, 2006, pp. 325–328, software available at <http://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context>.
- [22] NIST, “Nist i-vector machine learning challenge,” <http://ivectorchallenge.nist.gov/>, 2014.