Non-Negative Factor Analysis of Gaussian Mixture Model Weight Adaptation for Language and Dialect Recognition

Mohamad Hasan Bahari, Najim Dehak, Hugo Van hamme, Lukas Burget, Ahmed M. Ali, and Jim Glass

***Abstract*—Recent studies show that Gaussian mixture model (GMM) weights carry less, yet complimentary, information to GMM means for language and dialect recognition. However, state-of-the-art language recognition systems usually do not use this information. In this research, a non-negative factor analysis (NFA) approach is developed for GMM weight decomposition and adaptation. This modeling, which is conceptually simple and computationally inexpensive, suggests a new low-dimensional utterance representation method using a factor analysis similar to that of the i-vector framework. The obtained subspace vectors are then applied in conjunction with i-vectors to the language/dialect recognition problem. The suggested approach is evaluated on the NIST 2011 and RATS language recognition evaluation (LRE) corpora and on the QCRI Arabic dialect recognition evaluation (DRE) corpus. The assessment results show that the proposed adaptation method yields more accurate recognition results compared to three conventional weight adaptation approaches, namely maximum likelihood re-estimation, non-negative matrix factorization, and a subspace multinomial model. Experimental results also show that the intermediate-level fusion of i-vectors and NFA subspace vectors improves the performance of the state-of-the-art i-vector framework especially for the case of short utterances.**

***Index Terms*—Non-negative factor analysis, model adaptation, Gaussian mixture model weight, dialect recognition, language recognition.**

Manuscript received December 15, 2013; revised April 07, 2014; accepted April 07, 2014. Date of publication April 22, 2014; date of current version May 16, 2014. The works of M. H. Bahari and H. Van hamme were supported by the European Commission through the Marie-Curie ITN-project, Bayesian Biometrics for Forensics and the FWO as a travel grant for a long stay abroad. The work of N. Dehak was supported in part by the Defense Advanced Research Projects Agency (DARPA). The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Shinji Watanabe.

M. H. Bahari and H. Van hamme are with the Center for Processing Speech and Images, KU Leuven, 3001 Leuven, Belgium (e-mail: mohamadhasan.bahari@esat.kuleuven.be; hugo.vanhamme@esat.kuleuven.be).

N. Dehak and J. Glass are with the MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139 USA (e-mail: najim@csail.mit.edu; glass@csail.mit.edu).

L. Burget is with Speech@FIT, Brno University of Technology, 612 66 Brno, Czech Republic (e-mail: burget@fit.vutbr.cz).

A. M. Ali is with Qatar Computing Research Institute, Doha, Qatar (e-mail: amali@qf.org.qa).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2014.2319159

I. Introduction

LANGUAGE and dialect/accent recognition has received increased attention during the recent decades due to its importance for the enhancement of automatic speech recognition (ASR) [1], [2], multi-language translation systems, service customization, targeted advertising, and forensics softwares [3], [4].

Although research on text-independent language/dialect identification started in the early 1970s [5], [6], it remains a challenging task due to similarities of acoustic phonetics, phonotactics, and prosodic cues across different languages/dialects. Furthermore, in many practical cases we have no control over the available speech duration, channel characteristics, and noise level.

Recent language/dialect recognition techniques can be divided into phonotactic, and acoustic approaches [7]. Since phonotactic features and acoustic (spectral and/or prosodic) features provide complementary cues, state-of-the-art methods usually apply a combination of both through a fusion of their output scores [7]. A phone recognizer followed by language models (PRLM), parallel PRLM (PPRLM) and support vector machines PRLM techniques developed within the language recognition area, are successful phonotactic methods focusing on phone sequences as an important characteristic of different accents [8], [9].

The acoustic approaches, which are the main focus of this paper, enjoy the advantage of requiring no specialized language knowledge [7]. One effective acoustic method for accent recognition involves modeling speech recordings with Gaussian mixture model (GMM) mean supervectors before using them as features in a support vector machine (SVM) [7]. Similar Gaussian mean supervector techniques have been successfully applied to different speech analysis problems such as speaker recognition [10]. While effective, these features are of a high dimensionality resulting in high computational cost and difficulty in obtaining a robust model in the context of limited data. In the field of speaker recognition, recent advances using so-called i-vectors [11] have increased the classification accuracy considerably. The i-vector framework, which provides a compact representation of an utterance in the form of a low-dimensional feature vector, applies a simple factor analysis on GMM means. The same idea was also effectively applied in language/dialect recognition and speaker age estimation [12]–[14].

2329-9290 © 2014 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Recent studies show that GMM weights, which entail a lower dimension compared to Gaussian mean supervectors, carry less, yet complimentary, information to GMM means [14]–[16]. Zhang *et al.* applied GMM weight adaptation in conjunction with mean adaptation for a large vocabulary speech recognition system to improve the word error rate [16]. Li *et al.* investigated the application of GMM weight supervectors in speaker age group recognition and showed that score-level fusion of classifiers based on GMM weights and GMM means improves recognition performance [15]. In [14] the feature level fusion of i-vectors, GMM mean supervectors, and GMM weight supervectors is applied to improve the accuracy of accent recognition.

Three main approaches have been suggested for GMM weights adaptation namely maximum likelihood re-estimation (ML) [17], non-negative matrix factorization (NMF) [16] and subspace multinomial model (SMM) [18]. The ML approach is conceptually simple and computationally inexpensive. However, the generalization of the adapted model is not guaranteed and only the observed weights are updated appropriately and the rest will be zero. This disadvantage affects the system performance especially for the case of short speech signals. The NMF expresses the adapted weights as a linear combination of a small number of latent vectors that are estimated on the training data [16]. This approach reduces the number of parameters that must be estimated from the enrollment data, and hence is more reliable in the context of short utterances. In this approach, the subspace matrix and the subspace vectors are assumed to be non-negative. This assumption makes the estimation of the subspace matrix more difficult. NMF is also very sensitive to initialization of the subspace matrix, which is often performed randomly. Inspired from the i-vector framework, Kockmann *et al.* introduced an approach for Gaussian weight supervector decomposition for prosodic speaker verification [18]. The same approach was also used to apply intersession compensation in the context of phonotactic language recognition [19]. Soufifar *et al.* applied the same approach to extract low-dimensional phonotactic features for LRE [20], [21]. Although this method is attractive, it is computationally complex, and hence very time consuming.

In this research, we try to develop a new subspace method for GMM weight adaptation based on a factor analysis similar to that of i-vector framework. In this method, namely non-negative factor analysis (NFA), the applied factor analysis is constrained such that the adapted GMM weights are non-negative and sum up to one. The proposed method is computationally simple and considerably faster than SMM. It also provides a wider bound for the adapted weights compared to that of the NMF. The obtained subspace vectors are applied to language and dialect recognition on three corpora, namely NIST 2011 LRE, QCRI Arabic DRE and RATS LRE. The GMM weight subspace vectors are fused with i-vectors effectively to form new vectors representing the utterances to improve the performance of the state-of-the-art i-vector framework for the language and dialect recognition tasks.

The rest of this paper is organized as follows. Section II presents the background, and briefly describes the applied baseline systems. In Section III, the proposed method is elaborated in detail. The evaluation results are presented and discussed in Section V. The paper ends with conclusions in Section VI.

II. Background

A. Problem Formulation

In the language/dialect recognition problem, we are given a training dataset $S^{\text{tr}} = \{(\mathcal{X}_1, y_1), \ldots, (\mathcal{X}_s, y_s), \ldots, (\mathcal{X}_S, y_S)\}$, where $\mathcal{X}_s$ denotes the s^{th} utterance of the training dataset, and y_s denotes a label vector that shows the correct language/dialect of the utterance. Each label vector contains a one in the i^{th} row if $\mathcal{X}_s$ belongs to the i^{th} class, and zeros elsewhere. The goal is to approximate a classifier function (g), such that for an unseen observation $\mathcal{X}^{\text{tst}}$, $y = g(\mathcal{X}^{\text{tst}})$ is as close as possible to the true label.

The first step for approximating function g is converting variable-duration speech signals into fixed-dimensional vectors suitable for classification algorithms. In this research, i-vectors, the GMM weight supervectors obtained by the ML method, the NMF subspace vectors, the SMM subspace vectors, and the NFA subspace vectors are applied for this purpose, which are described in the following sections.

B. Universal Background Model

Consider a Universal Background Model (UBM) with the following likelihood function of data $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_\tau\}$.

$$p(\mathbf{x}_t|\lambda) = \sum_{c=1}^{C} b_c p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c)$$
$$\lambda = \{b_c, \mu_c, \mathbf{\Sigma}_c\}, c = 1, \ldots C, \quad (1)$$

where $\mathbf{x}_t$ is the acoustic vector at time t, b_c is the mixture weight for the c^{th} mixture component, $p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c)$ is a Gaussian probability density function with mean μ_c and covariance matrix $\mathbf{\Sigma}_c$, C is the total number of Gaussians in the mixture. The parameters of the UBM–λ–are estimated on a large amount of training data representing different classes (languages/dialects).

C. i-vector Framework

One effective acoustic method for language/dialect recognition involves adapting UBM Gaussian means to the speech characteristics of the utterances. Then the Gaussian means of each adapted GMM are extracted and concatenated to form a supervector. Finally, the obtained Gaussian mean supervectors, which characterize the corresponding utterance, are applied to identify the language/dialect [2]. This method has been shown to provide a good level of performance in language/dialect recognition [2]. Recent progress in this field, however, has found an alternate method of modeling GMM mean supervectors that provides superior recognition performance [12]. This technique assumes the GMM mean supervector, $\mathbf{M}$, can be decomposed as

$$\mathbf{M} = \mathbf{u} + \mathbf{T}\mathbf{v}, \quad (2)$$

where $\mathbf{u}$ is the mean supervector of the UBM, $\mathbf{T}$ spans a low-dimensional subspace and $\mathbf{v}$ are the factors that best describe the utterance-dependent mean offset $\mathbf{T}\mathbf{v}$. The vector $\mathbf{v}$ is treated as a latent variable with the standard normal prior and the i-vector

is its maximum-a-posteriori (MAP) point estimate. The subspace matrix $\mathbf{T}$ is estimated via maximum likelihood in a large training dataset. An efficient procedure for training $\mathbf{T}$ and for MAP adaptation of i-vectors can be found in [22]. In this approach, i-vectors are the low-dimensional representation of an audio recording that can be used for classification and estimation purposes.

D. Conventional GMM Weight Adaptation Approaches

In this section, three main approaches of Gaussian weights adaptation are briefly described. In this paper, the UBM weight and the adapted weight of the c^{th} Gaussian are denoted by b_c and w_c respectively.

1) Maximum Likelihood Re-estimation: In this method, the adapted weights w_c are obtained by maximizing the log-likelihood function of Eq. (1) over the Gaussian weights. Rather than directly maximizing the log-likelihood function, we can also maximize the following auxiliary function over w_c

$$\Phi(\lambda, w_c) = \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t} \log w_c p(x_t|\mu_c, \mathbf{\Sigma}_c). \tag{3}$$

where $\gamma_{c,t}$ is the occupation count for the c^{th} mixture component and the t^{th} segment, and τ is the total number of frames in the utterance. Occupation counts are calculated as follows:

$$\gamma_{c,t} = \frac{b_c p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c)}{\sum_{c=1}^{C} b_c p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c)} \tag{4}$$

Maximizing Eq. (3), will maximize the data likelihood [23].

Since $p(x_t|\mu_c, \mathbf{\Sigma}_c)$ remain unchanged in this maximization process, the auxiliary function Eq. (3) can be simplified to

$$\Phi(\lambda, w_c) = \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t} \log w_c, \tag{5}$$

Finally, the adapted weights w_c after the first Expectation Maximization (EM) iteration are obtained as follows:

$$w_c = \frac{1}{\tau} \sum_{t=1}^{\tau} \gamma_{c,t} \tag{6}$$

Although maximum likelihood results are not yet reached after the first EM iteration, we will refer to this approach as ML re-estimation. In this paper, neither in the ML re-estimation scheme nor in the weight adaptation methods given bellow, iterative re-insertion of the obtained adapted weights into $\gamma_{c,t}$ is used, i.e. the occupation counts $\gamma_{c,t}$ are obtained from the UBM and are kept fixed during the adaptation process.

2) Non-negative Matrix Factorization: The main assumption of the NMF based method [16] is that for a given utterance,

$$w_c = \mathbf{B}_c \mathbf{h}, \tag{7}$$

where $\mathbf{B}_c$ is a non-negative row vector forming the cth row of the non-negative subspace matrix $\mathbf{B}$, and $\mathbf{h}$ is a low-dimensional and non-negative vector representing the utterance. In this method, $\mathbf{B}_c$ and $\mathbf{h}$ are initialized randomly, and then updated using the multiplicative updating rules [24] to maximize the objective function Eq. (5). The adapted GMM weights are constrained to be non-negative and sum up to one. Since all elements of subspace matrix $\mathbf{B}$, and subspace vector $\mathbf{h}$ are non-negative, the adapted weights using NMF are also non-negative. To keep the sum of adapted GMM weights equal to one, the columns of subspace matrix $\mathbf{B}$ are normalized to sum up to one after updating it in each iteration. This normalization is also performed for the subspace vector $\mathbf{h}$. Details of this parameter re-estimation method can be found in [16].

The subspace matrix $\mathbf{B}$ is estimated over a large training dataset. It is then used to extract a subspace vector $\mathbf{h}$ for each utterance in train and test datasets. The obtained subspace vectors representing the utterances in train and test datasets can be used to classify languages/dialects.

3) Subspace Multinomial Model: Kockmann *et al.* introduced the SMM approach for Gaussian weight adaptation and decomposition with application to prosodic speaker verification [18]. The main assumption of this method is that for a given utterance,

$$w_c = \frac{\exp(z_c + \mathbf{A}_c \mathbf{q})}{\sum_{j=1}^{C} \exp(z_j + \mathbf{A}_j \mathbf{q})}, \tag{8}$$

where z_c is the c^{th} element of the origin of the supervector subspace, $\mathbf{A}_c$ is the c^{th} row of the subspace matrix and $\mathbf{q}$ is a low-dimensional vector representing the utterance.

In this method, $\mathbf{A}_c$ and $\mathbf{q}$ are estimated using a two-stage iterative algorithm similar to EM to maximize the objective function (5). For each stage of the EM-like algorithm, an iterative optimization approach similar to that of Newton-Raphson scheme is applied. Details of this parameter re-estimation approach, which involves calculation of Hessian matrix and estimating the subspace vectors one-by-one, can be found in [18].

The subspace matrix $\mathbf{A}$ is estimated over a large training dataset. It is then used to extract a subspace vector $\mathbf{q}$ for each utterance in train and test datasets. The obtained subspace vectors representing the utterances in train and test datasets are used to classify languages/dialects.

III. Non-negative Factor Analysis

In this section, a new subspace method, namely Non-negative Factor Analysis (NFA), is introduced for GMM weight adaptation. The basic assumption of this method is that for a given utterance, the c^{th} Gaussian weight of the adapted GMM (w_c) can be decomposed as follows

$$w_c = b_c + \mathbf{L}_c \mathbf{r}, \tag{9}$$

where b_c is the c^{th} weight of the UBM. $\mathbf{L}_c$ denotes the cth row of the matrix $\mathbf{L}$, which is a matrix of dimension $C \times \rho$ spanning a low-dimensional subspace ($\rho \ll C$); $\mathbf{r}$ is a ρ-dimensional vector that best describes the utterance-dependent weight offset $\mathbf{Lr}$.

In this framework, neither subspace matrix $\mathbf{L}$ nor subspace vector $\mathbf{r}$ are constrained to be non-negative. However, unlike the i-vector framework, the applied factor analysis for estimating the subspace matrix $\mathbf{L}$ and the subspace vector $\mathbf{r}$ is constrained such that the adapted GMM weights are non-negative and sum up to one. The procedure of calculating $\mathbf{L}$ and $\mathbf{r}$ involves a two-stage algorithm similar to EM to maximize the objective function (5). In the first stage, $\mathbf{L}$ is assumed to be known, and

we try to update $\mathbf{r}$. Similarly in the second stage, $\mathbf{r}$ is assumed to be known and we try to update $\mathbf{L}$. Each step is elaborated in the next subsections.

The subspace matrix $\mathbf{L}$ is estimated over a large training dataset. It is then used to extract a subspace vector $\mathbf{r}$ for each utterance in train and test datasets. The obtained subspace vectors representing the utterances in train and test datasets are used to classify languages and dialects in this paper.

A. Updating Subspace Vector $\mathbf{r}$

In the first stage of the applied iterative optimization procedure, vector $\mathbf{r}$ is estimated as follows:

1) Constrained optimization problem: Substituting w_c by $b_c + \mathbf{L}_c\mathbf{r}$ in the objective function of Eq. (5), we obtain

$$\Phi(\lambda, \mathbf{r}) = \sum_{t=1}^{\tau}\sum_{c=1}^{C} \gamma_{c,t} \log\left(b_c + \mathbf{L}_c\mathbf{r}\right) \tag{10}$$

or

$$\Phi(\lambda, \mathbf{r}) = \bar{\gamma}'(\mathcal{X}) \log\left(\mathbf{b} + \mathbf{L}\mathbf{r}\right), \tag{11}$$

where the $\log$ operates element-wise and $'$ denotes transpose. $\mathbf{b}$ and $\bar{\gamma}(\mathcal{X})$ are obtained as follows,

$$\bar{\gamma}(\mathcal{X}) = \sum_{t} \left[\, \gamma_{1,t} \quad \dots \quad \gamma_{C,t} \,\right]' \tag{12}$$

$$\mathbf{b} = \left[\, b_1 \quad \dots \quad b_C \,\right]' \tag{13}$$

Given an utterance $\mathcal{X}$, a maximum likelihood estimation of $\mathbf{r}$ can be found by solving the following constrained optimization problem:

$$\begin{aligned} &\max_{\mathbf{r}} \Phi(\lambda, \mathbf{r}) \\ &\text{Subject to} \\ &\mathbf{1}(\mathbf{b} + \mathbf{L}\mathbf{r}) = 1 \quad \text{Equality constraint} \\ &\mathbf{b} + \mathbf{L}\mathbf{r} > 0 \quad \text{Inequality constraint,} \end{aligned} \tag{14}$$

where $\mathbf{1}$ is a row vector of dimension C with all elements equal to 1. This constrained optimization problem has the following analytical solution for a square full-rank $\mathbf{L}$ (the proof for this relation is given in Appendix A):

$$\mathbf{r}(\mathcal{X}) = \mathbf{L}^{-1}\left[\frac{1}{\tau}\bar{\gamma}(\mathcal{X}) - \mathbf{b}\right] \tag{15}$$

For a skinny $\mathbf{L}$, where the number of rows is greater than the number of columns, solving this constrained optimization problem involves using iterative optimization approaches. Solving a constrained optimization problem is usually more time-consuming compared to an unconstrained one. Therefore, we relax the constraints, and convert the problem to an unconstrained optimization by the following simple tricks.

2) Reformulation of the equality constraint: The equality constraint is

$$\mathbf{1}\mathbf{b} + \mathbf{1}\mathbf{L}\mathbf{r} = 1. \tag{16}$$

We know that the UBM weights sum up to one, or $\mathbf{1}b = 1$. Hence

$$\mathbf{1}\mathbf{L}\mathbf{r} = 0. \tag{17}$$

If $\mathbf{1}$ is orthogonal to all columns of $\mathbf{L}$, i.e., $\mathbf{1}\mathbf{L} = 0$, the constraint of Eq. (17) holds for any possible $\mathbf{r}$. In the second stage of optimization, $\mathbf{L}$ is calculated such that $\mathbf{1}\mathbf{L} = 0$ holds.

3) Relaxing the inequality constraint: As can be seen in Eq. (14) there are C inequality constraints. If any inequality constraints are violated, the cost function of Eq. (14) cannot be evaluated. In numerical optimization, if we start from a feasible point, there will be a wall over which we cannot climb, as the cost function becomes infinite at the boundary. Therefore, by controlling the steps of the maximization approach, violating the inequality constraint can be easily avoided. The exception is when any component of $\bar{\gamma}'(\mathcal{X})$ is zero. To avoid this problem, we replace zero elements of $\bar{\gamma}'(\mathcal{X})$ by very small positive values.

4) Maximization using gradient ascent: By simplifying the problem to an unconstrained maximization, different optimization techniques can be applied to obtain the maximum likelihood estimate of $\mathbf{r}$ in a reasonable time. We use a simple gradient ascent method with the following updating formula,

$$\mathbf{r}_i = \mathbf{r}_{i-1} + \alpha_E \nabla\Phi(\lambda, \mathbf{r}_{i-1}) \tag{18}$$

$$\nabla\Phi(\lambda, \mathbf{r}) = \mathbf{L}'\frac{[\bar{\gamma}'(\mathcal{X})]}{[\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X})]}, \tag{19}$$

where $\frac{[.]}{[.]}$ denotes the element-wise division, subscript i is the index for gradient ascent iteration, α_E is the learning rate and ∇ denotes gradient operator. In the first step of this method, α_E is set to a non-critical (non-negative) value and then it is reduced at each unsuccessful step (e.g. halved) and increased in each successful step (multiplied by 1.5). An unsuccessful iteration is when $\Phi(\lambda, \mathbf{r})$ decreases or any of the inequality constraints are violated. On our data, six successful gradient ascent iterations were enough for convergence of subspace vectors $\mathbf{r}$.

5) Initialization: Like many optimization problems, a bad initialization leads to a bad result. In this section, we try to obtain a reasonable initial point to be used in the iterative optimization algorithm. As mentioned, the constrained optimization problem has an analytical solution in the case of a square full-rank $\mathbf{L}$ given in Eq. (15). After reformulation explained in Section III-A2, $\mathbf{L}$ is never of full-rank. However, for a skinny $\mathbf{L}$, we can use the Moore-Penrose pseudo-inverse instead of the inverse to obtain a vector of the same dimension as $\mathbf{r}$.

$$\mathbf{r}_{pinv} = \mathbf{L}^{\dagger}\left[\frac{1}{\tau}\bar{\gamma}(\mathcal{X}) - \mathbf{b}\right] \tag{20}$$

where $\dagger$ is the sign for Moore-Penrose pseudo-inverse; $\mathbf{r}_{pinv}$ is an optimal solution for minimizing the Euclidean distance between $\frac{1}{\tau}\bar{\gamma}$ and $\mathbf{b} + \mathbf{L}\mathbf{r}$. However, this solution ($\mathbf{r}_{pinv}$) may violate the inequality constraints of the problem, and hence be unfeasible. Since $w_c = b_c + \mathbf{L}_c\mathbf{r}$ and b_c are non-negative, a $\mathbf{r}$ with sufficiently small elements satisfies the inequality constraints. Therefore, by multiplying a small value θ to $\mathbf{r}_{pinv}$, we obtain a feasible initial point as follows:

$$\mathbf{r}_0 = \theta\mathbf{r}_{pinv} \tag{21}$$

We start from $\theta = 1$ and reduce (half) it until reaching a feasible initial point. On our data, $\theta = 0.1$ has been found small enough to obtain a feasible initial point.

B. Updating Subspace Matrix $\mathbf{L}$

In the M-step, assuming $\mathbf{r}$ is known for all utterances in the training database, matrix $\mathbf{L}$ can be obtained by solving the following constrained optimization problem.

$$\begin{aligned} &\max_{\mathbf{L}} \widetilde{\Phi}(\lambda, \mathbf{L}) \\ &\text{Subject to} \\ &\mathbf{1}(\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X}_s)) = 1 \quad \text{Equality constraint} \\ &\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X}_s) > 0 \quad \text{Inequality constraint} \\ &s = 1, \ldots S, \end{aligned} \tag{22}$$

where

$$\widetilde{\Phi}(\lambda, \mathbf{L}) = \sum_s \bar{\gamma}'(\mathcal{X}_s) \log\left[\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X}_s)\right] \tag{23}$$

This constrained optimization problem has no analytical solution. Therefore, iterative optimization approaches are required.

As mentioned in Section III-A3, violating the inequality constraints can be avoided easily in numerical optimization by starting from a feasible initial point and controlling the step size.

All equality constraints can be simplified to a single constraint $\mathbf{1L} = 0$ using the same trick mentioned in Section III-A2. To solve the resulting optimization problem with equality constraint $\mathbf{1L} = 0$, projected gradient algorithm [25] is applied.

$$\mathbf{L}_i = \mathbf{L}_{i-1} + \alpha_M \mathcal{P} \nabla \widetilde{\Phi}(\lambda, \mathbf{L_{i-1}}) \tag{24}$$

$$\nabla \widetilde{\Phi}(\lambda, \mathbf{L}) = \sum_s \frac{[\bar{\gamma}(\mathcal{X}_s)]}{[\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X}_s)]} \mathbf{r}'(\mathcal{X}_s) \tag{25}$$

$$\mathcal{P} = \mathbf{I} - \frac{1}{C}\mathbf{1}'\mathbf{1}, \tag{26}$$

where subscript i is the index for gradient ascent iterations, α_M is the learning rate, $\mathbf{I}$ is an identity matrix of size C, and $\mathcal{P}$ is a projection also called the centering matrix. In the first step of this algorithm, α_M is set to a non-critical (non-negative) value and then it is reduced at each unsuccessful step (halved) and increased in each successful step (multiplied by 1.5). An unsuccessful iteration is when $\widetilde{\Phi}(\lambda, \mathbf{L})$ decreases, or any of the inequality constraints are violated. On our data, six successful gradient ascent iterations were enough for convergence of subspace matrix $\mathbf{L}$.

1) Initialization: We use Principal Component Analysis (PCA) for initialization of $\mathbf{L}$. In other words, we first form matrix $\mathbf{N}$ from the ML estimations of GMM weights for all training utterances as follows:

$$\mathbf{N} = \left[\frac{\bar{\gamma}(\mathcal{X}_1)}{\tau(1)}, \ldots, \frac{\bar{\gamma}(\mathcal{X}_s)}{\tau(s)}, \ldots, \frac{\bar{\gamma}(\mathcal{X}_S)}{\tau(S)}\right] \tag{27}$$

Then, the first ρ principal components of $\mathbf{N}$ with high eigenvalues are used as initial point of $\mathbf{L}$ for maximization of $\widetilde{\Phi}(\lambda, \mathbf{L})$.

IV. Comparison Between NMF, SMM and NFA

In this section, flexibility and computational cost of NMF, SMM, and NFA are compared.

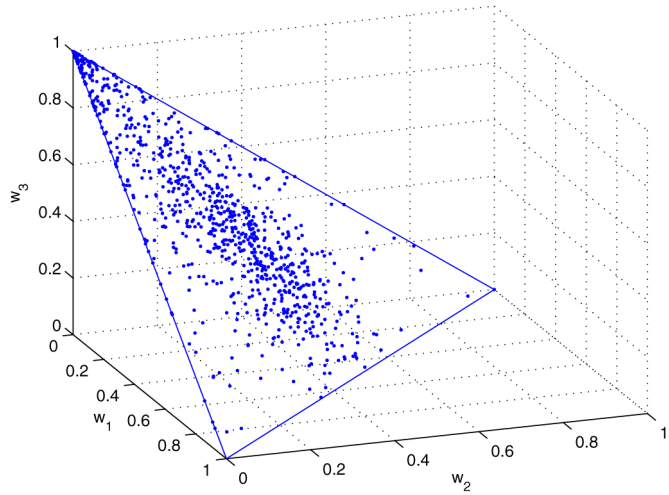


Fig. 1. The adapted weights of the UBM with three Gaussians using the ML method.

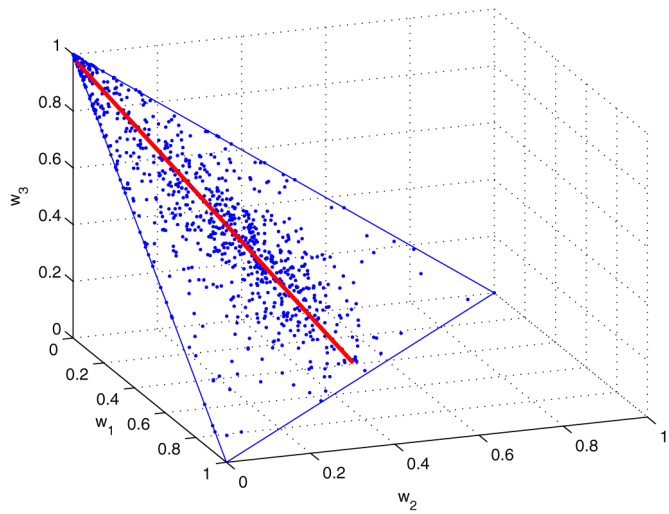


Fig. 2. The space of possible adapted weights of a UBM with three Gaussians using NMF.

A. Modeling

Fig. 1 shows the adapted weights of the UBM with three Gaussians using the ML re-estimation approach described in Section II-D1. In this figure, each dot shows the adapted weights using the ML approach for an utterance. Since the GMM weights are constrained to be positive, and sum up to 1, they are embedded in a simplex. As shown in this figure, the adapted weights using the ML method can be very small—zero or very near zero—because the adapted weights of unobserved Gaussians or weakly observed Gaussians are zero or very near zero respectively. Consider the utterances and the UBM of Fig. 1. Given these utterances as the training dataset, NMF, SMM and NFA are used to estimate a subspace matrices $\mathbf{B}$, $\mathbf{T}$ and $\mathbf{L}$ respectively.

For NMF, the straight line in Fig. 2 shows the set of any possible adapted weights obtained using the estimated subspace matrix $\mathbf{B}$, which is of dimension 3×2 and was estimated after 300 iterations of the multiplicative updating algorithm [24] starting from a random initialization. Since $\mathbf{h}$ is non-negative and is normalized such that its elements sum up to one, the

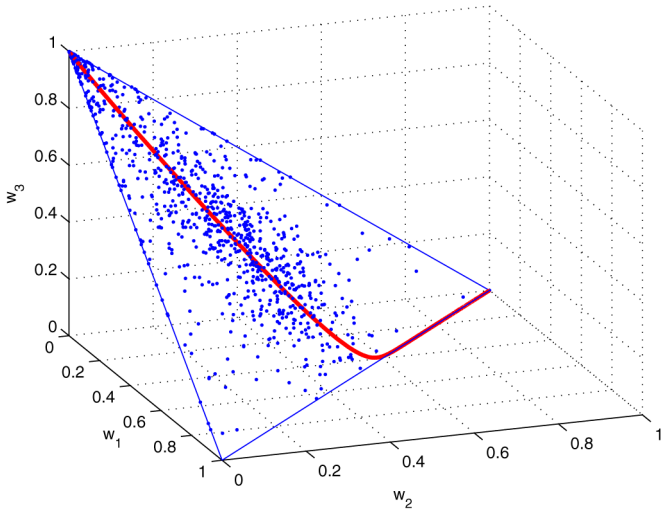


Fig. 3. The space of possible adapted weights of a UBM with three Gaussians using SMM.

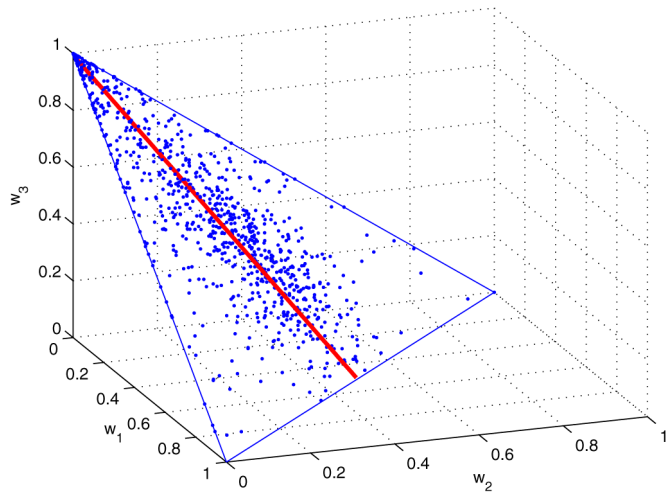


Fig. 4. The space of possible adapted weights of a UBM with three Gaussians using NFA.

adapted weights using Eq. (7) make a convex combination of the columns of $\mathbf{B}$. Hence, the adapted weights are constrained to a bounded straight line on the simplex, as shown in Fig. 2. As can be seen in this figure, although there are some data points near the border of the simplex, the straight line does not hit the border of the simplex. This shows that the subspace matrix $\mathbf{B}$ was not estimated appropriately. A closer analysis shows that this effect can be attributed to both slow convergence and falling into local minima. Depending on the initial value of $\mathbf{B}$, NMF may converge to an appropriate subspace matrix and the straight line can hit the border of the simplex. The multiplicative updating algorithm [24] does not guarantee convergence to the global minimum and is very sensitive to initialization, which is performed randomly in this example. In the GMM weight adaptation problem, where the dimension of input data and the number of training datapoints are considerably greater than those of this example, this problem is expected to be even more challenging.

For the SMM, the curved line in Fig. 3 shows the set of any possible adapted weights obtained using the estimated subspace matrix $\mathbf{A}$, which is of dimension 3×1. Since $\mathbf{q}$ is of dimension 1, and is not bounded, the adapted weights using Eq. (8) are embedded in a curved line hitting the corners of the simplex as shown in Fig. 3. Since this curved line necessarily hits two corners of the simplex, the adapted weights can take on very small values for unobserved, or weakly observed, Gaussians in two dimensions as for the ML results. This problem is addressed in [26] by adding a regularization term. However, the regularization parameter requires fine-tuning over a development dataset [26].

For NFA, the straight line in Fig. 4 shows the set of possible adapted weights obtained using the estimated subspace matrix $\mathbf{L}$, which is of dimension 3×1. Since $\mathbf{r}$ is of dimension 1, and is not constrained to be non-negative, the adapted weights using Eq. (9) are embedded in a straight line hitting the boundaries of the simplex as shown in Fig. 4. This straight line does not necessarily hit the corners of the simplex[1]. This natural constraint makes it less flexible compared to SMM, where the adapted weights can take very small values due the constraint that some simplex corner points are necessarily included in the obtained subspace. In contrast, both NMF and NFA avoid this problem because obtained subspaces of these approaches do not necessarily include simplex corners. The main difficulties of obtaining an appropriate subspace matrix in NMF are slow convergence rate, local optima and initialization, which will be further discussed in the next section.

B. Computation and Initialization

The procedure of updating the subspace matrix, and the subspace vectors is different between NMF, SMM and NFA frameworks.

In the applied NMF, the subspace matrix and subspace vectors are randomly initialized, and then multiplicative updating rules are applied to update the subspace matrix and subspace vectors. On our data, convergence was obtained in around 300 iterations.

In SMM, the initialization of the subspace matrix is similar to that of NFA, and the initial value of the subspace vectors is considered to be zero. SMM applies an optimization technique similar to that of Newton-Raphson, where computational complexity of construction and inversion of the approximated Hessian matrix grows cubically with the subspace dimension. In this procedure, the subspace vectors are estimated one-by-one, which does not allow compilers to optimally exploit the parallelism of modern computer architectures, while matrix formulations as in NMF and NFA, do. On our data, convergence of SMM subspace matrix re-estimation was obtained in 10 iterations.

In NFA, the subspace matrix and subspace vectors are initialized as described in Sections III-B1 and III-A5, respectively. NFA applies a simple gradient ascent technique to estimate a subspace matrix and subspace vectors. Like in NMF, in this

[1]"It nearly hits one corner of the simplex due to specific distribution of the given data in this example. However, this straight line generally starts from a boundary of the simplex and ends at another boundary of it depending on the distribution of the data.

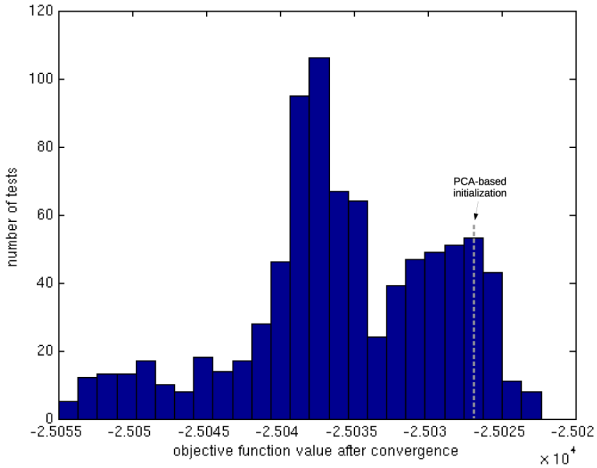


Fig. 5. The histogram of objective function value after convergence for 100 randomly initialized NFA factorizations.

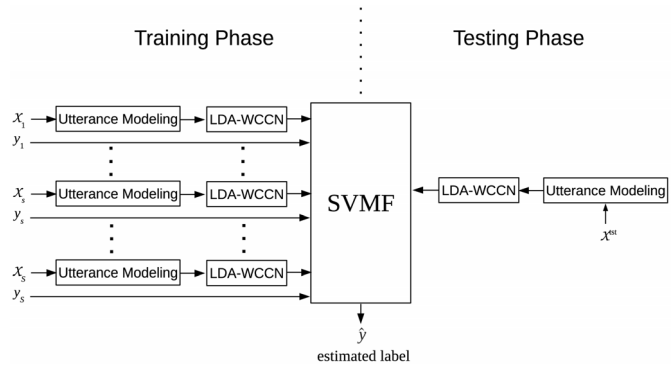


Fig. 6. The block-diagram of applied classification scheme NIST 2011 LRE and QCRI Arabic DRE experiments.

technique, the corresponding subspace vector for all utterances are treated as a single matrix, and then the gradient ascent technique is applied over the matrix. This makes the optimization significantly faster compared to estimating subspace vector for each utterance one-by-one. In this approach, convergence can be obtained in around 10 iterations of the applied two-stage optimization procedure.

Two stage optimization approaches in NMF, SMM and NFA do not guarantee the convergence to the global minimum, and hence the initialization of the subspace matrices and the subspace vectors are critical. An important advantage of SMM and NFA compared to NMF is that the subspace matrices of these methods are not constrained to be non-negative and PCA is used for their initialization as described in Section III-B1, while the initialization of the subspace matrix in NMF is more challenging as it is constrained to be non-negative.

To investigate the effect of the applied initialization in NFA, the toy problem of Section IV-A is considered. Fig. 5 shows the histogram of objective function value of the converged terials for over 850 randomly initialized NFA factorizations (subspace matrix initialization by random non-negative values is often used in NMF). The objective function value after convergence using the suggested initialization, which is shown by a dashed-line in the figure, is greater than that of NFA with random initialization in most of trials. Therefore, the suggested methods in Sections III-B1 and III-A5 yield a reasonable initial subspace matrix and subspace vectors to be used in the iterative optimization algorithm.

V. Experiments and Results

In this section, the performance of the proposed method and its characteristics are investigated on the NIST 2011 LRE, QCRI Arabic DRE and RATS LRE corpora.

A. NIST 2011 LRE

1) Database: The National Institute of Science and Technology (NIST) 2011 LRE corpus is composed of 24 languages—Bengali, Dari, English-American, English-Indian, Farsi/Persian, Hindi, Mandarin, Pashto, Russian, Spanish, Tamil, Thai, Turkish, Ukrainian, Urdu, Arabic-Iraqi, Arabic-Levantine, Arabic-Maghrebi, Arabic-MSA, Czech, Lao, Punjabi, Polish, and Slovak—collected over telephone conversations and narrowband recordings. This evaluation set composed by three conditions based on the duration of the test segments. These durations are 30 s, 10 s and 3 s.

The applied data for training and tuning are similar to that of the MIT Lincoln Laboratory (MITLL) system [27] submitted to the NIST 2011 LRE and were collected from the following sources:

- Telephone data from previous NIST (1996, 2003, 2005, 2007, 2009) LRE datasets, CallFriend, CallHome, Mixer, OHSU, and OGI-22 collections.
- Narrowband recordings collected from VOA broadcasts, Radio Free Asia, Radio Free Europe, and GALE broadcasts.
- Arabic corpora from LDC and Appen data were also obtained from telephone conversations, and some interview data
- Some extra data were also obtained from Special Broadcast Services (SBS) in Australia.
- NIST 2011 LRE development data also included telephone conversations and narrowband broadcast segments.

2) UBM and Features: In this experiment, the applied UBM has 2048 mixtures, and acoustic features are exactly the same as that of the MIT Lincoln Laboratory (MITLL) NIST 2011 LRE submission [27]. They are based on cepstral features extracted using a sliding window of 20 ms length, and 10 ms overlap. These features were subjected to vocal tract length normalization followed by RASTA filtering [28]. The obtained cepstral features were converted to a Shifted Delta Cepstral (SDC) representation based on the 7-1-3-7 configuration. This configuration produces a sequence of vectors of dimension 56. After extracting the SDC features and removing the non-speech frames, the feature vectors are mean and variance normalized over each speech recording. An intersession compensation technique, named feature Nuisance Attribute Projection (fNAP), is then applied on the features domain, similar to the approach proposed in [29].

3) Classification and Calibration: The block-diagram of the applied classification scheme is shown in Fig. 6. As can be interpreted from this figure, in the training phase, each utterance in the train dataset is converted to a vector using one of the utterance modeling approaches (ML, SMM, NMF, NFA,

or i-vector) described in Sections II-D, II-C and III. Then, the obtained vectors representing the utterances are length normalized–such that their second norm equal to unity–and transformed using linear discriminant analysis (LDA), such that the ratio of the transformed between-class-scatter and the transformed within-class-scatter is maximized [30]. The number of discriminant dimensions in the applied LDA equals the number of categories minus one. The low-dimensional vectors are then transformed using within-class covariance normalization (WCCN) to transform the within-class covariance of the vector space to an identity matrix [31]. In doing so, directions of relatively high within-class variation will be attenuated, and thus prevented from dominating the space [31]. The projection matrices of LDA and WCCN are trained using the training data from all languages. Then, the obtained transformed vectors along with their corresponding language/dialect labels are used to train a scoring approach working based on simplified Von-Mises-Fisher distribution [27]. This scoring approach, labeled as SVMF in this paper, is described in [27].

In the testing phase, the utterance modeling approach applied in the training phase is used to extract a vector from the utterance of an unseen speaker. Then the projection matrices of LDA and WCCN calculated in the training phase are applied to transform the obtained vector representing the test utterance to a low-dimensional space. Finally the trained SVMF uses the transformed vector to recognize the language/dialect of the test speaker. The SVMF score of the transformed test vector ν_{test} for the l^{th} language is obtained as follows

$$score_l = \nu'_{test}\bar{\nu}_l, \tag{28}$$

where $\bar{\nu}_l$ denotes the mean of the transformed vectors for the lth language in the training dataset.

To obtain well-calibrated scores on the evaluation dataset, linear logistic regression calibration [32], [33] is applied in the back-end. In this research, the FoCal Multiclass Toolkit [32] is applied to perform this calibration.

4) Performance Measure: In this experiment, the effectiveness of the proposed method is evaluated using log-likelihood-ratio cost (C_{llr}) [33], [34], which is also referred to as multi-class-cross-entropy in literatures [35]. C_{llr} is an application-independent performance measure for recognizers with soft decision output in the form of log-likelihood-ratios. This performance measure, which has been adopted for use in the NIST speaker recognition evaluation, was initially developed for binary classification problems such as speaker recognition. It was extended to multi-class classification problems such as language recognition [33]. In this research, we apply the FoCal Multiclass Toolkit [32] to calculate C_{llr}.

5) Comparison with Baseline Systems: Fig. 7 shows the C_{llr} of language recognition for all utterances in testing dataset (regardless of utterance duration) using the proposed method and baseline systems versus the subspace vector dimension. This figure shows that the proposed method and the SMM increase the performance of language recognition compared to the ML weight supervector. It is also shown that the best results of the proposed method and the SMM are obtained at target dimension 800 and 200 respectively and the performance of the proposed

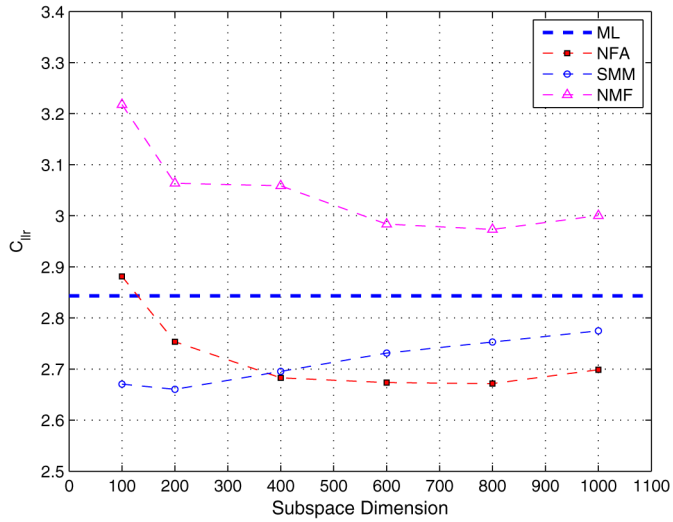


Fig. 7. The C_{llr} of language recognition using the proposed method and baseline systems versus subspace vector dimension.

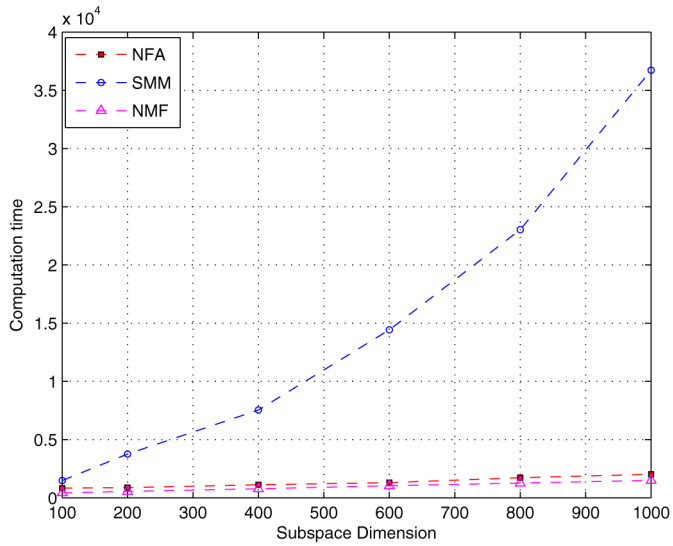


Fig. 8. The required computation time for estimating the subspace matrices using the proposed method and baseline systems versus subspace vector dimension.

method is robust against subspace dimension changes between dimensions 500 and 800.

For comparison purposes, all experiments on NIST 2011 LRE are performed using a computer with CPU model of Intel Xeon E5-1620 0 at 3.60 GHz and 16 GB of RAM. Fig. 8 shows the required computation time (elapsed time) for estimating the subspace matrices using the proposed method and baseline systems versus subspace vector dimension. This figure shows that the required computation time for estimating the subspace matrices using the SMM is significantly higher than that of NFA and NMF especially for higher subspace dimensions. The required time for NFA and NMF grows linearly by increasing the subspace vector dimension, while this growth is cubic in the case of SMM.

Fig. 9 shows the language recognition performance using the proposed method and baseline systems in different utterance length conditions. This bar chart demonstrates the results of NMF, SMM and NFA in their best subspace dimension. This

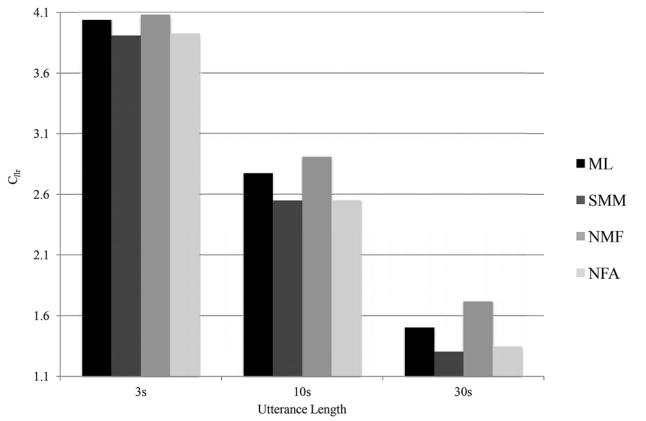


Fig. 9. The C_{llr} of language recognition using the proposed method and baseline systems in different utterance length conditions.

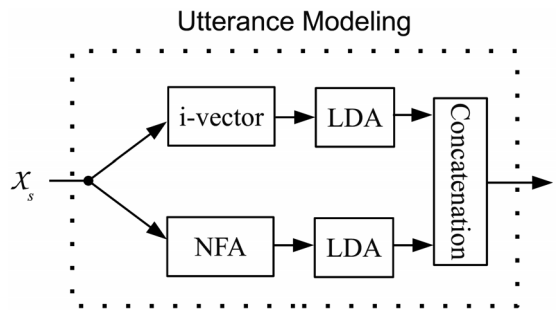


Fig. 10. The block-diagram of utterance modeling in intermediate-level fusion.

figure shows that the proposed method and SMM improve the ML estimations at 3 s, 10 s, and 30 s utterance length conditions. The obtained relative improvements [36] by the NFA compared to the ML baseline system in 3 s, 10 s and 30 s conditions are 2.7%, 8.1%, and 11.6% respectively.

6) Fusion with i-vector Framework: The goal of this research is improving the recognition accuracy of the state-of-the-art i-vector system. The applied baseline i-vector system in this research is the same as the ivec 1 subsystem of the MITLL NIST 2011 LRE submission [27]. The ivec 1 subsystem achieved the highest performance in comparison to other acoustic and phonotactic subsystems of the MITLL submission. To improve this system, an intermediate-level fusion of i-vectors and NFA subspace vectors is proposed. The block-diagram of the applied classification procedure in training and testing phases is the same as Fig. 6. However, the utterance modeling blocks are replaced with the illustrated block in Fig. 10. As shown in this figure, each i-vector, which is of dimension 600, is projected to a low-dimensional (the number of categories minus one) space using LDA. The LDA transformation matrix is calculated using all i-vectors in the training dataset. The same procedure is performed on the NFA subspace vectors. Then the obtained low-dimensional vectors are concatenated to form a new vector. Then, the obtained vectors modeling the utterances are applied to identify the utterance language using the classification procedure of Fig. 6, where LDA and WCCN are applied for session variability compensation and SVMF is used as a classifier.

TABLE I
THE C_{llr} OF LANGUAGE RECOGNITION USING THE PROPOSED METHOD AND BASELINE SYSTEMS AFTER INTERMEDIATE-LEVEL FUSION WITH I-VECTORS

Method	3s	10s	30s
i-vector	3.39	1.71	0.775
i-vector-ML	3.32	1.70	0.773
i-vector-NMF	3.31	1.66	0.762
i-vector-SMM	3.30	1.62	0.725
i-vector-NFA	3.28	1.60	0.717

TABLE II
THE NUMBER OF UTTERANCES FOR EACH DIALECT CATEGORY IN THE QCRI CORPUS

Method	Training	Development	Evaluation
Egyptian	1116	463	139
Levantine	1074	186	132
Gulf	1181	221	218
MSA	1480	254	207
Total	**5051**	**1124**	**696**

TABLE III
THE NUMBER OF UTTERANCES IN DIFFERENT DURATIONS IN THE QCRI CORPUS

Duration	Training	Development	Evaluation
shorter than 5s	723	141	97
5s-10s	754	156	103
10s-20s	968	225	123
20s-30s	649	153	100
30s-60s	835	207	102
Longer than 60s	366	115	41

Table I lists the i-vector based system and obtained results after the proposed intermediate-level fusion. The intermediate-level fusion of i-vector framework with NMF, SMM and NFA are performed using the best subspace dimension of these methods. As can be seen in this table, the obtained relative improvements [36] by this fusion compared to the state-of-the-art i-vector based recognizer in 3 s, 10 s, and 30 s conditions are 3.33%, 6.23%, and 7.45% respectively.

B. QCRI Arabic DRE

1) Database: The Qatar computing research institute (QCRI) Arabic DRE corpus consists of Broadcast News, in four dialects; Egyptian, Levantine, Gulf, and Modern Standard Arabic (MSA). Data recordings were done using satellite cable sampled at 16 kHz. The Aljazeera channel is the main source for the collected data. The recordings have been segmented into a wide range of durations to avoid speaker overlap, and avoid any non-speech parts such as music and background noise. Table II lists the number of utterances in each category for training, development and evaluation datasets.

Table III lists the number of utterances in different time durations.

2) UBM and Features: In the QCRI Arabic DRE experiment, the applied UBM has 512 mixtures and the feature extraction stage is based on a Shifted Delta cepstral representation. Speech is windowed at 20 ms with a 10 ms frame shift filtered through a Mel-scale filter bank. Each vector is then converted into a 56-dimensional vector following a shifted delta cepstral parameterization using a 7-1-3-7 configuration, and concatenated with the

TABLE IV
THE $E_{\rm ic}$ OF DIALECT RECOGNITION USING THE PROPOSED METHOD AND BASELINE SYSTEMS IN QCRI ARABIC DRE EXPERIMENT (%)

Method	Development	Evaluation
ML	31.9	33.5
NMF	31.2	32.6
SMM	36.9	34.0
NFA	30.1	30.7

TABLE V
THE $E_{\rm ic}$ OF DIALECT RECOGNITION USING THE PROPOSED METHOD AND BASELINE SYSTEMS AFTER INTERMEDIATE-LEVEL FUSION WITH I-VECTORS IN QCRI ARABIC DRE EXPERIMENT (%)

Method	Development	Evaluation
i-vector	19.6	19.7
i-vector-ML	15.9	15.8
i-vector-NMF	15.5	15.0
i-vector-SMM	16.4	15.9
i-vector-NFA	16.0	15.0

static cepstral coefficients. The SDC feature vectors are mean and variance normalized over each speech recording. The applied i-vectors in this experiment have 400 dimension.

3) Performance Measure: In this experiment, the effectiveness of the proposed method is evaluated using the percentage of incorrectly classified utterances ($E_{\rm ic}$), which can be calculated using the following relation:

$$E_{\rm ic} = \frac{N_{\rm ic}}{S_{\rm tst}} \tag{29}$$

where $N_{\rm ic}$ and $S_{\rm tst}$ denote the number of incorrectly classified utterances, and the total number of utterances in the test dataset respectively.

4) Comparison: In this experiment, the same classification and calibration procedure of Section V-A3 is used, and the block-diagram of the applied classification scheme is shown in Fig. 6. However, to calculate E_{ic}, rather that soft scores, we require hard decision, which is performed by maximizing over the obtained scores for each category.

Table IV lists the $E_{\rm ic}$ of dialect recognition using the proposed method and baseline systems. In this experiment, SMM, NMF, and NFA have been tested over different target dimensions between 50 and 500, and Table IV only includes the best results, which were obtained for target dimensions 400, 200, and 400 for NMF, SMM, and NFA respectively. As can be seen in this table, the NMF, and NFA subspace approaches improve the ML results in this experiment.

We also used the same intermediate-level fusion scheme described in Section V-A6 to improve the accuracy of the i-vector based system. Table V lists the $E_{\rm ic}$ of dialect recognition using the proposed method and baseline systems after intermediate-level fusion with i-vectors. As can be seen in this table, the average of E_{ic} over development and evaluation datasets for the i-vector framework and proposed fusion scheme are 19.65% and 15.5% respectively. Comparison of these values shows that the absolute and the relative improvements [36], obtained by intermediate-level fusion of the proposed method with the i-vector system are around 4%, and 21% respectively.

TABLE VI
THE NUMBER OF UTTERANCES FOR EACH CATEGORY IN THE RATS CORPUS

Language	Training	Development	Evaluation
Dar	3305	2733	184
Arle	46760	4023	1085
Urd	22775	4019	908
Pas	29605	4007	1032
Far	9006	3999	947
Non-Target	29208	9723	2518
Total	**140659**	**28504**	**6674**

C. RATS LRE

1) Database: The Robust Automatic Transcription of Speech (RATS) P2 evaluation corpus is partially sourced from existing databases including

- Fisher Levantine conversational telephone speech (CTS)
- Callfriend Farsi CTS.
- NIST LRE Data - Dari, Farsi, Pashto, Urdu and non-target languages.

New data, namely RATS Farsi, Urdu, Pashto, Levantine CTS, were also collected and added to the database. All recordings were retransmitted through eight different communication channels. The RATS goal is to categorize test set speech recordings into six different groups including five target languages, namely Dari (Dar), Arabic Levantine (Arle), Urdu (Urd), Pashto (Pas), Farsi (Far), and one non-target category which can be from 10 unknown languages. The RATS P2 evaluation corpus is divided into three disjoint databases namely training, development and evaluation. Table VI lists the number of utterances in each category for training, development and evaluation datasets. The duration of all utterances in the training and development datasets is 120 seconds (s). Therefore, shorter duration speech signals have been created by cutting the original utterances after speech activity detection. The evaluation set speech signals has four different durations 120 s, 30 s, 10 s and 3 s.

2) UBM and Features: In this experiment, the applied UBM has 2048 mixtures, and the feature extraction stage used in this experiment is based on a Shifted Delta cepstral representation. Speech is windowed at 20 ms with a 10 ms frame shift filtered through a Mel-scale filter bank. Each vector is then converted into a 56-dimensional vector following a shifted delta cepstral parameterization using a 7-1-3-7 configuration, and concatenated with the static cepstral coefficients. Speech activity detection based on a Brno university of technology neural network implementation is then applied to remove the silence [37]. The applied i-vectors in this experiment have 600 dimension.

3) Classification: In this experiment, we applied a four-layer Deep belief nets (DBN) [38], where the first hidden layer consists of 1600 units, the second hidden layer consists of 200 units and the output layer has 6 units (the number of language categories).

4) Comparison: Table VII lists the $E_{\rm ic}$ for the proposed method and baseline systems. The results of NMF and SMM are slightly worse than that of ML in this experiment, hence excluded from the table. The large number of utterances and highly degraded channels [39], which may rise the chance of falling into local minima, can be the reason of unsatisfactory

TABLE VII
THE E_{ic} OF DIALECT RECOGNITION USING THE PROPOSED METHOD AND BASELINE SYSTEMS IN RATS LRE EXPERIMENT (%)

System Configuration	Evaluation Dataset			
	120s	30s	10s	3s
ML	14.0	32.1	49.3	61.9
NFA	11.0	25.2	42.1	58.7
i-vector	8.9	24.5	39.0	53.2
Fusion	8.1	22.5	35.5	46.6

results in SMM and NMF. As can be seen in this table, the average of E_{ic} over 120 s, 30 s, 10 s, and 3 s time conditions for the NFA and ML are 34.23% and 39.3% respectively. Therefore, the absolute improvement obtained by the proposed method compared to the baseline ML system is 5%. However, the accuracy of NFA, which works based on Gaussian weights, is lower than the i-vector based system, which works based on Gaussian means. This concurs with previous studies demonstrating that GMM weight supervectors, which entail a lower dimension compared to Gaussian mean supervectors, carry less information than GMM means [14]–[16]. However, Gaussian weights provide a source of complementary information to the Gaussian means. Therefore, to enhance the accuracy of language recognition we apply a fusion of i-vectors and NFA vectors. The last row of Table VII shows the fusion results obtained by concatenating i-vectors with NFA subspace vectors. As can be seen in this table, the average of E_{ic} over 120 s, 30 s, 10 s, and 3 s time conditions for the i-vector framework and proposed fusion scheme are 31.4% and 28.17% respectively. Comparison of these values shows that the absolute and the relative improvements [36] obtained by the proposed fusion are around 3% and 10% respectively. The improvement is more evident in the case of short utterances.

VI. CONCLUSIONS

In this paper, a new subspace method, non-negative factor analysis (NFA), for GMM weight adaptation has been introduced. The proposed approach applies a constrained factor analysis and suggests a new low-dimensional utterance representation. Evaluation on three different language/dialect recognition corpora, namely NIST 2011 LRE, RATS LRE and QCRI Arabic DRE, show that the proposed utterance representation scheme yields more accurate recognition results compared to ML re-estimation, SMM, and NMF approaches, while keeping the required computation time similar to NMF and considerably less than SMM. To improve the recognition accuracy of the state-of-the-art i-vector framework, an intermediate, or feature level fusion of i-vectors and proposed subspace vectors has been suggested. Experimental results show that the obtained relative improvements of the fusion scheme compared to i-vector frameworks are 6%, 20%, and 10% for NIST 2011 LRE, QCRI Arabic DRE, and RATS LRE.

APPENDIX A

The function to be maximized is

$$\Phi(\lambda, \mathbf{r}) = \bar{\gamma}'(\mathcal{X}) \log\left(\mathbf{b} + \mathbf{L}\mathbf{r}\right) \tag{30}$$

The equality constraint is

$$\mathbf{1}\left(\mathbf{b} + \mathbf{L}\mathbf{r}\right) = 1 \tag{31}$$

By introducing a Lagrange multiplier we reach

$$z(x) = \bar{\gamma}'(\mathcal{X}) \log\left(\mathbf{b} + \mathbf{L}\mathbf{r}\right) + \beta\left[1 - \mathbf{1}\left(\mathbf{b} + \mathbf{L}\mathbf{r}\right)\right] \tag{32}$$

By differentiating Eq. (32) with respect to $\mathbf{r}$ and setting the result to 0 we reach

$$\frac{\left[\bar{\gamma}(\mathcal{X})\right]'}{\left[\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X})\right]'}\mathbf{L} = \beta\mathbf{1}\mathbf{L} \tag{33}$$

Since $\mathbf{L}$ is a full rank matrix, we can drop it from both sides of Eq. (33).

$$\frac{\left[\bar{\gamma}(\mathcal{X})\right]'}{\left[\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X})\right]'} = \beta\mathbf{1} \tag{34}$$

hence

$$\bar{\gamma}(\mathcal{X}) = \beta\left(\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X})\right) \tag{35}$$

Considering the equality constraint mentioned in Eq. (14) and multiplying with $\mathbf{1}$ on both sides of Eq. (35)

$$\mathbf{1}\bar{\gamma}(\mathcal{X}) = \beta\mathbf{1}\left(\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X})\right) \tag{36}$$

or

$$\tau = \beta \tag{37}$$

Therefore,

$$\bar{\gamma}(\mathcal{X}) = \tau\left(\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X})\right) \tag{38}$$

from which the Eq. (15) is obtained. Therefore, Eq. (15) is the analytical solution of the constrained optimization problem defined in Eq. (14).

Note that since τ and all elements of $\bar{\gamma}(\mathcal{X})$ in Eq. (38) are non-negative, the result of Eq. (15) keeps all elements of $\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X})$ non-negative as well.

REFERENCES

[1] F. Biadsy, *Automatic dialect and accent recognition and its application to speech recognition*. New York, NY, USA: Columbia Univ., 2011.

[2] A. Hanani, *Human and computer recognition of regional accents and ethnic groups from British English speech*. Birmingham, U.K.: Univ. of Birmingham, Jul. 2012.

[3] Y. Muthusamy, E. Barnard, and R. Cole, "Reviewing automatic language identification," *IEEE Signal Process. Mag.*, vol. 11, no. 4, pp. 33–41, Oct. 1994.

[4] M. A. Zissman and K. M. Berkling, "Automatic language identification," *Speech Commun.*, vol. 35, no. 1, pp. 115–124, 2001.

[5] R. G. Leonard and G. R. Doddington, "Automatic language identification," RADC/Texas Instruments, Inc., Dallas, TX, , Tech. Rep. RADC-TR-74-2007TI-347650, 1974.

[6] A. S. House and E. P. Neuburg, "Toward automatic identification of the language of an utterance. I. preliminary methodological considerations," *J. Acoust. Soc. Amer.*, vol. 62, p. 708, 1977.

[7] A. Hanani, M. Russell, and M. Carey, "Human and computer recognition of regional accents and ethnic groups from British English speech," *Comput. Speech Lang.*, vol. 27, no. 1, pp. 59–74, 2013.

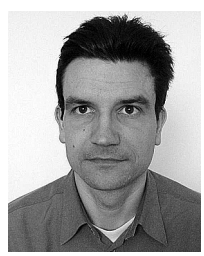
[8] M. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 1, pp. 31–44, Jan. 1996.

[9] W. M. Campbell, F. Richardson, and D. Reynolds, "Language recognition with word lattices and support vector machines," in *Proc. ICASSP*, 2007, pp. 989–992.

[10] W. Campbell, D. Sturim, and D. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 308–311, May 2006.
[11] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.
[12] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in *Proc. Interspeech*, 2011, pp. 857–860.
[13] M. H. Bahari, M. McLaren, H. Van hamme, and D. van Leeuwen, "Age estimation from telephone speech using i-vectors," in *Proc. Interspeech*, 2012, pp. 506–509.
[14] M. H. Bahari, R. Saeidi, H. Van hamme, and D. van Leeuwen, "Accent recognition using i-vector, Gaussian mean supervector and Gaussian posterior probability supervector for spontaneous telephone speech," in *Proc. ICASSP*, 2013, pp. 7344–7348.
[15] M. Li, K. J. Han, and S. Narayanan, "Automatic speaker age and gender recognition using acoustic and prosodic level information fusion," *Comput. Speech Lang.*, vol. 27, no. 1, pp. 151–167, 2013.
[16] X. Zhang, K. Demuynck, and H. Van hamme, "Rapid speaker adaptation in latent speaker space with non-negative matrix factorization," *Speech Commun.*, vol. 55, no. 9, pp. 893–908, 2013.
[17] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Process.*, vol. 10, no. 1, pp. 19–41, 2000.
[18] M. Kockmann, L. Burget, O. Glembek, L. Ferrer, and J. Cernocky, "Prosodic speaker verification using subspace multinomial models with intersession compensation," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc.*, 2010.
[19] O. Glembek, P. Matejka, L. Burget, and T. Mikolov, "Advances in phonotactic language recognition," *Interspeech'08*, pp. 743–746, 2008.
[20] M. Soufifar, M. Kockmann, L. Burget, O. Plchot, O. Glembek, and T. Svendsen, "ivector approach to phonotactic language recognition," in *Proc. Interspeech*, 2011, pp. 2913–2916.
[21] M. Soufifar, S. Cumani, L. Burget, and J. Cernocky *et al.*, "Discriminative classifiers for phonotactic language recognition with ivectors," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2012, pp. 4853–4856.
[22] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 5, pp. 980–988, Jul. 2008.
[23] A. P. Dempster, N. M. Laird, and D. B. Rubin *et al.*, "Maximum likelihood from incomplete data via the em algorithm," *J. R. Statist. Soc.-Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.
[24] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
[25] J. A. Snyman, *Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms*. New York, NY, USA: Springer Science+ Business Media, 2005, vol. 97.
[26] M. M. Soufifar, L. Burget, O. Plchot, S. Cumani, and J. Cernocky, "Regularized subspace n-gram model for phonotactic ivector extraction," in *Proc. Interspeech*, 2013, pp. 74–78.
[27] E. Singer, P. Torres-Carrasquillo, D. Reynolds, A. McCree, F. Richardson, N. Dehak, and D. Sturim, "The MITLL NIST LRE 2011 language recognition system," in *Proc. Speaker Odyssey*, 2012, pp. 209–215.
[28] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 578–589, Oct. 1994.
[29] C. Vair, D. Colibro, F. Castaldo, E. Dalmasso, and P. Laface, "Channel factors compensation in model and feature domain for speaker recognition," in *Proc. IEEE Odyssey Speaker Lang. Recogn. Workshop*, 2006, pp. 1–6.
[30] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern classification and scene analysis 2nd ed," 1995.
[31] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. Interspeech*, 2006, vol. 4, no. 2.2.
[32] N. Brummer, "Focal multi-class: Toolkit for evaluation, fusion and calibration of multi-class recognition scores," *Tutorial and User Manual. Spescom DataVoice*, 2007.
[33] N. Brummer and D. A. van Leeuwen, "On calibration of language recognition scores," in *Proc. IEEE Odyssey Speaker Lang. Recogn. Workshop*, 2006, pp. 1–8.
[34] N. Brummer, "Application-independent evaluation of speaker detection," in *Proc. ODYSSEY04- Speaker Lang. Recogn. Workshop*, 2004.
[35] L. J. Rodriguez-Fuentes, N. Brummer, M. Penagarikano, A. Varona, M. Diez, and G. Bordel, *The Albayzin 2012 language Recognition Evaluation Plan (Albayzin 2012 LRE)*, 2012.
[36] E. D. Bolker and M. Mast, *Common Sense Mathematics*. : Citeseerx, 2005.
[37] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Vesely, and P. Matejka, "Developing a speech activity detection system for the DARPA RATS program," in *Proc. Interspeech*, 2012.
[38] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
[39] K. Walker and S. Strassel, "The rats radio traffic collection system," in *Proc. Odyssey*, 2012.

Mohamad Hasan Bahari received his M.Sc. degrees in Electrical Engineering from Ferdowsi University of Mashhad, Iran, in 2010, before joining the Centre for the Processing of Speech and Images (PSI), KU Leuven, Belgium, where he was granted a Marie-Curie fellowship for a Ph.D. degree program. During winter, spring and fall 2013, he visited the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), where he proposed the non-negative factor analysis (NFA) framework. His research on automatic speaker characterization was granted the Research Foundation Flanders (FWO) for long stay abroad and awarded the International Speech Communication Association (ISCA) best student paper award at INTERSPEECH 2012. Although Mohamad Hasan's research has primarily revolved around automatic speaker characterization, his interests also extend to machine learning, and signal processing.

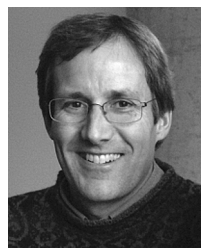
Najim Dehak received his Engineering degree in artificial intelligence in 2003 from Universite des Sciences et de la Technologie d'Oran, Algeria, and his M.S. degree in pattern recognition and artificial intelligence applications in 2004 from the Universite de Pierre et Marie Curie, Paris, France. He obtained his Ph.D. degree from Ecole de Technologie Superieure (ETS), Montreal in 2009. During his Ph.D. studies, he was also with the Centre de Recherche Informatique de Montreal (CRIM), Canada. In the summer of 2008, he participated in the Johns Hopkins University, CLSP Summer Workshop. During that time, he proposed a new system for speaker verification that uses factor analysis to extract speaker-specific features, thus paving the way for the development of the i-vector framework. Dr. Dehak is currently a research scientist in the Spoken Language Systems Group at the MIT-CSAIL and affiliate professor at ETS in Montreal. He is also a member of IEEE Speech and Language Processing Technical Committee. His research interests are in machine learning approaches applied to speech processing and speaker modeling. The current focus of his research involves extending the concept of an i-vector representation into other audio classification problems, such as speaker diarization, language and emotion-recognition.



Hugo Van hamme received the Ph.D. degree in electrical engineering from Vrije Universiteit Brussel (VUB) in 1992, the M.Sc. degree from Imperial College, London in 1988 and the Masters degree in engineering ("burgerlijk ingenieur") from VUB in 1987. Since 2002, he has been a professor at the Department of Electrical Engineering of K. U. Leuven. His main research interests are: applications of speech technology in education and speech therapy, computational models for speech recognition and language acquisition and noise robust speech recognition.

Lukas Burget (Ing. [MS]. Brno University of Technology, 1999, Ph.D. Brno University of Technology, 2004) is assistant professor at Faculty of Information Technology, University of Technology, Brno, Czech Republic. He serves as scientific director of the Speech@FIT research group. Dr. Burget supervises several Ph.D. students. From 2000 to 2002, he was a visiting researcher at OGI Portland, USA and from 2011 to 2012 he spent his sabbatical leave at SRI International, Menlo Park, USA. Lukas was invited to lead the "Robust Speaker Recognition over Varying Channels" team at the Johns Hopkins University CLSP summer workshop in 2008, and the team of BOSARIS workshop in 2010. Dr. Burget participated in several EU-sponsored projects (M4, 5th FP, AMI, 6th FP, AMIDA, 6th FP and MOBIO, 7th FP) as well as in several projects sponsored at the local Czech level. He was the principal investigator of US-Air Force EOARD sponsored project "Improving the capacity of language recognition systems to handle rare languages using radio broadcast data," was BUT's principal investigator in IARPA BEST project and works on RATS Patrol and BABEL programs sponsored by DARPA and IARPA respectively. His scientific interests are in the field of speech processing, namely acoustic modeling for speech, speaker and language recognition, including their software implementations. He has authored or coauthored more than 110 papers in journals and conferences. Lukas was the leader of teams successful in NIST LRE 2005, 2007 and NIST SRE 2006 and 2008 evaluations. He significantly contributed to the team developing AMI LVCSR systems successful in NIST RT 2005, 2006 and 2007 evaluations. He has served as reviewer for numerous speech-oriented journals and conferences. Dr. Burget is member of IEEE and ISCA.

Ahmed Ali (Ing. [MS]. Faculty of Engineering Cairo University, 1999) is senior software engineer at Qatar Computing Research Institute (QCRI). From 2000 to 2006, he was working as software engineer at IBM working on speech recognition solutions for various projects, such as Arabic viavoice, and websphere voice server. From 2006 to 2008, he was working at Marlow/UK based startup SpinVox, building multilingual speech recognition for voice mail to text application, later acquired by Nuance and moved to Cambridge from 2008 to 2011 responsible for the Acoustic Modeling at the Advanced Speech Group (ASG) in the voice mail to text team. In 2011, Ahmed joined QCRI as senior software engineer focusing on Modern Standard Arabic, and Arabic dialects for broadcast domain and lecture transcription. Ahmed is member of IEEE and serves as technical lead for various committees; such as e-Bag Project in the SEC, and Mentor for the hackathon. Ahmed has been leading the developing and deploying the Arabic BCN for Aljazeera.



James Glass is a Senior Research Scientist at the Massachusetts Institute of Technology where he heads the Spoken Language Systems Group in the Computer Science and Artificial Intelligence Laboratory. He is also a Lecturer in the Harvard-MIT Division of Health Sciences and Technology. He received his B.Eng. from Carleton University in 1982, and his S.M. and Ph.D. degrees in Electrical Engineering and Computer Science from MIT in 1985, and 1988, respectively. He has worked at the MIT Research Laboratory of Electronics, the Laboratory for Computer Science, and is currently a principal investigator in CSAIL. His primary research interests are in the area of automatic speech recognition, unsupervised speech processing, and spoken language understanding. He has lectured at MIT for over twenty years, supervised over 60 student theses, and published approximately 200 papers in these areas. He has twice served as a member of IEEE Speech and Language Technical Committee, as well as being on technical committees for several IEEE conferences and workshops, has been a Distinguished Lecturer for the International Speech Communication Association, and is an IEEE Fellow. He is currently an Associate Editor for the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, and a member of the Editorial Board for Computer, Speech, and Language.