Gaussian Mixture Model Weight Supervector Decomposition and Adaptation

Mohamad Hasan Bahari*, Najim Dehak, Hugo Van hamme

Abstract

This report proposes a novel approach for Gaussian Mixture Model (GMM) weights decomposition and adaptation. This modeling suggests a new low-dimensional utterance representation method, which uses a simple factor analysis similar to that of the i-vector framework. The suggested approach is applied to the Robust Automatic Transcription of Speech (RATS) language identification evaluation corpus, where the speech recordings are from highly degraded communication channels. In our experiments, after modeling each utterance using the proposed approach, a Deep Belief Networks (DBN) is utilized to recognize the language of utterances. The assessment results show that the proposed method improves conventional maximum likelihood weight adaptation. It is also shown that the absolute and relative improvement obtained by the score-level fusion of the i-vector framework and the proposed method are 5% and 17% respectively.

I. Introduction

Recent studies show that the GMM weights carry complimentary information to GMM means [1]–[3]. Consequently, incorporating them in the recognition system may increase the overall accuracy. Assuming the Universal Background Model (UBM) components represent the acoustic space in the training dataset,

This work was accomplished during Spring semester 2013 when M. H. Bahari was a visiting PhD student at the Spoken Language Systems (SLS) Group at the Computer Science and Artificial Intelligence Laboratory (CSAIL) of Massachusetts Institute of Technology (MIT).

This work is 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.

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

N. Dehak is with Spoken Language Systems (SLS) Group at the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL) (e-mail: najim@csail.mit.edu).

each element in the adapted GMM weights supervector of an utterance reflects the existence level of the specific components (phonemes) in the utterance. Since each language is constructed from a specific set of phonemes, GMM weights can be indicative for language identification. For example, in identification of Farsi from Arabic we can use the fact that unlike in Farsi, phoneme /p/ does not exist in Arabic. Therefore, if an utterance –regardless of its length– contains /p/, we can be sure that it is not Arabic. The main purpose of this report is to develop an approach to use this information effectively.

In the field of speaker/language recognition and age estimation, recent advances using i-vectors have increased the recognition accuracy considerably [4]–[6]. 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. Inspired from Joint Factor Analysis, Kockmann et al. introduced an approach for Gaussian weight supervector decomposition for prosodic speaker verification [7]. The same approach was also used to apply a channel compensation in the context of phonotactic Language recognition [8]. Soufifar et al. applied the same approach to extract low-dimensional phonotactic features for LRE and they named it "i-vector for phonotactic Language recognition" [9], [10]. Although this method seems to be similar to the i-vector framework –as it is named in [9]– is has some important differences. In the standard i-vector framework it is assumed that adapted means are the results of adding the UBM mean and an offset vector. The additive relation of UBM and the offset is changed to a multiplicative relation in this approach. Furthermore, no prior distribution is considered for the target low dimensional vector. To overcome these problems, a new method for GMM weight decomposition is suggested in this report.

II. BACKGROUND

A. Universal Background Model

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

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

$$\lambda = \{b_c, \mu_c, \mathbf{\Sigma}_c\}, \ c = 1, \cdots C \tag{1}$$

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 $\mathbf{\Sigma}_c$ and C is the total number of Gaussians in the mixture (in this work C is 2048).

B. Multinomial Subspace Model [7]

Consider the following log-likelihood of data for a multinomial model with C discrete classes:

$$\log p(\mathcal{X}) = \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t} \log \Phi_c \tag{2}$$

where $\gamma_{c,t}$ is the occupation count for class c and segment t. T denotes the total number of observations and Φ_c are the probabilities of multinomial distribution defined as follows:

$$\Phi_c = \frac{\exp(v_c + L_c r)}{\sum_{j=1}^{C} \exp(v_j + L_j r)} \tag{3}$$

where v_c is the c^{th} element of the origin of the supervector subspace, L_c is the c^{th} row of the subspace matrix and r is a low dimensional vector representing speaker and channel.

In this method, in each iteration of the Expectation Maximization (EM) algorithm, a Newton-Raphson approach is used to update L_c and r. Details of parameter reestimation can be found in [7]. As it can be interpreted from the Multinomial Subspace Model, the adapted weights are obtained by multiplying UBM weights supervector to an offset supervector Lr. However, this multiplicative relation, may lead to inaccurate results specially for short utterances, where the adaptation data is sparse. Since no prior distribution is assumed for r, it might be poorly estimated in the case of short utterances and multiplying Lr to UBM weights supervector may result in inaccurately adapted weights. To tackle this problem an i-vector like GMM weight supervector decomposition is elaborated in next section.

III. GMM Weight Supervector Decomposition

The basic assumption of this framework is that the c^{th} Gaussian weight of the adapted GMM (w_c) can be decomposed as follows.

$$w_c = b_c + L_c r \tag{4}$$

where b_c is the UBM weight of corresponding component. L_c denotes the c^{th} row of the matrix L, which is a matrix of dimension $C \times \rho$ spanning a low-dimensional subspace. r is a low dimensional vector that best describe the utterance-dependent weight offset Lr. The subspace matrix L is estimated via factor analysis to represent the directions that best separate different speech recordings in a large development data set.

Like in the i-vector framework, the procedure of calculating L and r involves the Expectation-Maximization (E-M) algorithm. In the Expectation-step, L is assumed to be known and we try to update r. Similarly in the Maximization-step, r is assumed to be known and we try to update L. Each step is elaborated as follows.

A. Updating vector r

In the Expectation-step, vector r is estimated as follows.

1) Constrained optimization problem: Consider the following log-likelihood of data $\mathcal{X}$

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

substituting w_c by $b_c + L_c r$ results in

$$\log p(\mathcal{X}) = \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t} \log \left(b_c + L_c r\right) \tag{6}$$

or

$$\log p(\mathcal{X}) = \bar{\gamma}'(\mathcal{X}) \log \left(b + Lr\right) \tag{7}$$

where the log operates element-wise, $'$ denotes transpose and $\bar{\gamma}(\mathcal{X})$ is

$$\bar{\gamma}(\mathcal{X}) = \sum_{t} \begin{bmatrix} \gamma_{1,t} & \cdots & \gamma_{C,t} \end{bmatrix}' \tag{8}$$

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

$$max\ f(r) \tag{9}$$

Subject to

$g(b + Lr) = 1$ Equality constraint

$b + Lr > 0$ Inequality constraint

where $f(r) = \bar{\gamma}'(\mathcal{X}) \log \left(b + Lr\right)$ and g is a row vector of dimension C with all elements equal to 1. This constrained optimization problem has an analytical solution for a square full-rank L (The proof for this relation is given in Appendix A).

$$r(\mathcal{X}) = L^{-1} \left[\frac{1}{\tau} \bar{\gamma}(\mathcal{X}) - b\right] \tag{10}$$

For a skinny L, where the number of rows is more than the number of columns, solving this constrained optimization problem involves using iterative optimization approaches. There are different tools to solve this large-scale constrained maximization problem in a reasonable time for each utterance, such as L-BFGS (limited-memory Broyden-Fletcher-Goldfarb-Shanno) [11]. However, using these methods for a large number of utterances can be too time-consuming. To decrease the computation time, we relax the constraints and convert the problem to an unconstrained optimization by the following simple techniques.

2) Relaxing the equality constraint: The equality constraint is

$$gb + gLr = 1 \tag{11}$$

We know that UBM weights sum up to 1 or $gb = 1$. Hence

$$gLr = 0 \tag{12}$$

If g is orthogonal to all columns of L, i.e., $gL = 0$, the constraint 12 holds for any possible r. In maximization-step, L is calculated such that $gL = 0$ holds.

3) Relaxing the inequality constraint: As can be seen in Equation 9 there are C inequality constraints. If any inequality constraints is violated, the cost function of 9 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. 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 r in a reasonable time. In this report, we use a simple gradient ascent method with the following updating formula.

$$r_i = r_{i-1} + \alpha_E \bigtriangledown f(r_{i-1}) \tag{13}$$

$$\bigtriangledown f(r) = L' \frac{[\bar{\gamma}'(\mathcal{X})]}{[b + Lr(\mathcal{X})]} \tag{14}$$

where $\frac{[A]}{[B]}$ denotes the element-wise division of matrix A and matrix B, subscript i is the index for gradient ascent iteration, α_E is the learning rate and $\bigtriangledown$ denotes gradient. In this algorithm, α_E is reduced at each unsuccessful step (e.g. halved). Unsuccessful is when $f(r)$ decreases or any of the inequality constraints are violated.

5) Initialization: Like in 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 L given in Relation 10. For a skinny L, the constrained optimization problem has no analytical solution. However, we can use the left pseudo-inverse instead of the inverse to obtain a vector of the same dimension as r.

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

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

$$r_0 = \theta r_{unfeasible} \tag{16}$$

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

6) Prior Distribution of r: By assuming a prior distribution on r the objective function of the above optimization problem changes to

$$v(r) = e^{\bar{\gamma}'(\mathcal{X}) \log (b+Lr)} \times g(r) \tag{17}$$

where $g(r)$ is the prior distribution of r. Now we have to apply Maximum-a-Posteriori (MAP) estimation to calculate the corresponding r for each utterance. Since the mean of the distribution of w is b and in this method $w = b + Lr$, the mean of the distribution of r is zero. On the other hand, for short utterances we expect that the adapted weights be similar to UBM weights b. Therefore, assuming that the prior of r is Gaussian or Laplacian with a small variance to keep w non-negative seems to be appropriate. For simplicity the prior distribution of r is assumed to be Gaussian in this report. Therefore the objective function and its gradient change to

$$f(r) = \bar{\gamma}'(\mathcal{X}) \log (b + Lr) - \frac{1}{2\delta^2} r'r \tag{18}$$

$$\bigtriangledown f(r) = L' \frac{[\bar{\gamma}'(\mathcal{X})]}{[b + Lr(\mathcal{X})]} - \frac{r}{\delta^2} \tag{19}$$

As it can be understood from this equation, the prior distribution forces r to have small elements and the standard deviation δ of prior distribution controls the balance between the two parts of the objective function.

B. Updating matrix L

In the Maximization-step, assuming r is known for all utterances in the training database, matrix L can be obtained by solving the following constrained optimization problem.

$$\begin{aligned} & max\ h(L) \\ & \text{Subject to} \\ & g(b + Lr(\mathcal{X}(s))) = 1 \quad \text{Equality constraint} \\ & b + Lr(\mathcal{X}(s)) > 0 \qquad \text{Inequality constraint} \\ & s = 1, \cdots S \end{aligned} \tag{20}$$

where

$$h(L) = \sum_s \bar{\gamma}'(\mathcal{X}(s)) \log\left[b + Lr(\mathcal{X}(s))\right] \tag{21}$$

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 steps size.

All equality constraints can be simplified to a single constraint $gL = 0$ using the same trick mentioned in Section III-A2. To solve the resulting optimization problem with equality constraint $gL = 0$, we apply projected gradient algorithm [12].

$$L_i = L_{i-1} + \alpha_M \mathcal{P} \bigtriangledown h(L_{i-1}) \tag{22}$$

$$\bigtriangledown h(L) = \sum_s \frac{[\bar{\gamma}(\mathcal{X}(s))]}{[b + Lr(\mathcal{X}(s))]} r'(\mathcal{X}(s)) \tag{23}$$

$$\mathcal{P} = I - \frac{1}{C} g'g \tag{24}$$

where subscript i is the index for gradient ascent iterations, α_M is the learning rate, I is an identity matrix of size C. In this algorithm, α_M is reduced at each unsuccessful step (e.g. halved) and increased in each successful step (multiplied by 1.5). Unsuccessful is when $h(L)$ decreases or any of the inequality constraints are violated.

1) Initialization: We use Principle Component Analysis (PCA) for initialization of L. In other words, we first form matrix W from the maximum likelihood estimations of GMM weights for all training utterances as follows.

$$W = \left[\frac{\bar{\gamma}(\mathcal{X}(1))}{\tau(1)}, \cdots, \frac{\bar{\gamma}(\mathcal{X}(s))}{\tau(s)}, \cdots, \frac{\bar{\gamma}(\mathcal{X}(S))}{\tau(S)}\right] \tag{25}$$

Then, the first ρ principle components of W are used as initial point of L for maximization of $h(L)$.

IV. GMM Weight subspace for language recognition

A. Classifier

A multilayer perceptron (MLP) is a supervised, feedforward neural networks, which is widely applied to regression and classification problems [13]. An MLP usually utilizes a derivative-based optimization algorithm such as backpropagation for training the network. The main deficiency of MLPs is that their objective function is non-convex and a derivative-based optimization algorithm may get stuck in a local minimum. This is more challenging in the case of a high dimensional input space or when more hidden layers are used. DBNs try to solve this problem by proper initialization of the network. This initialization is performed for each layer independently in a greedy approach. For initializing each layer, the hidden variables of the previous layer are considered as observed variables. This approach has found many applications in image recognition and speech technology [14], [15]. In this research, we applied a four-layer DBN where the input layer has 1000 neurons (dimension of the input space), the first hidden layer consists of 1000 neurons, the second hidden layer consists of 200 neurons and the output layer has 6 neurons (the number of language categories).

B. Training and Testing

The principle of the proposed language recognition approach is illustrated in Figure 1. As it can be interpreted from this figure, in the training phase, each utterance in the train data set is converted to a vector using the aforementioned utterance modeling approach. Then, the obtained vectors along with their corresponding language label are used to train a DBN.

In the testing phase, the proposed utterance modeling approach is applied to extract the feature vector from the utterance of an unseen speaker. Then the trained classifier uses the extracted vector to recognize the language of the test speaker.

V. Experimental Setup

A. Database

RATS P1 evaluation corpus is partially sourced from some 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, was also collected and added to the database. All recordings were retransmitted through eight different communication channels. The 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 P1 evaluation corpus is divided into three disjoint databases namely training, development and evaluation. Table I lists the number of utterances in each category for training, development and evaluation data sets. 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 120s, 30s, 10s and 3s.

B. Performance Measure

The effectiveness of the proposed method is evaluated using the percentage of correctly classified utterances (P_{cc}) and Confusion Matrix. P_{cc} is a simple performance measure which can be calculated

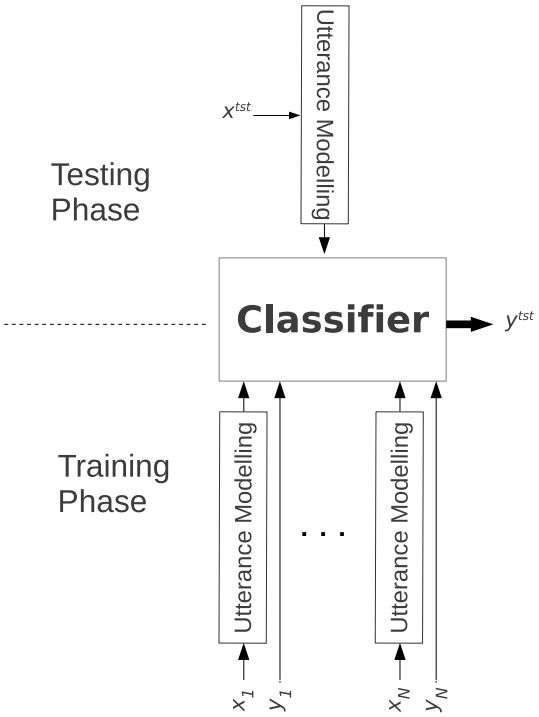

Fig. 1: *The block diagram of the language recognition systems in training and testing phases.*

TABLE I: *The number of utterances for each category in Training, Development and Evaluation databases.*

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**

using the following relation.

$$P_{\rm cc} = \frac{N_{cc}}{N_T} \times 100 \tag{26}$$

where N_{cc} and N_T denote the number of correctly classified utterances and the total number of utterances in the test data set respectively.

VI. RESULTS

In this section, the performance of the proposed method is investigated using some preliminary experiments. The feature extraction stage used in this work is based on a Shifted Delta cepstral representation. Speech is windowed at 20ms with a 10ms 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 scheme and concatenation to the static cepstral coefficients. Speech activity detection based on a Brno university of technology neural network implementation is then applied to remove the silence.

Table II lists the $P_{\rm cc}$ for three utterance modeling approaches. In the first one, labeled as GWS, each utterance is modeled by its corresponding GMM weight supervector obtained using the conventional maximum-likelihood method. The second utterance modeling method is standard i-vector framework and the third method, labeled as r-vector, is the proposed approach. r-vectors are obtained after six E-M iterations. The number of successful gradient descent iterations in the E-step and the M-step are 8 and 5 respectively. The Table II shows that the proposed weight supervector decomposition improves the results of GWS by more than 4%. However, as expected, the accuracy of the introduced method is lower than the i-vector based system.

TABLE II: *Comparison of i-vector, r-vector and GWS in language identification. The results are given in* P_{cc}.

System Configuration	Evaluation Dataset			
	120s	30s	10s	3s
GWS	86	68	51	38
r-vector	89	73	57	39
i-vector	90	78	61	48
Fusion	90	81	68	55

A. Fusion of the i-vector and r-vector systems

Score level fusion has been carried out to boost the recognition accuracy. The fusion is performed by training a two hidden layer DBN (layer one and two have 100 and 20 neurons respectively) on the outputs of DBNs on the development dataset. The last row of Table II shows the results of the fusion. As it can be interpreted from this table, fusion of r-vector with i-vector increases the recognition accuracy by 5% (the relative improvement is 17%) . The improvement is more evident in the case of short utterances.

VII. Future Work

The standard i-vector modeling is based on adapting only the GMM mean components. We believe that the subspace weight adaption we proposed could be integrated in the classical i-vector extraction in several ways. One possible combination is to extract an i-vector, which will be based on GMM supervector modeling, in two steps. In the fist step, we will start by adapting the weights of the universal background model to the given speech utterance. Then in the second step we will extract the new i-vector based on these new weights. This new regime of extracting the i-vector representation can be very useful for speaker as well as language recognition. Other combinations between the two subspace techniques can also be explored.

We also would like to apply the subspace techniques to speech recognition in a low resource language scenario using semi-continuous Hidden Markov Models. By exploiting the weight and mean subspaces learned from the phoneme inventory of multiple languages, we should be able to more reliably adapt the model for a new target language.

Another intended application is using the proposed method in updating weights for rapid speaker adaptation of large vocabulary speech recognition systems instead of common non-negative matrix factorization (NMF) based schemes.

APPENDIX A

The function to be maximized is

$$f(r) = \bar{\gamma}'(\mathcal{X}) \log (b + Lr) \tag{27}$$

The equality constraint is

$$g\,(b + Lr) = 1 \tag{28}$$

By introducing a Lagrange multiplier we reach

$$z(x) = \bar{\gamma}'(\mathcal{X}) \log (b + Lr) + \beta\,[1 - g(b + Lr)] \tag{29}$$

By differentiating 29 with respect to r and setting the result to 0 we reach

$$\frac{[\bar{\gamma}(\mathcal{X})]'}{[b + Lr(\mathcal{X})]'} L = \beta g L \tag{30}$$

Since L is a full rank matrix, we can drop it from both sides of Equation 30.

$$\frac{[\bar{\gamma}(\mathcal{X})]'}{[b + Lr(\mathcal{X})]'} = \beta g \tag{31}$$

hence

$$\bar{\gamma}(\mathcal{X}) = \beta\,(b + Lr(\mathcal{X})) \tag{32}$$

Considering the constraint mentioned in relation 28 and multiplying with g on both sides of relation 32

$$g\bar{\gamma}(\mathcal{X}) = \beta g\,(b + Lr(\mathcal{X})) \tag{33}$$

or

$$\tau = \beta \tag{34}$$

Therefore,

$$\bar{\gamma}(\mathcal{X}) = \tau\,(b + Lr(\mathcal{X})) \tag{35}$$

from which the relation 10 is obtained.

Note that since τ and all elements of $\bar{\gamma}(\mathcal{X})$ in relation 35 are non-negative, the result of 10 keeps all elements of $b + Lr(\mathcal{X})$ non-negative as well.

REFERENCES

[1] M. Li, K. J. Han, and S. Narayanan, "Automatic speaker age and gender recognition using acoustic and prosodic level information fusion," *Computer Speech and Language*, vol. 27, no. 1, pp. 151 – 167, 2013.

[2] X. Zhang, K. Demuynck *et al.*, "Rapid speaker adaptation in latent speaker space with non-negative matrix factorization," *Speech Communication*, 2013.

[3] M. H. Bahari, R. Saeidi, and D. van Leeuwen, "Accent recognition using i-vector, gaussian mean supervector and gaussian posterior probability supervector for spontaneous telephone speech," in *Proceedings ICASSP2013*, 2013, pp. 7344–7348.

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

[5] 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.

[6] M. H. Bahari, M. McLaren, and D. Van Leeuwen, "Age estimation from telephone speech using i-vectors," in *Interspeech*, 2012, pp. 506–509.

[7] M. Kockmann, L. Burget, O. Glembek, L. Ferrer, and J. Černockỳ, "Prosodic speaker verification using subspace multinomial models with intersession compensation," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[8] O. Glembek, P. Matejka, L. Burget, and T. Mikolov, "Advances in phonotactic language recognition," *Interspeech08*, pp. 743–746, 2008.

[9] M. Soufifar, M. Kockmann, L. Burget, O. Plchot, O. Glembek, and T. Svendsen, "ivector approach to phonotactic language recognition," in *Proc. of Interspeech*, 2011, pp. 2913–2916.

[10] M. Soufifar, S. Cumani, L. Burget, J. Cernocky *et al.*, "Discriminative classifiers for phonotactic language recognition with ivectors," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4853–4856.

[11] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[12] J. A. Snyman, *Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms*. Springer Science+ Business Media, 2005, vol. 97.

[13] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 683–697, 1992.

[14] H. Lee, Y. Largman, P. Pham, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," *Advances in neural information processing systems*, vol. 22, pp. 1096–1104, 2009.

[15] H. Lee, C. Ekanadham, and A. Ng, "Sparse deep belief net model for visual area v2," *Advances in neural information processing systems*, vol. 20, pp. 873–880, 2008.