Factor analysis based speaker verification using ASR

Hang Su[1,2], *Steven Wegmann*[1,3]

[1] International Computer Science Institute, Berkeley, California, US
[2] Dept. of Electrical Engineering & Computer Science, University of California, Berkeley, CA, USA
[3] Semantic Machines Inc, Berkeley, CA, USA

suhang3240@gmail.com swegmann@icsi.berkeley.edu

Abstract

In this paper, we propose to improve speaker verification performance by importing better posterior statistics from acoustic models trained for Automatic Speech Recognition (ASR). This approach aims to introduce state-of-the-art techniques in ASR to speaker verification task. We compare statistics collected from several ASR systems, and show that those collected from deep neural networks (DNN) trained with fMLLR features can effectively reduce equal error rate (EER) by more than 30% on NIST SRE 2010 task, compared with those DNN trained without feature transformations. We also present derivation of factor analysis using variational Bayes inference, and illustrate implementation details of factor analysis and probabilistic linear discriminant analysis (PLDA) in Kaldi recognition toolkit.

Index Terms: Speaker verification / identification, speech recognition, deep neural networks, kaldi

1. Introduction

Factor analysis [1, 2] has become a dominant methodology for speaker verification in the last few years. This model is trained to learn a low-dimensional subspace from high-dimensional Gaussian Mixture Model (GMM) supervector space. The projected low-dimensional vector is used to represent different identities, thus denoted as i-vector (identity vector). I-vectors are usually transformed using a probabilistic linear discriminant analysis (PLDA) model to produce verification scores [3], which could be seen as a score normalization step. It has been shown that this could improve speaker verification performance significantly.

While deep learning has been successfully used for acoustic modeling in speech recognition [4–6], it is a harder task to apply it to speaker verification. The reason for this is two-fold: 1. speaker verification is not a standard classification task where targets are defined during training – unknown speakers may show up during enrollment phase or testing phase; 2. training data for speakers models are limited, e.g. each recording may only be used to extract one i-vector for the speaker. However, a novel scheme is proposed in [7] where DNN is introduced to perform frame alignment in GMM supervector generation. This approach is shown to be effective for speaker verification and a 30% relative reduction on equal error rate (EER) was achieved. The authors reasoned that this approach allows system to factor out content information and make use of phonetic content. On the other hand, authors in [8] use bottleneck features extracted from a ASR deep neural network to do speaker and language recognition, and shows that it gives better performance when compared with DNN posteriors combined with MFCC feature.

In this paper, we further investigate the effectiveness of incorporating ASR acoustic model into factor analysis. Following the scheme in [7], we collect posterior statistics from Deep Neural Networks (DNN) trained with raw MFCC and MFCC with different feature transformations, including Linear Discriminant Analysis (LDA), Maximum Likelihood Linear Transformation (MLLT) and feature-space Maximum Likelihood Linear Regression (fMLLR). We also perform decoding for speech utterances and try to use decoded lattice posteriors for speaker verificiation. All these method have shown improvement over naive DNN trained with MFCC features. This also opens up a basic question for factor analysis based speaker verification: what is the best way to generate posteriors for i-vector extraction? On the other hand, we provide derivation of factor analysis for speaker verification using variational Bayesian framework, with bias term included in hidden variables as is done in Kaldi. Implementation details of factor analysis and PLDA in Kaldi toolkit are also discussed.

In following sections, we present standard speaker verification pipeline and illustrate the details of Kaldi's implementation. We then proceed to introduce LDA, MLLT and fMLLR transformations for speech recognition and sequence-discriminative training. Finally, we present experimental results comparing different systems.

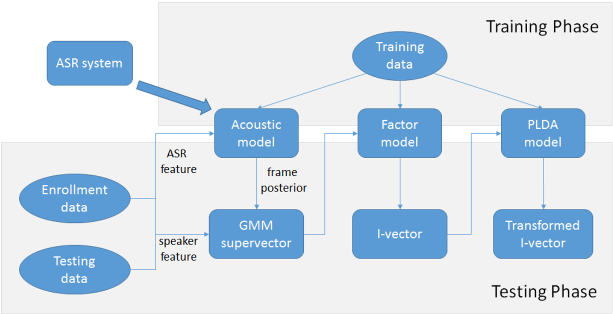

Figure 1: Speaker verification pipeline

2. Speaker verification pipeline

A general speaker verification pipeline is shown in Figure 1. Thanks to the scheme proposed in [7], one could use separate feature streams for frame posterior estimation and speaker ID front-end. The focus of this work is comparing ASR acoustic models for frame posterior generation.

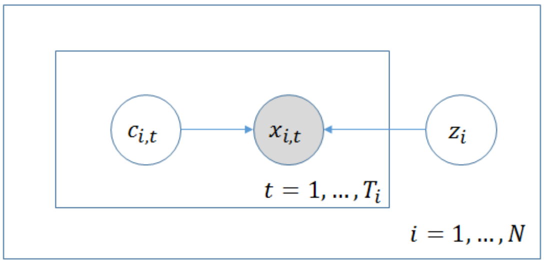

Figure 2: Graphical model of mixture factor analysis

2.1. Factor analysis for speaker identification

Factor analysis for speaker identification is well-formulated in [2]. In this section, we propose to preserve the GMM structure and mixture priors of the model, and derive training formulas using variational Bayes inference. This is different from the model used in [2] where fixed frame alignments are used for model formulation. Our approach is in line with what was mentioned in [7] when alignments are replaced by prior. This derivation makes it clear how we perform EM for mixture factor analysis.

Speech features are modeled by GMM with prior

$$\begin{aligned} x_{i,t}|c_{i,t}, z_i &\sim \mathcal{N}(A_{c_{i,t}} z_i, \Psi_{c_{i,t}}) \\ z_i \sim \mathcal{N}(\nu, I), &\quad c_{i,t} \sim p_c(k) \end{aligned} \tag{1}$$

where $x_{i,t}$ is p-dimensional feature vector for frame t of conversation i. $c_{i,t}$ indicates the mixture that generates $x_{i,t}$. z_i is a q-dimensional latent factor (i.e. i-vector) for this conversation. $A_{c_{i,t}}$ is a p by q projection matrix for mixture $c_{i,t}$ that projects i-vector to feature space, and $\Psi_{c_{i,t}}$ is covariance matrix for mixture $c_{i,t}$. $p_c(k)$ is prior distribution of Gaussian mixtures, with $\sum_k p_c(k) = 1$. The model is shown in Figure 2, and model parameters $\theta = \{A_c, \Psi_c | \forall c\}$.

To perform maximum likelihood estimation (MLE), we use likelihood function as our objective

$$p(x|\theta) = \prod_i p(z_i) \prod_t \sum_c p(x_{i,t}|c_{i,t}, z_i, \theta) p_c(c_{i,t}) \tag{2}$$

and it is maximized using EM algorithm with auxiliary function

$$\begin{aligned} Q(\theta|\theta^t) &= \mathbb{E}_{c,z|x,\theta^t} \log p(x, c, z|\theta) \\ &= \mathbb{E}_{z|x,\theta^t} \left[\mathbb{E}_{c|z,x,\theta^t} \log p(x, c, z|\theta) \right] \\ \log p(x, c, z|\theta) &\propto \log p(x, c|z, \theta) + \sum_i \log(p(z_i)) \\ \log p(x, c|z, \theta) &= \sum_i \sum_t \log(p(x_{i,t}|c_{i,t}, z_i) p(c_{i,t}|z_i)) \end{aligned} \tag{3}$$

where $x = \{x_{i,t} | \forall i, \forall t\}$, $c = \{c_{i,t} | \forall i, \forall t\}$, $z = \{z_i | \forall i\}$. Here, both z and c are considered as latent variables in EM framework.

Since z and c are conditional dependent, there is no close form solution to update them in a joint fashion. However, we could approximate auxiliary function and posterior distribution assuming conditional independence between z and c

$$Q(\theta|\theta^t) \approx \mathbb{E}_{z|x,\theta^t} \left[\mathbb{E}_{c|x,\theta^t} \log p(x, c, z|\theta) \right] \tag{4}$$

Following the derivation of EM for GMM in [9], the auxiliary function could be simplified as

$$Q(\theta|\theta^t) \propto \mathbb{E}_{z|x,\theta^t} \sum_i \left[\log p(z_i) + \sum_t \sum_k \log p(x_{i,t}|k, z_i) \gamma_{i,t}^k \right] \tag{5}$$

where $\gamma_{i,t}^k$ denotes posterior distribution of $c_{i,t}$ given $x_{i,t}$, i.e. $p_{c|x}(k|x_{i,t})$.

From here we need posterior distribution of z given x to proceed.

$$\begin{aligned} p(z_i|x_i) &\propto p(z_i) p(x_i|z_i) \\ &\propto p(z_i) \prod_t \sum_c p(x_{i,t}|c_{i,t}, z_i) p(c_{i,t}) \end{aligned} \tag{6}$$

Here $x_i = \{x_{i,t} | \forall t\}$, and similarly $c_i = \{c_{i,t} | \forall t\}$.

This is also intractable for analytical solution. However, we could use variational Bayes method [10] to approximate it by

$$\begin{aligned} p(z_i|x_i) &\approx p(z_i) e^{\mathbb{E}_{c_i|x_i} \log p(x_i, c_i|z_i, \theta)} \\ &\approx p(z_i) \prod_t \prod_k p(x_{i,t}|k, z_i)^{\gamma_{i,t}^k} \end{aligned} \tag{7}$$

So E-step gives

$$\begin{aligned} \mathbb{E}_{z_i|x_i} &= \text{Var}_{z_i|x_i} \cdot \left(\sum_k A_k^\top \Psi_k^{-1} \sum_t \gamma_{i,t}^k x_{i,t} + \nu \right) \\ \text{Var}_{z_i|x_i} &= \left(\sum_k \left(A_k^\top \sum_t \gamma_{i,t}^k \Psi_k^{-1} A_k \right) + I \right)^{-1} \end{aligned} \tag{8}$$

and maximize the auxiliary function w.r.t. z_i gives

$$\begin{aligned} A_k &= \left[\sum_i \left[\sum_t \gamma_{i,t}^k x_{i,t} \, \mathbb{E}_{z_i|x_i} \, z_i \right] \right] \left[\sum_i \left[\sum_t \gamma_{i,t}^k \, \mathbb{E}_{z_i|x_i} \, z_i z_i^\top \right] \right]^{-1} \\ \Psi_k &= \frac{1}{\sum_i \sum_t \gamma_{i,t}^k} \mathbb{E}_{z|x} \sum_i \sum_t \gamma_{i,t}^k (x_{i,t} - A_k z_i)(x_{i,t} - A_k z_i)^\top \end{aligned} \tag{9}$$

These formulas are consistent with those derived in [2] using posteriors in model formulation.

After updating projection matrix and covariance matrix, Kaldi also applies a Minimum-Divergence (MD) [11] step to speed up model learning, which includes an extra step to update prior ν. An extra transformation (Householder transformation) [12] is used to complete prior update.

2.2. PLDA for speaker identification

Several formulations of PLDA have been proposed by researchers, and they can be unified as the same one [13]. Kaldi's PLDA follows the formulation proposed in [14].

$$x_{i,j} = \mu_g + A u_{i,j}, \quad u_{i,j} \sim \mathcal{N}(v_i, I), \quad v_i \sim \mathcal{N}(0, \Psi) \tag{10}$$

where $x_{i,j}$ is sample j from speaker i (in this case, they are i-vectors extracted from previous step), μ_g is global mean of the data sample. $u_{i,j}$ is sample-specific latent vector in transformed space, and A is the transformation. v_i is speaker specific latent vector for speaker i, and its variance Ψ is a diagonal matrix. This model assumes equal variance for different identities, which could be seen as score normalization model.

Though this model could be trained by EM directly, the training process becomes easier if we convert it to two-covariance form [15]

$$\begin{aligned} y_i &\sim \mathcal{N}(0, \Sigma_B) \\ x_{i,j}|y_i &\sim \mathcal{N}(\mu_g + y_i, \Sigma_W) \end{aligned} \tag{11}$$

Here, y_i is latent vector for speaker i. Σ_B is between-class variance and Σ_W is within-class variance.

The conversion is done by setting

$$y_i = Av_i, \quad \Sigma_B = A^\top \Psi A, \quad \Sigma_W = A^\top A \tag{12}$$

μ_g is estimated as global mean of training data and is fixed during model training.

Kaldi uses an EM algorithm that is slight different from what was described in [13]. Model learning is speeded up by introducing $m_i = \frac{1}{n_i}\sum_j x_{i,j}$, so the model becomes

$$\begin{aligned} y_i &\sim \mathcal{N}(0, \Sigma_B) \\ m_i|y_i &\sim \mathcal{N}(\mu_g + y_i, n_i^{-1}\Sigma_W) \end{aligned} \tag{13}$$

and auxiliary function is

$$\begin{aligned} Q(\theta|\theta^t) &= \sum_i \mathbb{E}_{y_i|m_i,\theta^t} \log p(m_i, y_i|\theta) \\ p(m_i, y_i|\theta) &= p(m_i|y_i)p(y_i) \end{aligned} \tag{14}$$

In E-step, conditional expectation are derived using conjugate prior

$$\begin{aligned} \mathbb{E}_{y_i|m_i} &= (n_i\Sigma_W^{-1} + \Sigma_B^{-1})^{-1} n_i \Sigma_W^{-1}(m_i - \mu_g) \\ \text{Var}_{y_i|m_i} &= (n_i\Sigma_W^{-1} + \Sigma_B^{-1})^{-1} \end{aligned} \tag{15}$$

and M-step model update formulae is

$$\begin{aligned} \Sigma_W &= \frac{1}{N}\sum_i \mathbb{E}_{y_i|m_i}\, n_i(m_i - \mu_g - y_i)(m_i - \mu_g - y_i)^\top \\ \Sigma_B &= \frac{1}{N}\sum_i \mathbb{E}_{y_i|m_i}\, y_i y_i^\top \end{aligned} \tag{16}$$

The model is then converted back to the form shown in Equ. (10) by performing Cholesky decomposition of Σ_W and eigenvalue decomposition of transformed Σ_B.

Once model is trained, transformed vectors $u_{i,j}$ could be extracted form i-vector $x_{i,j}$, and then used for inference against enrollment data. This part is covered in Section 3.1 in [14].

3. Importing statistics from ASR

A novel framework for speaker recognition was proposed in [7] where a DNN trained for ASR is used to produce frame alignments. These alignments are used as $\gamma_{i,t}^k$ in equation 8 in our formulation. It was stated that this pipeline integrates information from speech content directly into statistics. In this work, we investigate the effectiveness of better senone posteriors.

Many techniques that improve ASR performance are based on transformation of feature / model, and another family of methods called sequence-discriminative training [16] analyzes conditional dependence between frames and optimizes objectives defined with regard to whole utterances.

3.1. Linear Discriminant Analysis for speech recognition

LDA is a well-known technique for speech recognition [17,18]. In general, we seek to obtain a transformation so that it maximizes the separability of transformed data. This is usually done by solving a generalized eigen-value decomposition problem. In Kaldi, LDA transformation matrix is computed to project MFCC features (with delta and acceleration) into a 40-dim subspace with triphone senones as class labels.

3.2. Maximum Likelihood Linear Transformation

MLLT (also known as Global Semi-tied Covariance) is another important technique for speech recognition [19, 20]. It is a global transformation matrix used to maximize frame log-likelihood with respect to some constraint. This is usually done using Expectation Maximization. In Kaldi, MLLT is performed on top of LDA features and is performed in feature space.

3.3. fMLLR transforms

fMLLR (also known as CMLLR) is a useful technique for speaker-adaptive training (SAT) of speech recognition [21]. It is a speaker-specific feature-space affine transformation that maximize frame log-likelihood, estimated using EM. Kaldi performs SAT on top of LDA and MLLT.

3.4. Sequence discriminative training

Sequence discriminative training was developed to address sequential feature of speech. In brief, it tries to optimize objectives that are closely related to sequence classification accuracy [16]. Popular objectives include Maximum Mutual Information (MMI) [22], boosted MMI [23], Minimum Phone Error [24] and state-level Minimum Bayes Risk [25].

4. Experiments

4.1. Datasets

We use 300-hour Switchboard-I Training set [26] for ASR model training. The data for ASR system development is the 1831-segment SWB part of the NIST 2000 Hub5 evaluation set [27]. The UBM and i-vector model training data consists of SWB and NIST SREs. The SWB data contains 21,254 utterances from 6,820 speakers of SWB 2 Phases I, II and III. The SRE dataset consists 18,715 utterances / channels from 3,009 speakers of SREs from 2004 to 2006. PLDA model is trained using NIST SREs from 2004 to 2008, which consists of 28,579 utterances from 5,321 speakers.

We evaluate our systems on the condition 5 extended task of SRE10 [28]. The evaluation consists of conversational telephone speech in both enrollment and test utterances. There are 387,112 trials, over 98% of which are non-target comparisons.

4.2. Setup

In this paper, the Kaldi toolkit [29] is used for both speech and speaker recognition. For speech recognition system, standard 13-dim MFCC feature is extracted and used for maximum likelihood GMM model training. Features are then transformed using LDA+MLLT before SAT training. After GMM training is done, three tanh-neuron DNN-HMM hybrid systems are trained using different kinds of features: 1. MFCC; 2. LDA + MLLT transformed MFCC; 3. LDA + MLLT + fMLLR transformed MFCC. Details of DNN training follows Section 2.2 in [30].

For speaker verification system, we follow the setup in [31]. The front-end consists of 20 MFCCs with a 25ms frame-length. The features are mean-normalized over a 3 second window. Delta and acceleration are appended to create 60 dimensional frame-level feature vectors. I-vector dimension is set to 600.

To get fMLLR transformations, we need to perform ASR for all speaker verification data and also a pre-ASR Voice Activity Detection (VAD). VAD is done by performing phone decoding with limited search beam, and speaker independent decoding and fMLLR decoding are done in an iterative fashion. These steps are time-consuming in practice, making it not applicable for real-time scenarios yet.

4.3. Effect of transformations

Table 1 shows EERs of factor analysis systems trained with different posteriors [1], and Figure 3 plots corresponding DET curve for these systems. All the experiments in this table use standard speaker ID MFCC features. As is shown, significant improvements are achieved when we use posteriors from DNN trained with transformations [2]. We could also see that improvements on EER aligns with speech recognition performance of ASR systems, and the best performance is from sequence discriminative training with LDA, MLLT and fMLLR transformation.

	eval2000	EER		
	WER	male	female	all
UBM (4096)	–	5.92	6.80	6.36
UBM (8192)	–	5.83	6.80	6.31
DNN-MFCC (8824)	19.4	5.63	7.05	6.39
+ LDA + MLLT	16.3	4.07	5.43	4.84
+ SAT (fMLLR)*	15.0	3.98	5.02	**4.55**
+ MPE*	13.5	3.58	4.75	**4.38**
GMM-fMLLR-latpost*	21.8	4.50	5.99	5.45
DNN-fMLLR-latpost*	15.0	4.16	5.00	4.66

Table 1: Speaker ID feature with different posteriors

4.4. Posterior from lattice

We also try to incorporate phonetic content using posteriors from decode lattices. We could see from Table 1 that these posteriors give comparable results as those come right out of acoustic models. However, they do require more computation, so in general these are not good alternatives for this task.

4.5. Using ASR features for speaker verification

We learn from Section 4.3 that posteriors generated from fMLLR-DNN benefit speaker verification a lot. This is somewhat surprising because fMLLR transformation is believed to remove speaker specific information. However, it gives better posterior estimates, and thus help speaker verification. In this section, we would like to use transformed features for speaker verification directly. Table 2 compares different features front-end for factor analysis, where they all share the same posteriors from fMLLR based DNN. “Default” denotes standard MFCC feature used in previous experiments, “ASR LDA+MLLT” denotes MFCC feature transformed by LDA and MLLT, and “ASR fMLLR” denotes MFCC feature transformed by LDA, MLLT and fMLLR. We could see from the table that both LDA+MLLT and fMLLR features degrade system performances, which is consistent with our knowledge. Meanwhile, it is interesting to note that these features, though transformed to remove speaker specific characteristics, still contain speaker information and can be used for speaker ID. This might raise an issue when one wants to protect speaker information by applying fMLLR transform on speech features and transmit over the Internet.

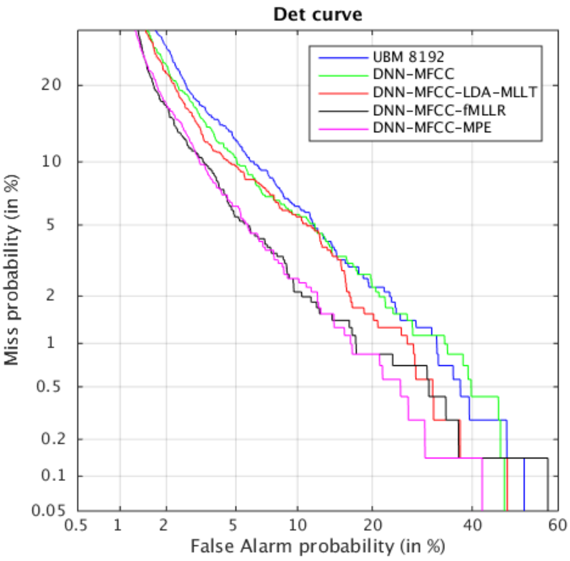

Figure 3: DET curve for speaker ID performance

Speaker recog feats	EER		
	male	female	all
Default	3.98	5.02	**4.55**
ASR LDA+MLLT	5.43	7.24	6.35
ASR fMLLR	7.85	9.42	8.84

Table 2: fMLLR posteriors with different feature front-end

5. Conclusions

In this paper, we study the effectiveness of state-of-the-art ASR techniques for speaker verification. We found that speaker verification performance aligns with speech recognition performance when we import posteriors from acoustic models trained for ASR. Out of all the systems, DNN trained with fMLLR features and MPE objective produces posteriors that benefit factor analysis most. We also presented derivation of factor analysis in the framework of GMM with mixture prior, using variational Bayes inference, and explains implementation details in Kaldi toolkit.

6. Acknowledgements

We would like to thank Daniel Povey, Daniel Garcia-Romero and David Snyder for their development of speaker verification systems in Kaldi.

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1450916.

[1] EERs in these experiments are worse than those reported in [31] because we use less data for UBM, FA and PLDA model training. Specifically, we left out Switchboard Cellular, SRE 2005 test set, SRE 2006 test set and SRE 2008 due to computation issue.

[2] Asterisk (*) indicates experiments require decoding of speech

7. References

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

[2] P. Kenny, G. Boulianne, and P. Dumouchel, “Eigenvoice modeling with sparse training data,” *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 3, pp. 345–354, 2005.

[3] P. Kenny, “Bayesian speaker verification with heavy-tailed priors.” in *Odyssey*, 2010.

[4] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[5] F. Seide, G. Li, and D. Yu, “Conversational speech transcription using context-dependent deep neural networks.” in *Interspeech*, 2011.

[6] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *ICASSP*. IEEE, 2013.

[7] Y. Lei, S. Nicolas, L. Ferrer, and M. McLaren, “A novel scheme for speaker recognition using a phonetically-aware deep neural network,” in *ICASSP*. IEEE, 2014.

[8] F. Richardson, D. Reynolds, and N. Dehak, “Deep neural network approaches to speaker and language recognition,” *Signal Processing Letters, IEEE*, vol. 22, no. 10, pp. 1671–1675, 2015.

[9] J. A. Bilmes, “A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models,” *International Computer Science Institute*, vol. 4, no. 510, p. 126, 1998.

[10] H. Attias, “A variational bayesian framework for graphical models,” *Advances in neural information processing systems*, vol. 12, no. 1-2, pp. 209–215, 2000.

[11] N. Brümmer, “The em algorithm and minimum divergence,” *Agnitio Labs Technical Report. Online: http://niko.brummer.googlepages.com/EMandMINDIV.pdf*, 2009.

[12] “Householder transformation,” http://en.wikipedia.org/wiki/Householder_transformation, accessed: 2016-03-23.

[13] A. Sizov, K. A. Lee, and T. Kinnunen, “Unifying probabilistic linear discriminant analysis variants in biometric authentication,” in *Structural, Syntactic, and Statistical Pattern Recognition*. Springer, 2014, pp. 464–475.

[14] S. Ioffe, “Probabilistic linear discriminant analysis,” in *ECCV*. Springer, 2006.

[15] N. Brümmer and E. De Villiers, “The speaker partitioning problem,” in *Odyssey*, 2010.

[16] D. Povey, “Discriminative training for large vocabulary speech recognition,” Ph.D. dissertation, University of Cambridge, 2005.

[17] R. Haeb-Umbach and H. Ney, “Linear discriminant analysis for improved large vocabulary continuous speech recognition,” in *ICASSP*. IEEE, 1992.

[18] L. C. Wood, D. J. Pearce, and F. Novello, “Improved vocabulary-independent sub-word hmm modelling,” in *ICASSP*. IEEE, 1991.

[19] M. J. Gales, “Maximum likelihood linear transformations for hmm-based speech recognition,” *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[20] ——, “Semi-tied covariance matrices for hidden markov models,” *Speech and Audio Processing, IEEE Transactions on*, vol. 7, no. 3, pp. 272–281, 1999.

[21] C. J. Leggetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models,” *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, 1995.

[22] L. Bahl, P. F. Brown, P. V. De Souza, and R. L. Mercer, “Maximum mutual information estimation of hidden markov model parameters for speech recognition,” in *proc. icassp*, vol. 86, 1986, pp. 49–52.

[23] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, “Boosted mmi for model and feature-space discriminative training,” in *ICASSP*. IEEE, 2008.

[24] D. Povey and P. C. Woodland, “Minimum phone error and i-smoothing for improved discriminative training,” in *ICASSP*. IEEE, 2002.

[25] V. Goel and W. J. Byrne, “Minimum bayes-risk automatic speech recognition,” *Computer Speech & Language*, vol. 14, no. 2, pp. 115–135, 2000.

[26] J. Godfrey and E. Holliman, “Switchboard-1 release 2 ldc97s62,” Linguistic Data Consortium, 1993.

[27] J. Fiscus, W. M. Fisher, A. F. Martin, M. A. Przybocki, and D. S. Pallett, “2000 nist evaluation of conversational speech recognition over the telephone: English and mandarin performance results,” in *Proc. Speech Transcription Workshop*. Citeseer, 2000.

[28] “The nist year 2010 speaker recognition evaluation plan,” http://www.nist.gov/itl/iad/mig/upload/NIST_SRE10_evalplan-r6.pdf, National Institute of Standards and Technology, accessed: 2016-03-22.

[29] D. Povey, A. Ghoshal, G.Boulianne, L. Burget, O.Glembek, N. Goel, M. Hannermann, P. Motlíček, Y. Qian, P. Schwartz, J. Silovský, G. Stemmer, and K. Veselý, “The kaldi speech recognition toolkit,” in *ASRU*. IEEE, 2011.

[30] K. Veselýl, M. Hannemann, and L. Burget, “Semi-supervised training of deep neural networks,” in *ASRU*. IEEE, 2013.

[31] D. Snyder, D. Garcia-Romero, and D. Povey, “Time delay deep neural network-based universal background models for speaker recognition,” in *ASRU*. IEEE, 2015.