Available online at www.sciencedirect.com

ScienceDirect

Computer Speech and Language 28 (2014) 940–958

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Simplified supervised i-vector modeling with application to robust and efficient language identification and speaker verification☆,☆☆

Ming Li[a,b,c,*], Shrikanth Narayanan[a]

[a] Signal Analysis and Interpretation Laboratory (SAIL), University of Southern California, Los Angeles, CA 90089, USA
[b] Sun Yat-Sen University Carnegie Mellon University Joint Institute of Engineering, Sun Yat-Sen University, Guangzhou, China
[c] Sun Yat-Sen University Carnegie Mellon University Shunde International Joint Research Institute, Shunde, China

Received 16 October 2012; received in revised form 23 January 2014; accepted 15 February 2014
Available online 12 March 2014

Abstract

This paper presents a simplified and supervised i-vector modeling approach with applications to robust and efficient language identification and speaker verification. First, by concatenating the label vector and the linear regression matrix at the end of the mean supervector and the i-vector factor loading matrix, respectively, the traditional i-vectors are extended to label-regularized supervised i-vectors. These supervised i-vectors are optimized to not only reconstruct the mean supervectors well but also minimize the mean square error between the original and the reconstructed label vectors to make the supervised i-vectors become more discriminative in terms of the label information. Second, factor analysis (FA) is performed on the pre-normalized centered GMM first order statistics supervector to ensure each gaussian component's statistics sub-vector is treated equally in the FA, which reduces the computational cost by a factor of 25 in the simplified i-vector framework. Third, since the entire matrix inversion term in the simplified i-vector extraction only depends on one single variable (total frame number), we make a global table of the resulting matrices against the frame numbers' log values. Using this lookup table, each utterance's simplified i-vector extraction is further sped up by a factor of 4 and suffers only a small quantization error. Finally, the simplified version of the supervised i-vector modeling is proposed to enhance both the robustness and efficiency. The proposed methods are evaluated on the DARPA RATS dev2 task, the NIST LRE 2007 general task and the NIST SRE 2010 female condition 5 task for noisy channel language identification, clean channel language identification and clean channel speaker verification, respectively. For language identification on the DARPA RATS, the simplified supervised i-vector modeling achieved 2%, 16%, and 7% relative equal error rate (EER) reduction on three different feature sets and sped up by a factor of more than 100 against the baseline i-vector method for the 120 s task. Similar results were observed on the NIST LRE 2007 30 s task with 7% relative average cost reduction. Results also show that the use of Gammatone frequency cepstral coefficients, Mel-frequency cepstral coefficients and spectro-temporal Gabor features in conjunction with shifted-delta-cepstral features improves the overall language identification performance significantly. For speaker verification, the proposed supervised

☆ This paper has been recommended for acceptance by M. Adda-Decker.

☆☆ The MATLAB codes package is shared at http://jie.sysu.edu.cn/~mli/SimplifiedSupervisedIvectorMatlab.zip.

* Corresponding author at: Sun Yat-Sen University Carnegie Mellon University Joint Institute of Engineering, Sun Yat-Sen University, Guangzhou, China, Tel: +86 20 84115633.

E-mail addresses: liming46@mail.sysu.edu.cn, mingli1@cmu.edu (M. Li), shri@sipi.usc.edu (S. Narayanan).

URLs: http://jie.sysu.edu.cn/ mli/ (M. Li), http://sail.usc.edu/ (S. Narayanan).

http://dx.doi.org/10.1016/j.csl.2014.02.004

i-vector approach outperforms the i-vector baseline by relatively 12% and 7% in terms of EER and norm old minDCF values, respectively.

Keywords: Language identification; Speaker verification; I-vector; Supervised i-vector; Simplified i-vector; Simplified supervised i-vector

1. Introduction

The goal of language identification (LID) is to automatically determine the language spoken in a given segment of speech. In real world applications e.g., security or defense, the speech signals could come from extremely noisy and distorted communication channels, such as short wave AM broadcasting. Thus robust LID on noisy and degraded data represents an important need.

Several algorithmic and computational advances have enabled impressive LID performance in the state of the art. Approaches using phonotactic information, namely PRLM (phoneme recognizer followed by language models) and PPRLM (parallel PRLM), have been shown to be quite successful (Zissman, 1995; Yan and Barnard, 1995). In this phonotactic modeling framework, a set of tokenizers are used to transcribe the input speech into token strings or lattices which are later scored by n-gram language models (Gauvain et al., 2004) or mapped into a bag of trigrams feature vector for support vector machine (SVM) modeling (Li et al., 2007a). Although the traditional hidden Markov model (HMM) based phone recognizer (Rabiner, 1989) is widely used as tokenizer in the state-of-the-art systems, other types of tokenizations could also be applied here (Li et al., 2013), for example, Gaussian Mixture Model (GMM) tokenization (Torres-Carrasquillo et al., 2002b), universal phone recognition (UPR) (Li et al., 2007a), articulatory attribute-based approach (Siniscalchi et al., 2010), deep neural networks based phone recognizer (Hinton et al., 2012; Deng and Li, 2013), just to name a few. A recent literature review on LID is provided in Li et al. (2013).

With the introduction of shifted-delta-cepstral (SDC) acoustic features (Torres-Carrasquillo et al., 2002a), promising results using GMM framework with factor analysis (Castaldo et al., 2007; Kenny et al., 2007a,b, 2008), supervector modeling (Campbell et al., 2006; Li et al., 2007b) and maximum mutual information (MMI) based discriminative training (Burget et al., 2006) have also been reported for LID. In this work, we focus on the acoustic level systems.

Another critical speech processing application domain is speaker verification (SV); this domain also is challenged by significant variability in the speech signal. In particular, the use of joint factor analysis (JFA) (Kenny et al., 2007a,b, 2008) has also contributed to the state of the art performance in text independent speaker verification. It is a powerful technique for compensating the variability caused by different channels and sessions.

Recently, total variability i-vector modeling has gained significant attention in both LID and SV domains due to its excellent performance, low complexity and small model size (Dehak et al., 2011a,b; Martinez et al., 2011). In this modeling, first, a single factor analysis is used as a front end to generate a low dimensional total variability space which jointly models language, speaker and channel variabilities all together (Dehak et al., 2011b). Then, within this i-vector space, variability compensation methods, such as Within-Class Covariance Normalization (WCCN) (Hatch et al., 2006), Linear Discriminative analysis (LDA) and Nuisance Attribute Projection (NAP) (Campbell et al., 2006), are performed to reduce the variability for subsequent modeling (e.g., using SVM, logistic regression (LR) (Martinez et al., 2011) and neural network (Matejka et al., 2012) for LID and probabilistic linear discriminant analysis (PLDA) (Matejka et al., 2011; Kenny, 2010) for SV, respectively).

However, the i-vector training and extraction algorithms are computationally expensive, especially for large GMM model sizes and large training data sets (Aronowitz and Barkan, 2012; Glembek et al., 2011). Both Glembek et al. (2011) and Aronowitz and Barkan (2012) used a pre-calculated UBM weighting vector to approximate each utterance's 0th order GMM statistics vector to avoid the computationally-expensive GMM component wise matrix operations for the SV task. This approximation resulted in 10–25 times computational cost reduction at the expense of a significant performance degradation (about 17% EER) (Aronowitz and Barkan, 2012). By enforcing this approximation in both training and extraction stages, the performance degradation can be reduced notably (Glembek et al., 2011) on condition that there is no or very little mismatch between train/test data and UBM data. Therefore, we investigated an alternative robust and efficient solution for LID and SV tasks in this work.

We perform factor analysis (FA) on the pre-normalized centered GMM first-order statistics supervector to ensure each gaussian component's statistics sub-vector is treated equally in the factor analysis which reduces the computational cost significantly (by a factor of 25). In this way, each utterance is represented with one single pre-normalized supervector as the feature vector plus one total frame number to control its importance against the prior. Each component's sub-vector of statistics is normalized by its own occupancy probability square root, thus mitigating the mismatch between the global pre-calculated average weighting vector (Glembek et al., 2011 adopted the UBM weights) and each utterance's own occupancy probability distribution vector. Furthermore, since there is only one global total frame number inside the matrix inversion, we propose to pre-construct a global lookup table of the resulting matrices against the frame numbers' log values; the reason to choose the log domain is that the smaller the total frame number, the more important it is against the prior. By looking at the table, each utterance's i-vector extraction is further sped up by a factor of 4 with only a small table index quantization error. The larger the table, the smaller this quantization error.

Moreover, as a single unsupervised method, i-vectors cover language, speaker, channel and other variabilities all together which necessarily requires variability compensation methods (both LDA and WCCN are linear) as the back end. This motivated us to investigate joint optimization to minimize the weighted summation of both the re-construction error and the linear classification error simultaneously. Compared to the sequential optimization used for traditional i-vectors, this proposed joint optimization can select the top eigen directions only related to the given labels. This can help reduce the non-relevant information in the i-vector space, such as noise and variabilities from undesired sources (e.g., non-language related factors for LID and session/channel/language related variabilities for SV).

In this work, the traditional i-vectors are extended to label-regularized supervised i-vectors by concatenating the label vector and the linear regression matrix at the end of the mean supervector and the i-vector factor loading matrix, respectively. There are some obvious extensions of this supervised i-vector framework. We can let the appended label vector be the parameter vector that we want to perform regression with (e.g., ages Li et al., 2012; Bahari et al., 2012, paralinguistic measures Schuller et al., 2013) to make the proposed framework suitable for regression problems. The reason for using a linear classification/regression matrix W is that many back end classification modules in LID and SV are linear (linear kernel SVM, inner product, WCCN, LDA, etc.). Moreover, if the classification or regression relation is not linear, we can use non-linear mapping as a preprocessing step before generating the label vectors. The contribution weight of each supervector dimension and each target class in the objective function is automatically calculated by iterative EM training. The traditional i-vector system serves as our baseline.

Finally, inspired by the success of robust features for noisy data based SV tasks (Shao and Wang, 2008; Lei et al., 2012), we also applied the auditory-inspired Gammatone frequency cepstral coefficients (GFCC) (Shao and Wang, 2008; Zhao and Wang, 2013) features and the spectro-temporal Gabor features (Kleinschmidt and Gelbart, 2002) for robust LID task on the noisy data as additional performance improvement steps. When fused with traditional MFCC and SDC feature based systems, the overall system performance was further enhanced.

The remainder of the paper is organized as follows. The baseline and the proposed algorithms are explained in Sections 2 and 3, respectively. Complexity analysis is given in Section 3.3. Experimental results and discussions are presented in Section 4 while conclusions are provided in Section 5.

2. The i-vector baseline

In the total variability space, there is no distinction between the language effects, speaker effects and the channel effects. Rather than separately using the eigenvoice matrix V and the eigenchannel matrix U (Kenny et al., 2007b), the total variability space simultaneously captures the speaker and channel variabilities (Dehak et al., 2011a). Given a C component GMM UBM model λ with $\lambda_c = \{p_c, \boldsymbol{\mu_c}, \boldsymbol{\Sigma_c}\}$, $c = 1, \ldots, C$ and an utterance with a L frame feature sequence $\{\boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_L}\}$, the 0th and centered 1st order Baum-Welch statistics on the UBM are calculated as follows:

$$N_c = \sum_{t=1}^{L} P(c|\boldsymbol{y_t}, \lambda) \tag{1}$$

$$\boldsymbol{F_c} = \sum_{t=1}^{L} P(c|\boldsymbol{y_t}, \lambda)(\boldsymbol{y_t} - \boldsymbol{\mu_c}) \tag{2}$$

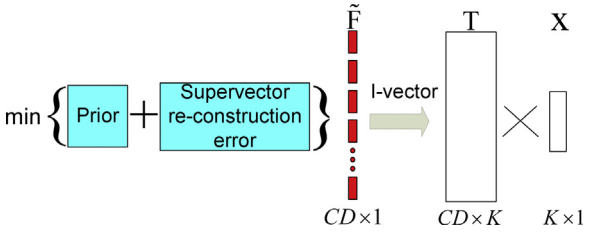

Fig. 1. Schematic of the i-vector baseline approach. See Section 2 for details on notation.

where $c = 1, \ldots, C$ is the GMM component index and $P(c|\boldsymbol{y_t}, \lambda)$ is the occupancy probability for $\boldsymbol{y_t}$ on λ_c. The corresponding centered mean supervector $\tilde{\boldsymbol{F}}$ is generated by concatenating all the $\tilde{\boldsymbol{F}}_{\boldsymbol{c}}$ together:

$$\tilde{\boldsymbol{F}}_{\boldsymbol{c}} = \frac{\sum_{t=1}^{L} P(c|\boldsymbol{y_t}, \lambda)(\boldsymbol{y_t} - \boldsymbol{\mu_c})}{\sum_{t=1}^{L} P(c|\boldsymbol{y_t}, \lambda)}. \tag{3}$$

The centered GMM mean supervector $\tilde{\boldsymbol{F}}$ can be projected on a low rank factor loading matrix $\boldsymbol{T}$ following the standard factor analysis framework as shown in Fig. 1:

$$\tilde{\boldsymbol{F}} \rightarrow \boldsymbol{Tx}, \tag{4}$$

where $\boldsymbol{T}$ is a rectangular total variability matrix of low rank and $\boldsymbol{x}$ is the so-called i-vector (Dehak et al., 2011a). Considering a C-component GMM and D dimensional acoustic features, the total variability matrix $\boldsymbol{T}$ is a $CD \times K$ matrix which can be estimated the same way as learning the eigenvoice matrix $\boldsymbol{V}$ in Kenny et al. (2005) except that here we consider that every utterance is produced by a new speaker or in a new language (Dehak et al., 2011a).

Given the centered mean supervector $\tilde{\boldsymbol{F}}$ and total variability matrix $\boldsymbol{T}$, the i-vector is computed as follows (Dehak et al., 2011a):

$$\boldsymbol{x} = (\boldsymbol{I} + \boldsymbol{T^t \Sigma^{-1} N T})^{-1} \boldsymbol{T^t \Sigma^{-1} N \tilde{F}} \tag{5}$$

where $\boldsymbol{N}$ is a diagonal matrix of dimension $CD \times CD$ whose diagonal blocks are $N_c \boldsymbol{I}$, $c = 1, \ldots, C$ and $\boldsymbol{\Sigma}$ is a diagonal covariance matrix of dimension $CD \times CD$ estimated in the factor analysis training step. It models the residual variability not captured by the total variability matrix $\boldsymbol{T}$ (Dehak et al., 2011a). The mathematical interpretation under the standard factor analysis framework is provided in Section 3.1.

In this total variability space, two channel compensation methods, namely Linear Discriminant Analysis (LDA) and Within Class Covariance Normalization (WCCN) (Hatch et al., 2006), are typically applied to reduce variability. LDA attempts to transform the axes to minimize the intra-class variance due to the variability effects and maximize the variance between classes while WCCN uses the inverse of the within-class covariance to normalize the cosine kernel. After LDA and WCCN steps, cosine distance is employed for i-vector modeling. The cosine kernel between two i-vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is defined as follows:

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{< \boldsymbol{x}_1, \boldsymbol{x}_2 >}{\|\boldsymbol{x}_1\|_2 \|\boldsymbol{x}_2\|_2} \tag{6}$$

Finally, PLDA or SVM is adopted as the classifier.

3. Simplified and supervised i-vector

3.1. Label-regularized supervised i-vector

The i-vector training and extraction can be re-interpreted as a classic factor analysis based generative modeling problem. We can assume that the mean supervector is generated by the hidden variable i-vector. For the jth utterance, the prior and the conditional distribution is defined as following multivariate Gaussian distributions:

$$P(\boldsymbol{x_j}) = \mathcal{N}(\boldsymbol{0}, I), \qquad P(\tilde{\boldsymbol{F}}_{\boldsymbol{j}}|\boldsymbol{x_j}) = \mathcal{N}(\boldsymbol{T x_j}, \boldsymbol{N_j^{-1} \Sigma}) \tag{7}$$

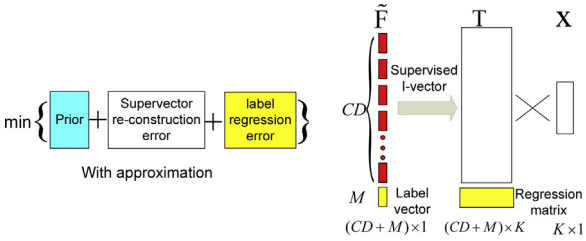

Fig. 2. Schematic of the simplified supervised i-vector formulation.

therefore, the posterior distribution of the hidden variable i-vector $\boldsymbol{x}$ given the observed $\tilde{\boldsymbol{F}}$ is:

$$P(\boldsymbol{x}_j|\tilde{\boldsymbol{F}}_j) = \mathcal{N}((\boldsymbol{I} + \boldsymbol{T}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{N}_j\boldsymbol{T})^{-1}\boldsymbol{T}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{N}_j\tilde{\boldsymbol{F}}_j, (\boldsymbol{I} + \boldsymbol{T}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{N}_j\boldsymbol{T})^{-1}). \tag{8}$$

The mean of the posterior distribution (point estimate) is adopted as the estimation of i-vector.

As shown in Fig. 2, the traditional i-vectors are extended to the label-regularized supervised i-vectors by concatenating the label vector and the linear regression matrix at the end of the mean supervector and the i-vector factor loading matrix, respectively. These supervised i-vectors are optimized not only to reconstruct the mean supervectors well but also to minimize the mean square error between the original and the reconstructed label vectors, and thus can make the supervised i-vectors become more discriminative in terms of the regularized label information.

$$P(\boldsymbol{x}_j) = \mathcal{N}(\boldsymbol{0}, I), \qquad P\left(\begin{bmatrix}\tilde{\boldsymbol{F}}_j \\ \boldsymbol{L}_j\end{bmatrix}|\boldsymbol{x}_j\right) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{T}\boldsymbol{x}_j \\ \boldsymbol{W}\boldsymbol{x}_j\end{bmatrix}, \begin{bmatrix}\boldsymbol{N}_j^{-1}\boldsymbol{\Sigma}_1 \\ n_j^{-1}\boldsymbol{\Sigma}_2\end{bmatrix}\right) \tag{9}$$

In (7), (8), (9), $\boldsymbol{x}_j$, $\boldsymbol{N}_j$, $\tilde{\boldsymbol{F}}_j$ and $\boldsymbol{L}_j$ denote the jth utterance's i-vector, $\boldsymbol{N}$ vector, mean supervector and label vector, respectively. $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ denote the variance for CD dimensional mean supervector and M dimensional label vector, respectively. $n_j = \sum_{c=1}^{C} N_{cj}$ where N_{cj} denotes the N_c for the jth utterance. The reason for using a global scalar n_j is that each target class is treated equally in terms of frame length importance, the variance $\boldsymbol{\Sigma}_2$ is adopted to capture the variance and accuracy for each particular class.

We design two types of label vectors as follows (type 1 is mainly for identification purposes and type 2 is for verification tasks):

$$\text{Supervised } type\ 1: \qquad \mathbf{L}_{ij} = \begin{cases} 1 & \text{if } utterance \text{ j is from class i} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

For type 1 label vectors, we want the regression matrix $\mathbf{W}$ to correctly classify the class labels. We denote the dimensionality of label vector $\mathbf{L}_j$ as M. Suppose there are H target classes (H languages for LID or H speakers for SV), $\mathbf{L}_j$ is an H ($M = H$) dimensional binary vector with only one non-zero element with the value of 1 and $\mathbf{W}$ is a $M \times K$ linear regression matrix.

$$\text{Supervised } type\ 2: \qquad \mathbf{L}_j = \bar{\mathrm{x}}_{\mathrm{s_j}}, \mathbf{W} = \mathbf{I}. \tag{11}$$

Type 2 label vectors specify the sample mean vector of all the supervised i-vectors from the same class in the last iteration $\bar{\mathrm{x}}_{\mathrm{s_j}}$ and let the regression matrix be an identity matrix (similar to the one in WCCN). The reason is to reduce the within class covariance and help all the supervised i-vectors to move towards their class sample mean. Therefore, $M = K$ in this case.

The log likelihood of the total Γ utterances is:

$$\sum_{j=1}^{\Gamma} \ln(P(\tilde{\boldsymbol{F}}_j, \boldsymbol{L}_j, \boldsymbol{x}_j)) = \sum_{j=1}^{\Gamma} \left\{ \ln\left(P\left(\begin{bmatrix}\tilde{\boldsymbol{F}}_j \\ \boldsymbol{L}_j\end{bmatrix}|\boldsymbol{x}_j\right)\right) + \ln(P(\boldsymbol{x}_j)) \right\} \tag{12}$$

Combining (9) and (12) together and removing non-relevant items, we can get the objective function J for the Maximum Likelihood (ML) EM training:

$$\begin{aligned} J &= \sum_{j=1}^{\Gamma}\Big(\frac{1}{2}\boldsymbol{x_j^t x_j} + \frac{1}{2}(\tilde{\boldsymbol{F}}_{\boldsymbol{j}} - \boldsymbol{T x_j})^t {\boldsymbol{\Sigma}_1}^{-1} \boldsymbol{N_j}(\tilde{\boldsymbol{F}}_{\boldsymbol{j}} - \boldsymbol{T x_j}) + \frac{1}{2}(\boldsymbol{L_j} - \boldsymbol{W x_j})^t {\boldsymbol{\Sigma}_2}^{-1} n_j(\boldsymbol{L_j} - \boldsymbol{W x_j}) \\ &\quad -\frac{1}{2}\ln(|{\boldsymbol{\Sigma}_1}^{-1}|) - \frac{1}{2}\ln(|{\boldsymbol{\Sigma}_2}^{-1}|)\Big) \end{aligned} \tag{13}$$

It is worth noting that we simply set $\boldsymbol{T}$ and $\boldsymbol{W}$ to be uncorrelated in order to get (13). The reason to make this simplification is that in the factor analysis generative model, given the i-vector $\boldsymbol{x}$, the mean supervector $\boldsymbol{F}$ and label vector $\boldsymbol{L}$ are conditionally independent.

For the E-step, we estimate $E(\boldsymbol{x_j})$ and $E(\boldsymbol{x_j}\mathrm{x_j}^t)$:

$$E(\boldsymbol{x_j}) = (\boldsymbol{I} + \boldsymbol{T^t \Sigma_1^{-1} N_j T} + \boldsymbol{W^t \Sigma_2^{-1}} n_j \boldsymbol{W})^{-1}(\boldsymbol{T^t \Sigma_1^{-1} N_j \tilde{F}_j} + \boldsymbol{W^t \Sigma_2^{-1}} n_j \boldsymbol{L_j}), \tag{14}$$

$$E(\boldsymbol{x_j x_j}^t) = E(\boldsymbol{x_j})E(\boldsymbol{x_j})^t + (\boldsymbol{I} + \boldsymbol{T^t \Sigma_1^{-1} N_j T} + \boldsymbol{W^t \Sigma_2^{-1}} n_j \boldsymbol{W})^{-1}. \tag{15}$$

Then, for the M-step, we need to minimize the following expected objective function:

$$\begin{aligned} E(J) &= \sum_{j=1}^{\Gamma}\Big(\frac{1}{2}Tr[E(\boldsymbol{x_j x_j}^t)] + \frac{1}{2}\tilde{\boldsymbol{F}}_{\boldsymbol{j}}^t \boldsymbol{\Sigma_1^{-1} N_j}\tilde{\boldsymbol{F}}_{\boldsymbol{j}} + \frac{1}{2}Tr[\boldsymbol{T}E(\boldsymbol{x_j x_j}^t)\boldsymbol{T}^t \boldsymbol{\Sigma_1^{-1} N_j}] - \tilde{\boldsymbol{F}}_{\boldsymbol{j}}^t \boldsymbol{\Sigma_1^{-1} N_j T}E(\boldsymbol{x_j}) \\ &+\frac{1}{2}\boldsymbol{L_j}^t \boldsymbol{\Sigma_2^{-1}} n_j \boldsymbol{L_j} + \frac{1}{2}Tr[\boldsymbol{W}E(\boldsymbol{x_j x_j}^t)\boldsymbol{W}^t \boldsymbol{\Sigma_2^{-1}} n_j] - \boldsymbol{L_j}^t \boldsymbol{\Sigma_2^{-1}} n_j \boldsymbol{W}E(\boldsymbol{x_j}) - \frac{1}{2}\ln(|{\boldsymbol{\Sigma}_1}^{-1}|) - \frac{1}{2}\ln(|{\boldsymbol{\Sigma}_2}^{-1}|)\Big) \end{aligned} \tag{16}$$

In deriving(16), we used $Tr[\boldsymbol{T x_j x_j}^t \boldsymbol{T}^t \boldsymbol{\Sigma_1^{-1} N_j}] = Tr[\boldsymbol{x_j}^t \boldsymbol{T}^t \boldsymbol{\Sigma_1^{-1} N_j T x_j}]$. By setting the derivatives of $E(J)$ towards $\boldsymbol{T}$ and $\boldsymbol{W}$ to be $\boldsymbol{0}$, we can get:

$$\sum_{j=1}^{\Gamma} \boldsymbol{N_j T}E(\boldsymbol{x_j x_j}^t) = \sum_{j=1}^{\Gamma} \boldsymbol{N_j}\tilde{\boldsymbol{F}}_{\boldsymbol{j}}E(\boldsymbol{x_j}^t) \tag{17}$$

$$\sum_{j=1}^{\Gamma} n_j \boldsymbol{W}E(\boldsymbol{x_j x_j}^t) = \sum_{j=1}^{\Gamma} n_j \boldsymbol{L_j}E(\boldsymbol{x_j}^t) \tag{18}$$

Since n_j is a scalar, the new $\boldsymbol{W}$ matrix is updated as:

$$\text{Type 1}: \qquad \boldsymbol{W}_{new} = [\sum_{j=1}^{\Gamma} n_j \boldsymbol{L_j}E(\boldsymbol{x_j}^t)][\sum_{j=1}^{\Gamma} n_j E(\boldsymbol{x_j x_j}^t)]^{-1} \tag{19}$$

Since the regression matrix $\boldsymbol{W}$ in the type 2 setup is set to identity, we do not update $\boldsymbol{W}$ for type 2 setup. For the $\boldsymbol{T}$ matrix, we followed the strategy used in Dehak et al. (2011a) to update component by component since N_{cj} is also a scalar.

$$\boldsymbol{T_{cnew}} = [\sum_{j=1}^{\Gamma} N_{cj}\tilde{\boldsymbol{F}}_{\boldsymbol{cj}}E(\boldsymbol{x_j}^t)][\sum_{j=1}^{\Gamma} N_{cj}E(\boldsymbol{x_j x_j}^t)]^{-1} \tag{20}$$

In (20), $\boldsymbol{T_c}$ denotes the $[(c-1)D+1:cD]$ rows sub-matrix of $\boldsymbol{T}$ and $\tilde{\boldsymbol{F}}_{\boldsymbol{cj}}$ is the $[(c-1)D+1:cD]$ elements sub-vector of $\tilde{\boldsymbol{F}}_{\boldsymbol{j}}$. Similarly, by setting $\frac{\partial Ej}{\partial({\boldsymbol{\Sigma}_1}^{-1})}$ and $\frac{\partial Ej}{\partial({\boldsymbol{\Sigma}_2}^{-1})}$ to be $\boldsymbol{0}$, we can get:

$$\boldsymbol{\Sigma}_1 = \frac{diag\{\sum_{j=1}^{\Gamma}(\boldsymbol{N_j}(\tilde{\boldsymbol{F}}_{\boldsymbol{j}} - \boldsymbol{T}_{new}E(\boldsymbol{x_j}))\tilde{\boldsymbol{F}}_{\boldsymbol{j}}^t)\}}{\Gamma} \tag{21}$$

$$\text{Type 1}: \qquad \boldsymbol{\Sigma}_2 = \frac{diag\{\sum_{j=1}^{\Gamma}(n_j(\mathrm{L_j} - \mathbf{W}_{new}E(\mathrm{x_j}))\mathrm{L_j}^t)\}}{\Gamma} \tag{22}$$

Table 1
Complexity of the proposed methods for a single utterance's i-vector extraction (EM iteration number is 6, $log(n_j)$ table index size is 300, time was measured on a Intel I7 CPU with a single thread and 12 GB memory). For LID, GMM size $C = 2048$, feature dimension $D = 56$, $\boldsymbol{T}$ matrix rank $K = 600$, target class number $M = 6$. For SV, GMM size $C = 1024$, feature dimension $D = 36$, $\mathbf{T}$ matrix rank $K = 500$, label vector dimension $M = 2543$, 500 (type 1,2).

Methods	Approximated complexity	Time (LID)	Time (SV)
I-vector	$O(K^3 + K^2C + KCD)$	8 s	2.82 s
Supervised I-vector	$O(K^3 + K^2C + K(CD + M))$	8 s	2.85 s
Simplified I-vector without table	$O(K^3 + KCD)$	0.22 s	0.062 s
Simplified I-vector with table	$O(KCD)$	0.06 s	0.022 s
Simplified supervised I-vector without table	$O(K^3 + K(CD + M))$	0.22 s	0.066 s
Simplified supervised I-vector with table	$O(K(CD + M))$	0.06 s	0.023 s

$$\text{Type 2}: \qquad \boldsymbol{\Sigma}_2 = \frac{diag\{\sum_{j=1}^{\Gamma}(n_j(\mathrm{L_j} - E(\mathrm{x_j}))^t(\mathrm{L_j} - E(\mathrm{x_j})))\}}{\Gamma} \tag{23}$$

These 2 variance vectors describe the energy that cannot be represented by factor analysis and control the importance in the joint optimization objective function (13). $\boldsymbol{\Sigma}_2$ for the type 2 label vectors is just the diagonal elements of the within class covariance matrix in WCCN. After several iterations of EM training, the parameters are learned. For the subsequent supervised i-vector extraction, we let $\boldsymbol{\Sigma}_2$ to be infinity since we do not know the ground truth label information. This will make (14) converge back to (5). After the supervised i-vector extraction, the classification methods steps are the same as in the traditional i-vector modeling.

There are some obvious extensions of this supervised i-vector framework. We can make $\boldsymbol{L}$ as the parameter vector that we want to perform regression with and this can make the proposed framework suitable for regression problems. Moreover, if the classification or regression relation is not linear, we can use non-linear mapping as a preprocessing step before generating $\boldsymbol{L}$.

3.2. Simplified i-vector

I-vector training and extraction is computationally expensive. Consider the GMM size, feature dimension, factor loading matrix size to be C, D, and K, respectively. The complexity for generating a single i-vector is $O(K^3 + K^2C + KCD)$ (Glembek et al., 2011). As shown in Table 1 ($K = 600$, $C = 2048$, $D = 56$), the K^2C term dominates the overall complexity. In this work, we make two approximations to reduce the complexity.

The K^3 term comes from the matrix inversion while the K^2C term is from $\boldsymbol{T^t}\ \boldsymbol{\Sigma^{-1}}\mathrm{N_j T}$ in (5). When C is large, this K^2C term's computational cost is huge. The fundamental reason is that each Gaussian component λ_c has different N_c for each utterance j which means some sub-vectors $\tilde{\boldsymbol{F}}_{\boldsymbol{cj}}$ have less variance than others in $\tilde{\boldsymbol{F}}_{\boldsymbol{j}}$ and need utterance specific intra mean supervector re-weighting in the objective function. We first decompose the $\boldsymbol{N_j}$ vector into $\boldsymbol{N_j} = n_j\boldsymbol{m_j}$ where $n_j = \sum_{c=1}^{C} N_{cj}$, $m_{cj} = N_{cj}/n_j$ and $\sum_{c=1}^{C} m_{cj} = 1$. $\boldsymbol{m_j}$ is the re-weighting vector and n_j (total frame number) controls the confidence at the global level. Our motivation is to re-weight each utterance's mean supervector with its own $(\boldsymbol{m_j})^{1/2}$ before the factor analysis step which makes each dimension of the new supervector $\hat{\boldsymbol{F}}_{\boldsymbol{j}}$ be treated equally in the approximated modeling (25).

$$\hat{\boldsymbol{F}}_{\boldsymbol{c}} = diag([\frac{\sum_{t=1}^{L} P(c|\boldsymbol{y_t}, \lambda)}{\sum_{c=1}^{C}\sum_{t=1}^{L} P(c|\boldsymbol{y_t}, \lambda)}]^{1/2}) \frac{\sum_{t=1}^{L} P(c|\boldsymbol{y_t}, \lambda)(\boldsymbol{y_t} - \boldsymbol{\mu_c})}{\sum_{t=1}^{L} P(c|\boldsymbol{y_t}, \lambda)} = diag([\frac{N_{cj}}{n_j}]^{1/2}) \frac{\sum_{t=1}^{L} P(c|\boldsymbol{y_t}, \lambda)(\boldsymbol{y_t} - \boldsymbol{\mu_c})}{\sum_{t=1}^{L} P(c|\boldsymbol{y_t}, \lambda)}$$
$$\hat{\boldsymbol{F}}_{\boldsymbol{j}} = diag(\boldsymbol{m_j}^{1/2})\tilde{\boldsymbol{F}}_{\boldsymbol{j}} \tag{24}$$

So the intra supervector imbalance is compensated by this pre-weighting, and each utterance is represented by $\hat{\boldsymbol{F}}_{\boldsymbol{j}}$ as the general feature vector and n_j as the confidence value for the subsequent machine learning algorithms. We

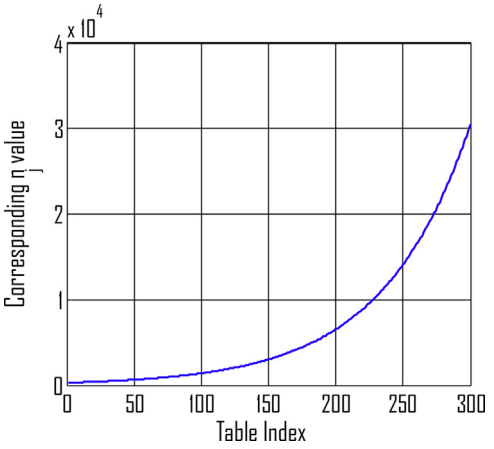

Fig. 3. The n_j quantization curve in the log domain, 300 indexes.

perform factor analysis in the following way by linearly projecting this new normalized supervector $\hat{\boldsymbol{F}}$ on a dictionary $\hat{\boldsymbol{T}}$:

$$\hat{\boldsymbol{F}} = \hat{\boldsymbol{T}}\boldsymbol{x},\ P(\boldsymbol{x_j}) = \mathcal{N}(\boldsymbol{0}, I),\ P(\hat{\boldsymbol{F}}_{\boldsymbol{j}}|\boldsymbol{x_j}) = \mathcal{N}(\hat{\boldsymbol{T}}\boldsymbol{x_j}, \boldsymbol{m_j}\boldsymbol{N_j}^{-1}\boldsymbol{\Sigma}) = \mathcal{N}(\hat{\boldsymbol{T}}\boldsymbol{x_j}, \boldsymbol{m_j}(n_j\boldsymbol{m_j})^{-1}\boldsymbol{\Sigma}) = \mathcal{N}(\hat{\boldsymbol{T}}\boldsymbol{x_j}, n_j^{-1}\boldsymbol{\Sigma}) \tag{25}$$

Therefore, the posterior distribution of the i-vector $\hat{\boldsymbol{x}}$ given the observed $\hat{\boldsymbol{F}}$ is:

$$P(\boldsymbol{x_j}|\hat{\boldsymbol{F}}_{\boldsymbol{j}}) = \mathcal{N}((\boldsymbol{I} + \hat{\boldsymbol{T}}^{\boldsymbol{t}}\boldsymbol{\Sigma}^{-\boldsymbol{1}}n_j\hat{\boldsymbol{T}})^{-1}\hat{\boldsymbol{T}}^{\boldsymbol{t}}\boldsymbol{\Sigma}^{-\boldsymbol{1}}n_j\hat{\boldsymbol{F}}_{\boldsymbol{j}}), (\boldsymbol{I} + \hat{\boldsymbol{T}}^{\boldsymbol{t}}\boldsymbol{\Sigma}^{-\boldsymbol{1}}n_j\hat{\boldsymbol{T}})^{-1}). \tag{26}$$

From the above equation, we can find that the complexity is reduced to $O(K^3 + KCD)$ since n_j is not dependent on any GMM component. By replacing the 1st GMM statistics supervector $\tilde{\boldsymbol{F}}_{\boldsymbol{j}}$ with the pre-normalized supervector $\hat{\boldsymbol{F}}_{\boldsymbol{j}}$ and setting $\boldsymbol{N_j}$ to a scalar n_j, the i-vector training equations become the proposed simplified i-vector solution.

Moreover, since the entire term $(\boldsymbol{I} + \hat{\boldsymbol{T}}^{\boldsymbol{t}}\boldsymbol{\Sigma}^{-\boldsymbol{1}}n_j\hat{\boldsymbol{T}})^{-1}\hat{\boldsymbol{T}}^{\boldsymbol{t}}\boldsymbol{\Sigma}^{-\boldsymbol{1}}$ in (26) only depends on the scalar total frame number n_j, we can create a global table of this quantity against the log value of n_j. The reason to choose log domain is that the smaller the total frame number, the more important it is against the prior. If n_j is very large compared to the prior, then the two n_j get canceled. By enabling the table lookup, the complexity of each utterance's i-vector extraction is further reduced to $O(KCD)$ with a small table index quantization error. The larger the table, the smaller this quantization error. Fig. 3 shows the quantization distance curve. We can see that the quantization error is relatively small when n_j is small.

In this work, we also derived the simplified supervised i-vector modeling's solution as follows:

$$E(\boldsymbol{x_j}) = (\boldsymbol{I} + \hat{\boldsymbol{T}}^{\boldsymbol{t}}\boldsymbol{\Sigma_1^{-1}}n_j\hat{\boldsymbol{T}} + \boldsymbol{W}^{\boldsymbol{t}}\boldsymbol{\Sigma_2^{-1}}n_j\boldsymbol{W})^{-1}(\hat{\boldsymbol{T}}^{\boldsymbol{t}}\boldsymbol{\Sigma_1^{-1}}n_j\hat{\boldsymbol{F}}_{\boldsymbol{j}} + \boldsymbol{W}^{\boldsymbol{t}}\boldsymbol{\Sigma_2^{-1}}n_j\boldsymbol{L_j}), \tag{27}$$

$$E(\boldsymbol{x_j}\boldsymbol{x_j}^t) = E(\boldsymbol{x_j})E(\boldsymbol{x_j})^t + (\boldsymbol{I} + \hat{\boldsymbol{T}}^{\boldsymbol{t}}\boldsymbol{\Sigma_1^{-1}}n_j\hat{\boldsymbol{T}} + \boldsymbol{W}^{\boldsymbol{t}}\boldsymbol{\Sigma_2^{-1}}n_j\boldsymbol{W})^{-1}. \tag{28}$$

$$\boldsymbol{W}_{new} = [\sum_{j=1}^{\Gamma} n_j\boldsymbol{L_j}E(\boldsymbol{x_j}^t)][\sum_{j=1}^{\Gamma} n_jE(\boldsymbol{x_j}\boldsymbol{x_j}^t)]^{-1} \tag{29}$$

$$\hat{\boldsymbol{T}}_{new} = [\sum_{j=1}^{\Gamma} n_j\hat{\boldsymbol{F}}_{\boldsymbol{j}}E(\boldsymbol{x_j}^t)][\sum_{j=1}^{\Gamma} n_jE(\boldsymbol{x_j}\boldsymbol{x_j}^t)]^{-1} \tag{30}$$

$$\boldsymbol{\Sigma}_1 = \frac{diag\{\sum_{j=1}^{\Gamma}(n_j(\hat{\boldsymbol{F}}_{\boldsymbol{j}} - \hat{\boldsymbol{T}}_{new}E(\boldsymbol{x_j}))\hat{\boldsymbol{F}}_{\boldsymbol{j}}^{\ t})\}}{\Gamma} \tag{31}$$

$$\boldsymbol{\Sigma}_2 = \frac{diag\{\sum_{j=1}^{\Gamma}(n_j(\boldsymbol{L_j} - \boldsymbol{W}_{new}E(\boldsymbol{x_j}))\boldsymbol{L_j}^t)\}}{\Gamma} \tag{32}$$

Since the label vector dimensionality $M \ll CD$, the complexity is almost the same as previously shown, $O(CDK)$. It is worth noting that for best accuracy, we only perform approximation using the global table for training purposes. When in testing mode, (27) is still employed (with $\Sigma_2^{-1} = 0$). All the experimental results based on simplified i-vector or simplified supervised i-vector in Section 4 are generated in this way.

3.3. Complexity analysis

The complexity of the proposed methods for a single utterance is shown in Table 1. We use both LID and SV examples to demonstrate the efficiency of the proposed methods. We can see that the proposed simplified and simplified supervised i-vector systems achieve significant complexity cost reduction (by more than a factor of 100) which has a potentially large impact on mobile devices implementation.

4. Experimental Results

In this section, we will evaluate the proposed methods on three different LID and SV databases, namely the DARPA RATS, the NIST LRE 2007 and the NIST SRE 2010. First, we present the corpus, classification tasks, feature extraction and score level fusion in Section 4.1. Then experimental results and discussion are provided in Section 4.2. For the DARPA RATS based LID task, we also provide the details of the USC RATS LID submission system and compare with other state-of-the-art systems.

4.1. Corpus, classification tasks, feature extraction and score level fusion

4.1.1. LID in house evaluation on the DARPA RATS

We first use the DARPA Robust Automatic Transcription of Speech (RATS) data corpus (Walker and Strassel, 2012; Matejka et al., 2012; DARPA, 2012; RATS, 2012; Han et al., 2013)[1] to evaluate the proposed methods for LID under noisy channels. Each speech recording was collected through various degraded, weak and/or noisy communication channels (labeled A through H), all with low varied SNR. The audio file format is 16-kHz 16-bit PCM MS WAV/RIFF with lossless FLAC compression. In this LID task, each testing sample of speech needs to be scored on every target language hypothesis (a target language of interest to be detected). In other words, the task is to decide whether that target language was in fact spoken in the given sample ("yes" or "no") based on an automated analysis of the speech signal. Multiple "yes" answers for a single testing sample are allowed. So we used equal error rate (EER), miss rate with 10% false alarm,[2] minimum average cost C_{avg}[3] and Detection Error Trade-off (DET) curve as the metrics for evaluation.

It can be observed from Fig. 4 that the speech signals can be highly noisy and degraded with significant SNR variability. As a preprocessing step, we performed Speech Activity Detection (SAD) on the raw signal to obtain the speech boundaries. The data set used to train the SAD system is the corpus provided by LDC for the RATS project (DARPA, 2012; RATS, 2012). We used the Multi-Resolution Long-term spectral variability (MR-LTSV) SAD system which uses various resolutions of the LTSV feature originally proposed in Ghosh et al. (2011) and updated in Tsiartas et al. (2013). The classifier we used to classify a frame as speech/non-speech is the K-NN classifier (Cover and Hart, 1967). The parameters of the MR-LTSV features and the number of optimal neighbors of K-NN have been optimized on five randomly picked files for training and five randomly picked files for developing for each channel A-H, given in the RATS corpus. Approximately, the parameters have been optimized on 1 h of training and 1 h of developing data

[1] http://www.darpa.mil/Our_Work/I2O/Programs/Robust_Automatic_Transcription_of_Speech_(RATS).aspx (please copy the full url including those underscores).

[2] This point in the DET curve is required in the task.

[3] C_{avg} is the official evaluation measure for RATS database. C_{avg} is defined in the NIST LRE 2009 evaluation plan with $C_{Miss} = C_{FalseAlarm} = 1$, $P_{target} = 0.5$, $P_{outofset} = 0.2$, http://www.itl.nist.gov/iad/mig/tests/lre/2009/.

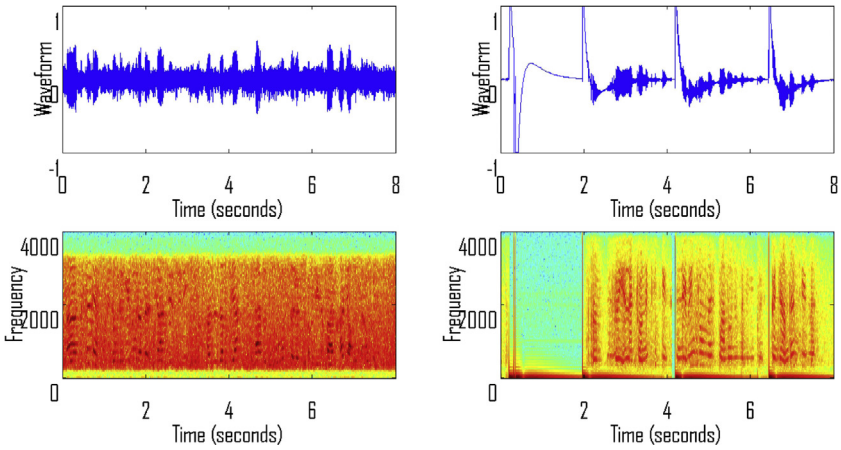

Fig. 4. Waveform and spectrum of two example speech segments in RATS database (dv2_0011_B and dv2_0439_F).

for each channel. In addition, we have used 100 ms frame shift and smoothed the per frame decision using a median filter as described in Ghosh et al. (2011).

We extracted 3 different features on 8K re-sampled valid speech data which have been reported to be useful in the noisy channel LID or SV task, namely MFCC-SDC, MFCC, Gammatone frequency cepstral coefficients (GFCC). For MFCC-SDC feature extraction, a 25 ms Hamming window with 10 ms shifts was adopted. Each utterance was converted into a sequence of 56-dimensional MFCC SDC feature vectors (Torres-Carrasquillo et al., 2002a), each consisting of a 49 SDC (7-1-3-7) features and 7 MFCC coefficients including C0. We also extracted the 36 dimensional MFCC features consisting of 18 MFCC coefficients and their first derivatives. For the robust LID task on noisy data, we applied the GFCC features as an additional auditory-inspired feature set (Shao and Wang, 2008) in conjunction with the conventional MFCCs and SDCs. 44 dimensional GFCC features using 64 filter banks (22 dimensional GFCCs without C0 and their first derivatives) were generated. It is shown in Fig. 5 that GFCC features have more detailed resolution on the low dimensional part of the frequency response and performed better than MFCC features for the

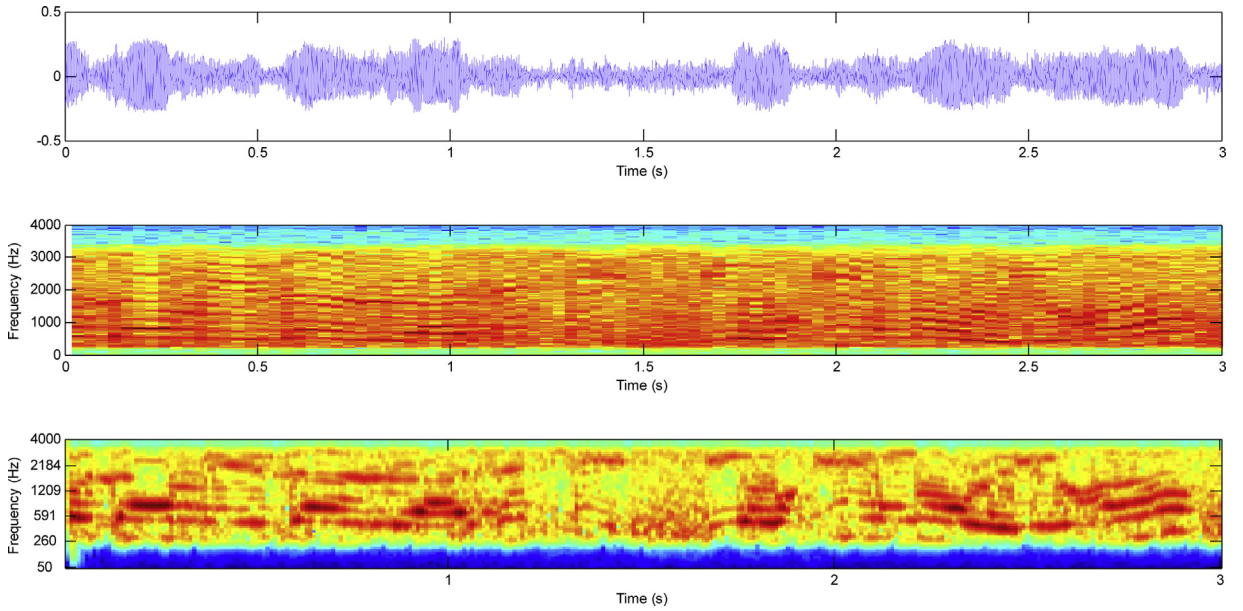

Fig. 5. Waveform (top), spectrum (middle), and cochleagram (bottom) of one speech segment in RATS database.

Table 2
Target and nontarget languages in RATS database.

Target languages	Arabic	Farsi	Pashto	Dari	Urdu
Nontarget languages	English	Mandarin	Spanish	Italian	Thai
	Vietnamese	Russian	Japanese	Bengali	Korean

Table 3
Recording numbers for target and nontarget languages in RATS database dev2 120 s task.

Recording numbers	Arabic	Farsi	Dari	Pashto	Urdu	Other
Train	27453	692	736	19185	12037	19672
Dev2	307	291	59	309	276	672

robust SV task on a variety of low SNR conditions (Shao and Wang, 2008). In this work, when fused with traditional MFCC feature based systems, the overall system performance was found to be enhanced.

LIBLINEAR (Fan et al., 2008) was employed for the logistic regression (LR) based back end classification. The target and nontarget languages are shown in Table 2. The training data for target and nontarget languages are from ldc2011e111 and ldc2012e03 (Matejka et al., 2012), respectively. The training data was also randomly ten times subsampled at the feature level as the UBM training data. The UBM GMM model size is 2048. The dev2 data set from ldc2012e06 is employed as the evaluation data set. Our focus is on the 120 s and 30 s tasks with 1914 and 1782 testing segments, respectively. The recording sample number for each target language in the 120 s task is shown in Table 3. There is a verification trial on every target language for each testing segment which results in 9570 and 8910 trials in total. The major parameters in this work are defined in Table 1.

Due to the limited amount of development data, we simply employed the weighted summation fusion approach with parameters tuned by cross validation. Let there be G input subsystems where the ith subsystem outputs its own posterior probability vector $l_i(\boldsymbol{y})$ for every trial. Then the fused score vector $\acute{l}(\boldsymbol{y})$ is given by:

$$\acute{l}(\boldsymbol{y}) = \sum_{i=1}^{G} \eta_i l_i(\boldsymbol{y}) \tag{33}$$

The weight, η_k, can be tuned by validation data. For the LR system in the LID task, log-likelihood normalization was adopted to map the log likelihood scores into the posterior probabilities (Li et al., 2012).

It is worth noting that in our official submission system (described in Section 4.1.2), the linear logistic regression based fusion module in the FoCal toolkit (Brümmer, 2007) is adopted to increase the performance.

4.1.2. USC LID system for RATS evaluation phase 2

In this section, we provide the details of our USC LID submission system for RATS evaluation phase 2 which is also based on the proposed simplified supervised i-vector representation. Compared to the in house evaluation system described in the previous section, this submission system employed the Wiener filter and IBM VAD as front end processing, and replaced the LR kernel from linear to polynomial for back end classification. Furthermore, spectro-temporal Gabor features are also included as an additional feature set and fused at the score level. Linear logistic regression in the FoCal toolkit (Brümmer, 2007) is adopted here for fusion. All these modifications together lead to competitive performance and we adopt this result to compare with other start-of-the-art systems for all 4 duration conditions in Table 9.

4.1.3. LID on the NIST LRE 2007

In addition to the noisy channels based DARPA RATS data, we adopted the 2007 NIST Language Recognition Evaluation (LRE) (NIST, 2007) 30 s closed set general task as the evaluation database for clean channel LID. There are 14 target languages and the valid speech in each testing utterance is about 30 s long. Data of target languages from Call Friend, OGI Multilingual, OGI 22 languages, NIST LRE 1996, NIST LRE 2003, NIST LRE 2005, NIST LRE 2007 supplemental training as well as a subset of NIST SRE 2004–2006 were used as our training data. We first extracted the same 56 dimensional MFCC-SDC feature as described in Section 4.1.1; then employed a Czech phoneme recognizer

Table 4
Corpora used to estimate the UBM, total variability matrix, JFA factor loading matrix, WCCN, LDA, PLDA and the normalization data for NIST SRE 2010 task condition 5.

	Switchboard	NIST04	NIST05	NIST06	NIST08
UBM		√	√		
T	√	√	√	√	√
JFA V	√				
JFA U		√	√	√	√
JFA D		√			
WCCN	√	√	√	√	√
LDA		√	√	√	√
PLDA		√	√	√	√
Znorm		√	√		
Snorm					√
Tnorm				√	

(Schwarz et al., 2006) to perform speech activity detection by simply dropping all feature frames that are decoded as silence or noises. We divided the features of each training conversation into multiple 30 s (3000 frames) segments. There are totally 81848 training segments, 2158 testing utterances, and 30212 testing trials. A 2048 components GMM UBM model was trained from 20000 training segments randomly selected from the training data. After GMM statistics vectors were calculated, the proposed simplified i-vector, simplified supervised i-vector as well as the i-vector baseline were applied. The back end variability compensation method (WCCN) and the classification method (both linear and 2nd polynomial kernel LR) are the same as in the DARPA RATS experiments. There is no score level fusion since we only adopted one feature set and one classifier. The performance is reported in optimum average cost C_{avg} value as suggested by NIST (2007).

4.1.4. Speaker verification on the NIST SRE 2010

We also performed experiments on the NIST SRE 2010 (NIST, 2010) for SV tasks. Our focus is the female part of the common condition 5 (a subset of tel-tel) in the core task. We used EER, the normalized old minimum decision cost value (norm old minDCF) and norm new minDCF as the metrics for evaluation (NIST, 2010). For cepstral feature extraction, a 25 ms Hamming window with 10 ms 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 applied the same Czech phoneme recognizer (Schwarz et al., 2006) as the one employed in our NIST LRE 2007 experiments for VAD. Feature warping is applied to mitigate variabilities.

The training data for the NIST SRE 2010 task included Switchboard II part1 to part3 and NIST SRE 2004, 2005, 2006 and 2008 corpora on the telephone channel. The description of the dataset used in each step is provided in Table 4. The gender-dependent GMM UBMs consist of 1024 mixture components. The JFA baseline system (Kenny et al., 2007a,b, 2008) is trained using the BUT toolkit (Burget et al., 2008) and linear common channel point estimate scoring (Glembek et al., 2009) is adopted. The speaker factor size and channel factor size is 300 and 100, respectively. ZTnorm was applied on the JFA subsystem while Snorm was employed in the i-vector subsystem. The PLDA implementation is based on the UCL toolkit (Prince and Elder, 2007) where the sizes of speaker loading matrix **U** and variability loading matrix **G** are 250 and 80, respectively. Simple weighted linear summation is adopted here as the score level fusion. Other parameter settings are reported in the caption of Table 1.

4.2. Results and discussion

4.2.1. LID in house evaluation on the DARPA RATS

The performances of the proposed methods on three feature sets for in house LID evaluation are shown in Tables 5 and 6. First, we can observe that WCCN worked well to compensate the variability for most systems. This makes sense since there are 8 highly degraded communication channels for each language. Second, the simplified i-vector achieved comparable results to the i-vector baseline showing only a small degradation in classification. Third, the simplified supervised i-vector outperformed the simplified i-vector dramatically.

Table 5
Performance of the proposed methods with LR modeling for LID RATS dev2 120 s task.

ID	Method	WCCN	EER/P_{miss} at 10% P_{fa}/min C_{avg} (%)		
			MFCC-SDC 56dim	MFCC 36dim	GFCC 44dim
1	I-vector		6.6/4.5/9.02	5.7/4.2/8.05	6.1/4.3/8.27
2	I-vector	√	6.3/4.4/8.69	5.7/3.9/8.06	5.8/3.9/8.04
3	Simplified I-vector		7.6/6.7/9.85	7.0/5.3/8.97	7.3/6.1/8.97
4	Simplified I-vector	√	6.2/4.4/8.77	5.7/4.2/7.85	6.3/4.4/8.54
5	Simplified supervised I-vector type 1		6.6/4.9/8.99	**5.1/3.5/7.73**	**5.6/3.8/8.01**
6	Simplified supervised I-vector type 1	√	6.2/4.5/8.52	**4.8/3.0/7.49**	**5.4/3.7/7.92**

Table 6
Performance of the proposed methods with LR modeling for LID RATS dev2 30 s task.

ID	Method	WCCN	EER/P_{miss} at 10% P_{fa}/min C_{avg} (%)		
			MFCC-SDC 56dim	MFCC 36dim	GFCC 44dim
1	I-vector		12.5/15.5/14.12	12.1/13.6/13.27	13.0/15.8/14.47
2	I-vector	√	12.9/16.5/15.13	12.0/14.1/13.42	13.4/16.2/14.45
3	Simplified I-vector		14.1/19.0/15.32	12.7/14.2/13.28	13.0/15.6/14.15
4	Simplified I-vector	√	13.1/16.0/15.31	11.4/12.3/13.19	12.0/14.0/13.44
5	Simplified supervised I-vector type 1		13.9/17.6/16.00	**11.4/13.0/13.20**	**12.0/14.2/13.99**
6	Simplified supervised I-vector type 1	√	12.7/16.2/14.58	**11.1/11.8/13.01**	**11.7/13.7/13.73**

Before WCCN was applied, the simplified supervised i-vector systems on all three feature sets outperformed the simplified i-vector systems by more than 10% relatively in terms of EER. After WCCN was adopted for variability compensation, for MFCC 36 dimensional features and GFCC 44 dimensional features, the simplified supervised i-vector still achieved 10%–20% relative error reduction and more than 100 times faster compared to the i-vector baseline. For MFCC-SDC 56 dimensional features, the improvement is more reflected on the min C_{avg} cost values which servers as the official performance measure for RATS.

Moreover, we also observed significant error reduction by fusing 3 feature sets based systems together in Table 7. Fig. 7 shows the DET curves for each feature set as well as the 3 feature set fusion. The fusion system achieved 3.7% EER and 7.6% EER for 120 s and 30 s tasks, respectively.

4.2.2. *USC LID submission system for RATS evaluation phase 2*

In this subsection, we show the results of the USC RATS phase 2 LID submission System on the dev2 sets for all four duration settings in Table 8. As shown in Fig. 6 and Section 4.1.2, this system still utilized the proposed simplified supervised i-vector for representation and adopted 4 different feature sets. Compared to the in house evaluation system, this system has better front end processing (Wiener filtering), VAD schemes, newly included spectro-temporal Gabor features, polynomial kernel based LR classification and linear logistic regression based fusion methods. All these modifications lead to a significant performance improvement compared to the aforementioned in house evaluation.

In Table 8, we can observe that score level fusion reduced the error rate dramatically the same way as in the previous in house evaluation which demonstrates that these different feature sets are complementary. Fig. 8 shows the EER performance of each individual channel in the LID RATS dev2 120 s task. In this task, there is no testing file

Table 7
Performance of score level fusion with systems from multiple features for LID RATS dev2 120 s and 30 s task.

Durations	EER/P_{miss} at 10% P_{fa}/min C_{avg} (%)			
	MFCC-SDC	MFCC	GFCC	Fusion 3 features
120 s	6.2/4.5/8.52	4.8/3.0/7.49	5.4/3.7/7.92	**3.7/1.9/6.41**
30 s	12.7/16.2/14.58	11.1/11.8/13.01	11.7/13.7/13.73	**7.6/6.5/9.73**

Table 8
Performance of the USC LID phase 2 submission system for LID RATS dev2 task.

Methods	EER/P_{miss} at 10% P_{fa}/min C_{avg} (%)			
	120 s	30 s	10 s	3 s
MFCC	4.0/2.1/6.7	8.0/7.4/9.7	12.4/14.9/15.3	19.2/33.0/19.9
MFCC-SDC	4.0/2.3/4.5	9.7/9.5/12.3	14.4/19.8/16.7	22.1/37.1/22.8
GFCC	3.5/1.9/6.3	7.3/5.9/9.3	13.1/16.4/15.1	16.8/25.9/18.7
Gabor	4.1/2.7/6.8	8.9/8.1/10.9	15.2/20.6/17.9	19.2/30.2/21.3
Fusion	**2.6/0.7/4.8**	**5.2/3.3/7.2**	**8.6/7.7/10.2**	**15.5/21.4/15.7**

from channel "D" and "all" means the combined standard testing scenario whose results are the same as the ones in Table 8. First, we can see that fusion improves the performance not only for the combined "all" setting but also for each individual noisy channel which indicates the robustness of different feature sets fusion. Second, each feature set performs differently on different channels and no single one is always better than others. For example, MFCC-SDC outperforms MFCC on channel "A,C,H" but vice versa on channels "E,F,H"; GABOR feature performs the best on channels "A,E" but the worst on channels "B,H". This diverse performance pattern makes the fusion work and motivates us to consider channel specific fusion in our future work.

Compared to the state-of-the-art systems in Table 9, the proposed methods achieved comparable and competitive results for all four durations. Furthermore, by further combining with IBM's systems at the score level, the IBM-USC joint system achieved 5%–13% relative EER reduction against the latest IBM system.

4.2.3. LID on the NIST LRE 2007

The performance results on the NIST LRE 2007 30 s general task are shown in Table 10. We can find out that the proposed simplified supervised i-vector modeling outperformed the i-vector baseline in both recognition performance

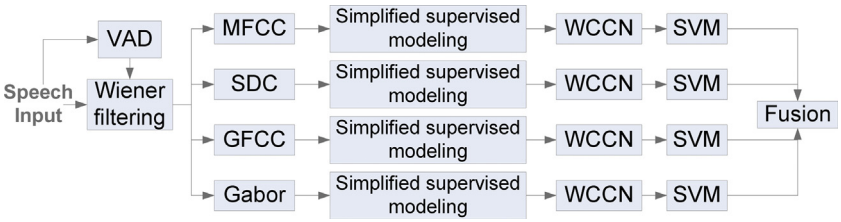

Fig. 6. USC RATS phase 2 LID submission system overview.

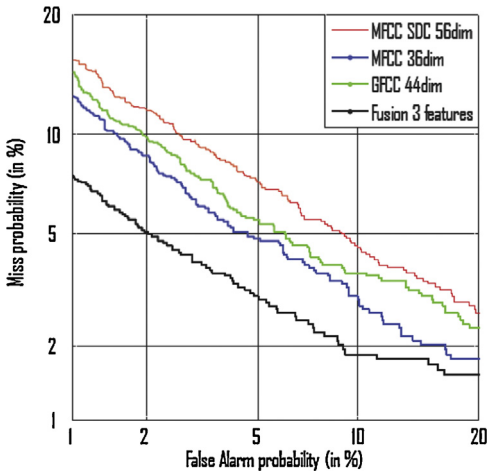

Fig. 7. DET curves for LID performance in Table 7 (120 s).

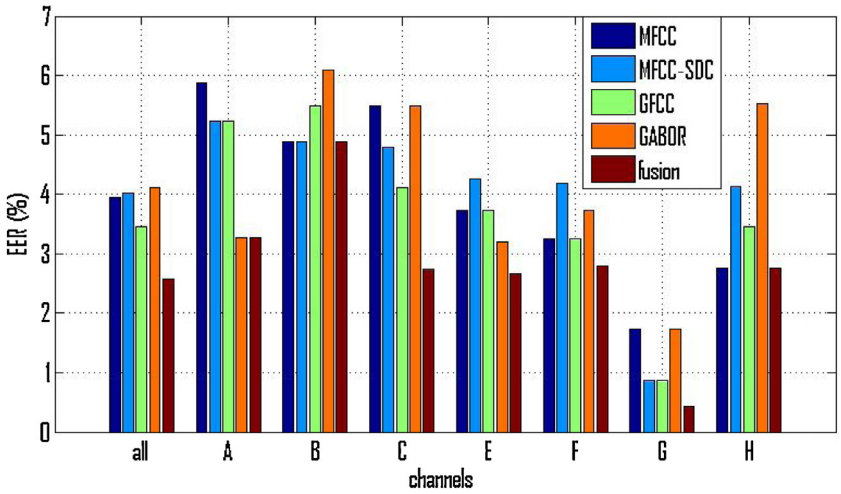

Fig. 8. Performance of the USC LID phase 2 submission system for LID RATS dev2 120 s task on different channels.

and computational cost under the clean channel condition and with more target languages. Furthermore, 2nd order polynomial kernel performed better than the linear kernel in the logistic regression based back end classification.

4.2.4. *Speaker verification on the NIST SRE 2010*

The results of the i-vector baseline and the proposed supervised i-vector, simplified i-vector as well as the simplified supervised i-vector methods for SV tasks are shown in Table 11. We can observe that LDA, PLDA, and Snorm contributed to increase the performance for all the systems. WCCN reduced the EER by more than 40% for all systems except the type 2 simplified supervised i-vector (T2SSIV). For T2SSIV, WCCN is not that important since the label regularized joint optimization already includes the within class covariance in the objective function (see (11, 13, 14, 23)). This was reflected by a 30% EER reduction (I-vector 9.02%, T2SSIV 6.45%) in the cosine distance raw scoring without any back end processing.

Table 9
Performance comparison for LID RATS dev2 tasks (∅ denotes not evaluated or reported).

Methods	EER/P_{miss} at 10% P_{fa}/min C_{avg} (%)			
	120 s	30 s	10 s	3 s
Acoustic I-vector BBN Phase 1 (Matejka et al., 2012)	∅/∅/9.03	∅/∅/11.96	∅/∅/17.83	∅/∅/27.10
Acoustic I-vector BUT Phase 1 (Matejka et al., 2012)	∅/∅/7.83	∅/∅/9.97	∅/∅/14.52	∅/∅/21.46
BBN fusion system Phase 1 (Matejka et al., 2012)	3.30/0.72/6.56	6.16/3.9/8.33	9.57/9.32/11.40	17.14/25.81/17.45
IBM fusion system Phase 1 without USC (Omar et al., 2012)	2.9/∅/∅	6.9/∅/∅	12.7/∅/∅	18.9/∅/∅
USC LID system (Section 4.1.2)	2.6/0.7/4.8	5.2/3.3/7.2	8.6/7.7/10.2	15.5/21.4/15.7
IBM fusion system Phase 2 **without USC** (Han et al., 2013)	1.9/∅/∅	3.7/∅/∅	6.1/∅/∅	11.4/∅/∅
IBM-USC joint system Phase 2 **with USC** (Han et al., 2013)	1.8/∅/∅	3.2/∅/∅	5.6/∅/∅	10.0/∅/∅

Table 10
Performance on the NIST LRE 2007 general language recognition 30 s task with WCCN.

Methods	Linear kernel		2nd order polynomial kernel	
	EER (%)	min C_{avg} (%)	EER (%)	min C_{avg} (%)
I-vector	2.83	2.97	2.73	2.73
Simplified I-vector	3.06	3.29	2.64	2.87
Simplified supervised I-vector	2.59	2.77	2.59	2.61

Table 11
Performance of the proposed methods for the NIST SRE 2010 task female part condition 5 (T1SSIV: type 1 simplified supervised i-vector, T2SSIV: type 2 simplified supervised i-vector).

Method	LDA	WCCN	PLDA	S norm	EER%	Norm minDCF	
						New	Old
I-vector					9.02	0.724	0.409
I-vector	250			√	7.87	0.668	0.307
I-vector	250	√		√	3.91	0.454	0.190
I-vector	250	√	√	√	**3.37**	**0.415**	**0.165**
Supervised i-vector type 1	250			√	7.64	0.640	0.278
Supervised i-vector type 1	250	√		√	4.01	0.425	0.170
Supervised i-vector type 1	250	√	√	√	**2.95**	**0.420**	**0.154**
Simplified i-vector					8.94	0.758	0.374
Simplified i-vector	250			√	7.96	0.696	0.311
Simplified i-vector	250	√		√	4.79	0.527	0.213
Simplified i-vector	250	√	√	√	**3.45**	**0.545**	**0.192**
Simplified supervised i-vector type 1					8.65	0.746	0.341
Simplified supervised i-vector type 1	250			√	7.06	0.654	0.289
Simplified supervised i-vector type 1	250	√		√	3.95	0.518	0.197
Simplified supervised i-vector type 1	250	√	√	√	**3.13**	**0.541**	**0.176**
Simplified supervised i-vector type 2					6.45	0.645	0.285
Simplified supervised i-vector type 2	250			√	5.35	0.575	0.228
Simplified supervised i-vector type 2	250	√		√	4.51	0.549	0.195
Simplified supervised i-vector type 2	250	√	√	√	**3.06**	**0.569**	**0.179**
Simplified supervised i-vector type 2	250		√	√	3.08	0.581	0.189

Furthermore, type 1 supervised i-vector (T1SUP-IV) and type 1 simplified supervised I-vector (T1SSIV) outperformed i-vector and simplified i-vector by 5%-10% relatively for all the modeling configurations (3.37% and 3.45% EER vs 2.95% and 3.13% EER). Also as shown in Table 12 (ID 6 vs 5), after fusing with JFA baseline, supervised i-vector still outperformed the i-vector baseline by 9% relative EER reduction. Therefore, adding label information in the i-vector training indeed improves the performance. The lower improvement of T2SSIV compared to T1SSIV might be due to the diagonal version of $\mathbf{\Sigma}_2$ against the triangular WCCN matrix.

Moreover, simplified supervised i-vector systems (T1SSIV and T2SSIV) achieved better EER but worse norm cost compared to the i-vector baseline. However, the computational cost is reduced by a factor of 120. And after fusing with the JFA system (Table 12 ID 7 vs 5), this gap is reduced to only 3% to 6% relatively. Therefore, simplified supervised i-vector has the potential to replace the computationally expensive i-vector baseline when fusing with the JFA system.

It is worth noting that the supervised version of all the systems only performed better on EER and norm old minDCF values. How to further reduce the norm new minDCF is our current focus. Future work also includes applying the non-simplified type 2 supervised i-vector as well as evaluating different label vector designs.

Table 12
Performance of the proposed systems in fusion.

ID	Systems	EER%	Norm minDCF	
			New	Old
1	JFA linear scoring ZTnorm	3.62	0.414	0.193
2	I-vector LDA WCCN PLDA Snorm	3.37	0.415	0.165
3	Supervised i-vector type 1 LDA WCCN PLDA Snorm	2.95	0.420	0.154
4	Simplified supervised i-vector type 1 LDA WCCN PLDA Snorm	3.13	0.541	0.176
5	Fusion ID 1 +ID 2	2.77	0.372	0.152
6	**Fusion ID 1 +ID 3**	**2.53**	**0.370**	**0.146**
7	Fusion ID 1 +ID 4	2.82	0.377	0.162

4.2.5. Different roles of simplified and supervised i-vectors in the system

Simplified i-vector and supervised i-vector play different roles in the system and behave differently in terms of error rate and computational cost. First, from Table 1, we can observe that simplified i-vector achieved significant complexity cost reduction (by more than a factor of 100). But it does not improve the recognition performance and in most cases it slightly degrades the baseline result due to the simplification. In Table 11, after all the back end processing (LDA, WCCN, PLDA, Snorm), simplified i-vector degraded the EER results by relatively 2.4% compared to the i-vector baseline. Second, the performance improvements come from the supervised i-vector. In Table 11 after back end processing, supervised i-vector alone achieved 12.5% relative EER reduction (3.37% to 2.95%) while simplified supervised i-vector only reduced the baseline EER by relatively 9.2% (3.37% to 3.06%) but with more than 100 times increase in speed. Third, by comparing the results of system ID 5 and 6 in Table 12, we discover that the performance improvement from supervised i-vector is preserved after the fusion with JFA. Finally, these two techniques could be combined together as simplified supervised i-vector to achieve comparable or better recognition performance against the i-vector baseline and at the same time reduce the computational cost by a factor of more than 100.

5. Conclusions

This paper presented a robust and efficient approach using simplified and supervised i-vector modeling with applications to automatic language identification (LID) and speaker verification (SV). Several new algorithmic and computational modeling ideas are proposed. First, by concatenating the label vector and the linear regression matrix at the end of the mean supervector and the i-vector factor loading matrix, respectively, the traditional i-vectors are extended to the label-regularized supervised i-vectors. These supervised i-vectors are optimized not only to reconstruct the mean supervectors well but also to minimize the mean square error between the original and the reconstructed label vectors; this can thus make the supervised i-vectors become more discriminative in terms of the regularized label information. Second, factor analysis (FA) is performed on the pre-normalized GMM first order statistics supervector to ensure each gaussian component's statistics sub-vector is treated equally in the FA which reduces the computational cost by a factor of 25. Further computational improvement is obtained by pre-computing a global cache table of the resulting matrices against the total frame numbers' log values. By using the lookup table, each utterance's i-vector extraction is further sped up by another factor of 4 with just a small table index quantization error. The larger the table, the smaller this quantization error. Third, simplified and supervised i-vector modeling can be used together as the simplified supervised i-vector which outperforms the i-vector baseline in terms of both robustness and efficiency. The solutions for training and extracting these simplified version i-vectors are provided for both the traditional unsupervised and the proposed supervised i-vectors. Finally, Gammatone frequency cepstral coefficients (GFCC) and Gabor features are adopted as yet additional (complementary) auditory-inspired and spectro-temporal features for LID. By fusing GFCC and Gabor features with the traditional MFCC and SDC features based systems, overall LID performance is improved significantly.

References

Aronowitz, H., Barkan, O., 2012. Efficient approximated i-vector extraction. In: Proc. ICASSP, pp. 4789–4792.

Bahari, M.H., McLaren, M., Hamme, H., Leeuwen, D., 2012. Age estimation from telephone speech using i-vectors. In: Proc. INTERSPEECH.

Brümmer, N., 2007. Focal multi-class: toolkit for evaluation, fusion and calibration of multi-class recognition scores tutorial and user manual, Software available at http://sites.google.com/site/nikobrummer/focalmulticlass

Burget, L., Fapšo, M., Hubeika, V., 2008. But system description: Nist sre 2008. In: NIST Speaker Recognition Evaluation Workshop, pp. 1–4, Software available at http://speech.fit.vutbr.cz/software/joint-factor-analysis-matlab-demo

Burget, L., Matejka, P., Cernocky, J., 2006. Discriminative training techniques for acoustic language identification. In: Proc. ICASSP.

Campbell, W., Sturim, D., Reynolds, D., 2006. Support vector machines using GMM supervectors for speaker verification. IEEE Signal Processing Letters 13, 308–311.

Castaldo, F., Colibro, D., Dalmasso, E., Laface, P., Vair, C., 2007. Compensation of nuisance factors for speaker and language recognition. IEEE Transactions on Audio, Speech, and Language Processing 15, 1969–1978.

Cover, T., Hart, P., 1967. Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13, 21–27.

DARPA, RATS project, 2012. http://www.darpa.mil/Our_Work/I2O/Programs/Robust_Automatic_Transcription_of_Speech_(RATS).aspx

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P., 2011a. Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing 19, 788–798.

Dehak, N., Torres-Carrasquillo, P., Reynolds, D., Dehak, R., 2011b. Language recognition via i-vectors and dimensionality reduction. In: Proc. INTERSPEECH, pp. 857–860.
Deng, L., Li, X., 2013. Machine learning paradigms for speech recognition: an overview. IEEE Transactions on Audio, Speech, and Language Processing 21, 1060–1089.
Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C., 2008. Liblinear: a library for large linear classification. The Journal of Machine Learning Research 9, 1871–1874.
Gauvain, J., Messaoudi, A., Schwenk, H., 2004. Language recognition using phone lattices. In: Proc. ICSLP.
Ghosh, P., Tsiartas, A., Georgiou, P.G., Narayanan, S., 2011. Robust voice activity detection using long-term signal variability. IEEE Transactions Audio, Speech, and Language Processing 19, 600–613.
Glembek, O., Burget, L., Dehak, N., Brummer, N., Kenny, P., 2009. Comparison of scoring methods used in speaker recognition with joint factor analysis. In: Proc. ICASSP, pp. 4057–4060.
Glembek, O., Burget, L., Matejka, P., Karafiát, M., Kenny, P., 2011. Simplification and optimization of i-vector extraction. In: Proc. ICASSP, pp. 4516–4519.
Han, K., Ganapathy, S., Li, M., Omar, M., Narayanan, S., 2013. Trap language identification system for rats phase ii evaluation. In: Proc. INTERSPEECH.
Hatch, A., Kajarekar, S., Stolcke, A., 2006. Within-class covariance normalization for SVM-based speaker recognition. In: Proc. INTERSPEECH, pp. 1471–1474.
Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al., 2012. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Processing Magazine 29, 82–97.
Kenny, P., 2010. Bayesian speaker verification with heavy tailed priors. In: Proc. ODYSSEY.
Kenny, P., Boulianne, G., Dumouchel, P., 2005. Eigenvoice modeling with sparse training data. IEEE Transactions on Speech and Audio Processing 13, 345–354.
Kenny, P., Boulianne, G., Dumouchel, P., Ouellet, P., 2007a. Speaker and session variability in GMM-based speaker verification. IEEE Transactions on Audio, Speech and Language Processing 15, 1448–1460.
Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P., 2007b. Joint factor analysis versus eigenchannels in speaker recognition. IEEE Transactions on Audio, Speech, and Language Processing 15, 1435–1447.
Kenny, P., Ouellet, P., Dehak, N., Gupta, V., Dumouchel, P., 2008. A study of interspeaker variability in speaker verification. IEEE Transactions on Audio, Speech, and Language Processing 16, 980–988.
Kleinschmidt, M., Gelbart, D., 2002. Improving word accuracy with gabor feature extraction. In: Proc. INTERSPEECH.
Lei, H., Meyer, B.T., Mirghafori, N., 2012. Spectro-temporal gabor features for speaker recognition. In: Proc. ICASSP, IEEE, pp. 4241–4244.
Li, H., Ma, B., Lee, C., 2007a. A vector space modeling approach to spoken language identification. IEEE Transactions on Audio, Speech, and Language Processing 15, 271–284.
Li, H., Ma, B., Lee, K.A., 2013. Spoken language recognition: from fundamentals to practice. In: Proceedings of the IEEE 101, pp. 1136–2115.
Li, M., Han, K., Narayanan, S., 2012. Automatic speaker age and gender recognition using acoustic and prosodic level information fusion. Computer Speech & Language.
Li, M., Suo, H., Wu, X., Lu, P., Yan, Y., 2007b. Spoken language identification using score vector modeling and support vector machine. In: Proc. INTERSPEECH, pp. 350–353.
Martinez, D., Plchot, O., Burget, L., Glembek, O., Matejka, P., 2011. Language recognition in ivectors space. In: Proc. INTERSPEECH, pp. 861–864.
Matejka, P., Glembek, O., Castaldo, F., Alam, M., Plchot, O., Kenny, P., Burget, L., Cernocky, J., 2011. Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification. In: Proc. ICASSP, pp. 4828–4831.
Matejka, P., Plchot, O., Soufifar, M., Glembek, O., DHaro, L., Vesely, K., Grezl, F., Ma, J., Matsoukas, S., Dehak, N., 2012. Patrol team language identification system for DARPA RATS p1 evaluation. In: Proc. INTERSPEECH.
NIST, 2007. The 2007 NIST language recognition evaluation. http://www.itl.nist.gov/iad/mig/tests/lre/2007/
NIST, 2010. The NIST 2010 Speaker Recognition Evaluation Plan, www.itl.nist.gov/iad/mig/tests/spk/2010/index.html.
Omar, M., Ganapathy, S., Han, K., Pelecanos, J., Yaman, S., Zhu, W., Roukos, S., 2012. Targeted robust audio processing (trap) system rats phase i review. In: RATS PI Meeting, Washington, DC, May.
Prince, S., Elder, J., 2007. Probabilistic linear discriminant analysis for inferences about identity. In: Proc. ICCV, pp. 1–8.
Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. In: Proceedings of the IEEE, pp. 257–328.
RATS, DARPA Rats corpus, 2012. https://secure.ldc.upenn.edu/intranet/dataMatrixGenerate.jsp
Schuller, B., Steidl, S., Batliner, A., Burkhardt, F., Devillers, L., Müller, C., Narayanan, S., 2013. Paralinguistics in speech and language state-of-the-art and the challenge. Computer Speech & Language 27, 4–39.
Schwarz, P., Matejka, P., Cernocky, J., 2006. Hierarchical structures of neural networks for phoneme. In: Proc. ICASSP, pp. 325–328, Software available at http://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context.
Shao, Y., Wang, D., 2008. Robust speaker identification using auditory features and computational auditory scene analysis. In: Proc. ICASSP, pp. 1589–1592.
Siniscalchi, S.M., Reed, J., Svendsen, T., Lee, C.H., 2010. Exploiting context-dependency and acoustic resolution of universal speech attribute models in spoken language recognition. In: Proc. INTERSPEECH, pp. 2718–2721.
Torres-Carrasquillo, P., Singer, E., Kohler, M., Greene, R., Reynolds, D., Deller Jr., J., 2002a. Approaches to language identification using gaussian mixture models and shifted delta cepstral features. In: Proc. ICSLP, pp. 89–92.
Torres-Carrasquillo, P.A., Reynolds, D.A., Deller, J.R., 2002b. Language identification using gaussian mixture model tokenization. In: Proc. ICASSP, IEEE.

Tsiartas, A., Chaspari, T., Katsamanis, A., Ghosh, P.K., Li, M., Van Segbroeck, M., Potamianos, A., Narayanan, S.S., 2013. Multi-band long-term signal variability features for robust voice activity detection. In: Proc. INTERSPEECH.

Walker, K., Strassel, S., 2012. The rats radio traffic collection system. In: Proc. of Odyssey.

Yan, Y., Barnard, E., 1995. An approach to automatic language identification based on language-dependent phone recognition. In: Proc. ICASSP, pp. 3511–3514.

Zhao, X., Wang, D., 2013. Analyzing noise robustness of MFCC and GFCC features in speaker identification. In: Proc. ICASSP, pp. 7204–7208.

Zissman, M., 1995. Language identification using phoneme recognition and phonotactic language modeling. In: Proc. ICASSP, pp. 3503–3506.