Odyssey 2012
The Speaker and Language Recognition Workshop
25-28 June 2012, Singapore

Factor Analysis of Mixture of Auto-Associative Neural Networks for Speaker Verification

Sri Garimella, Hynek Hermansky

Center for Language and Speech Processing
Department of Electrical and Computer Engineering,
The Johns Hopkins University, Baltimore, USA
{sivaram, hynek}@jhu.edu

Abstract

This paper introduces the theory of factor analysis of the mixture of Auto-Associative Neural Networks (AANNs) with application in speaker verification. First, we formulate the problem of learning a low-dimensional subspace in part of the mixture of AANNs parameter space, and subsequently derive the update equations by minimizing loss function of the mixture. Second, we apply this technique to build a neural network based speaker verification system, in which the low-dimensional subspace is trained to capture both speaker and channel variabilities. This low-dimensional (or i-vector) representation is used as features for the probabilistic linear discriminant analysis (PLDA) model, as in state-of-the-art speaker verification systems. The proposed factor analysis approach shows promising results on the NIST-08 speaker recognition evaluation (SRE), and yields 18% relative improvement in minimum detection cost function (minDCF) over the previously proposed subspace based mixture of AANNs system.

1. Introduction

The goal of the speaker verification is to verify whether a given utterance belongs to a claimed speaker or not based on a sample utterance from claimed speaker. In other words, the task is to verify whether a given two utterances of a speaker verification trial belong to the same speaker or not. Traditional speaker verification systems use likelihood ratio between Gaussian Mixture Model (GMM) based Universal Background Model (UBM) and its maximum aposteriori (MAP) adapted speaker-specific model for making decision [1].

Recently, several factor analysis approaches have been proposed for GMM based speaker verification systems [2, 3, 4, 5, 6]. These methods assume that the supervector of means of a GMM (speaker-specific part of GMM parameter space) is not observable, and further constrain it to lie in a low-dimensional subspace with standard normal distribution as prior. The subspace is learned using the maximum likelihood principle on large amounts of development data containing multiple utterances from several speakers. The main advantage of this approach is that it facilitates robust point estimates of coordinates (also known as i-vector [4]) of a given utterance in the subspace. In [2], separate subspaces corresponding to both speaker and channel are learned, and where as in [4], a single subspace known as total variability space is learned. State-of-the-art GMM based speaker verification systems treat i-vectors as features and subsequently train a probabilistic linear discriminant analysis (PLDA) model [7, 8]. More recently, discriminative training of PLDA [9] and length normalization of i-vectors [10] have significantly improved the performance of speaker verification systems. Further, a different back-end classifier such as Kernel Partial Least Squares (KPLS) seems to be complementary to PLDA [11].

In the past, Auto-Associative Neural Networks (AANNs) have been proposed as an alternative to GMMs for modeling the distribution of data [12]. An AANN is a feed-forward neural network trained to reconstruct its input at its output through a hidden compression layer [13]. AANNs have several advantages compared to the GMMs - they relax the assumption of feature vectors to be locally normal and can capture higher order moments. In [12, 14], AANNs have been applied to speaker verification. However, they did not meet the performance of GMM based systems. This could be due to the limitation of a single AANN being used for modeling the entire acoustic space, and (or) due to the lack of subspace methods when training speaker-specific AANN models. To address this issue, we have proposed to use subspace based mixture of AANNs for speaker verification [15]. The mixture consisted of several AANNs tied using posterior probabilities of various broad phoneme classes, which are obtained from a separate multilayer perceptron (MLP) classifier trained on labeled data. Supervector of last layer weight matrices (after vectorizing) of all AANN components was considered to be the speaker-specific part of the mixture. The supervector was first retrained using each utterance (and thus modeled as observable), and then projected on to a low-dimensional subspace to reduce its dimensionality. The subspace was learned to preserve most of the variability of supervectors in the weighted least squares sense (analogous to principal component analysis (PCA)) [15].

In this paper, we propose a novel factor analysis technique for the mixture of AANNs to learn a low-dimensional subspace in the supervector space of last layer weights. We assume that the speaker-specific supervector is not directly observable[1], but constrained to be in a low-dimensional subspace. The subspace is learned by directly minimizing loss function of the

The research presented in this paper was partially funded by the IARPA BEST program under contract Z857701, the DARPA RATS program under D10PC20015 and the JHU Human Language Technology Center of Excellence. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the IARPA or DARPA or JHU HLTCOE.

[1] In other words, we do not directly adapt the last layer weights of the mixture of AANNs based Universal Background Model to obtain a speaker-specific model.

mixture on development data. Since loss function of the mixture of AANNs is different from that of the GMM, the update equations for training the subspace are also different as will be derived in this paper. The resultant low-dimensional (or i-vector) representation is used as features for the probabilistic linear discriminant analysis (PLDA) model, as in state-of-the-art GMM based speaker verification systems. The proposed neural network based speaker verification system is tested on the telephone conditions of the NIST-08 speaker recognition evaluation (SRE). Experimental results show that the proposed factor analysis technique shows promising results, and yields 18% relative improvement in minimum detection cost function (minDCF) over the previously proposed subspace based mixture of AANNs method [15].

The remainder of the paper is organized as follows. The earlier work on AANNs and mixture of AANNs is summarized in Section 2. Section 3 describes the proposed factor analysis technique for mixture of AANNs. The neural network based speaker verification system is described in Section 4. Experimental results are presented in Section 5. Conclusions are provided in Section 6.

2. Related Work

2.1. AANNs

AANN is a five layer feed-forward neural network trained to reconstruct the input feature vector at its output through a hidden compression layer [13]. It consists of three non-linear hidden layers between the linear input and output layers. The second hidden layer contains fewer nodes than the input layer, and is known as the compression layer.

For an input feature vector $\mathbf{f}_i$, the network produces an output $\mathbf{g}\left(\mathbf{f}_i, \Theta\right)$ which depends both on the input $\mathbf{f}_i$ and the parameters Θ of the network (the set of weights and biases). While training the network, the parameters Θ are adjusted to typically minimize the squared error loss function in (1):

$$\min_{\{\Theta\}} \sum_{i=1}^{n} \|\mathbf{f}_i - \mathbf{g}\left(\mathbf{f}_i, \Theta\right)\|^2, \tag{1}$$

where n is the number of feature vectors of the training data. The network parameters are learned using the stochastic gradient descent algorithm. The gradient of the loss function with respect to any parameter can be efficiently computed using the the chain rule of calculus which results in a standard error back-propagation algorithm.

Once the AANN is well trained, the average reconstruction error of input vectors that are drawn from the distribution of the training data will be small compared to vectors drawn from a different distributions [12]. Previously proposed AANN based speaker verification systems exploited this principle [12, 14]. A single AANN is trained on large amounts of data containing multiple speakers. This AANN captures speaker independent distribution of the input acoustic feature vectors, and is used as the universal background model (UBM). For each speaker in the enrollment set, a speaker-specific AANN model is obtained by retraining the entire UBM-AANN using enrollment data. During the test phase, the average reconstruction error of the test data is computed under both UBM-AANN model and the claimed speaker AANN model. The final score of each trial for making decision is computed as the difference between these average reconstruction errors.

2.2. Mixture of AANNs

Mixture of AANNs consists of several independent AANNs, each modeling only part of the input vector space [15]. The means by which a given feature vector of the data is assigned to an appropriate AANN is by using a separate MLP trained on labeled data to estimate the posterior probability of the underlying class. The objective function below is minimized for training the mixture:

$$\min_{\{\Theta_1,\ldots,\Theta_c\}} \sum_{i=1}^{n}\sum_{j=1}^{c} \gamma_i^j \|\mathbf{f}_i - \mathbf{g}\left(\mathbf{f}_i, \Theta_j\right)\|^2, \tag{2}$$

where c is the number of mixture components or classes, Θ_j denotes the parameters (weights and biases) of j^{th} AANN of the mixture, γ_i^j is the posterior probability of j^{th} class given i^{th} feature vector $\mathbf{f}_i$, and $\mathbf{g}\left(\mathbf{f}_i, \Theta_j\right)$ represents the output of the j^{th} AANN when its input is $\mathbf{f}_i$. Note that (2) can be written as:

$$\sum_{j=1}^{c} \left(\min_{\{\Theta_j\}} \sum_{i=1}^{n} \gamma_i^j \|\mathbf{f}_i - \mathbf{g}\left(\mathbf{f}_i, \Theta_j\right)\|^2 \right). \tag{3}$$

It can be observed from (3) that each AANN component can be independently trained using the standard back-propagation approach. The only modification in the back-propagation algorithm when training the j^{th} AANN component is to multiply the error vector corresponding to the input $\mathbf{f}_i$ with γ_i^j.

3. Factor analysis of mixture of AANNs

The idea of factor analysis is to constrain the supervector of last layer weights of AANN mixture components to lie in a low-dimensional subspace such that it minimizes the mixture of AANNs loss function over the entire development data. In this process, the rest of the mixture parameters are held fixed at values learned during speaker independent training such as UBM. The notations used in this section are summarized below.

m - number of speakers
$r(s)$- number of sessions of the s^{th} speaker
$n(l,s)$ - number of frames in l^{th} session of s^{th} speaker
c - number of classes
$\mathbf{f}_{i,l,s}$ - i^{th} acoustic feature vector of l^{th} session, s^{th} speaker
$\gamma_{i,l,s}^j$ - posterior probability that $\mathbf{f}_{i,l,s}$ belongs to j^{th} class
$\mathbf{h}_{i,l,s}^j$ - fourth layer output of j^{th} AANN when the input is $\mathbf{f}_{i,l,s}$
$\mathbf{W}_{l,s}^j$ - last layer weight matrix of j^{th} AANN for l^{th} session, s^{th} speaker
$\mathbf{b}^j$ - output bias vector of the j^{th} AANN component
d - dimensionality of the input acoustic feature vector
d' - dimensionality of the fourth layer output

In the mixture of AANNs loss function below, we first vectorize $\mathbf{W}_{l,s}^j$ and then express it as a j^{th} part of the supervector so that a subspace structure can be imposed on it. The mixture of AANNs loss function with speaker and session specific last layer weights is given by

$$\begin{aligned} L &= \sum_{s=1}^{m}\sum_{l=1}^{r(s)}\sum_{i=1}^{n(l,s)}\sum_{j=1}^{c} \gamma_{i,l,s}^j \left\| \mathbf{f}_{i,l,s} - \mathbf{b}^j - \mathbf{W}_{l,s}^j \mathbf{h}_{i,l,s}^j \right\|_2^2 \\ &= \sum_{s=1}^{m}\sum_{l=1}^{r(s)}\sum_{i=1}^{n(l,s)}\sum_{j=1}^{c} \gamma_{i,l,s}^j \left\| \mathbf{f}_{i,l,s} - \mathbf{b}^j - \mathbf{H}_{i,l,s}^j \mathbf{w}_{l,s}^j \right\|_2^2 \end{aligned} \tag{4}$$

where,

$$\mathbf{w}_{l,s}^{j} = \text{Row ordered}(\mathbf{W}_{l,s}^{j}),$$

$$\mathbf{H}_{i,l,s}^{j} = \mathbf{I}_d \otimes \left(\mathbf{h}_{i,l,s}^{j}\right)^T = \begin{bmatrix} \left(\mathbf{h}_{i,l,s}^{j}\right)^T & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \left(\mathbf{h}_{i,l,s}^{j}\right)^T \end{bmatrix}_{d \times dd'}.$$

The dimensionality of $\mathbf{W}_{l,s}^{j}$ is $d \times d'$ and that of $\mathbf{w}_{l,s}^{j}$ is $dd' \times 1$. The vector $\mathbf{w}_{l,s}^{j}$ is obtained by arranging rows of $\mathbf{W}_{l,s}^{j}$ as columns one after the other. Matrix $\mathbf{H}_{i,l,s}^{j}$ is equal to the Kronecker product of $\mathbf{I}_d$ (a $d \times d$ identity matrix) and $\left(\mathbf{h}_{i,l,s}^{j}\right)^T$.

We can simplify (4) by constructing the following matrices. The objective is to replace the innermost summation over number of classes.

$$\mathbf{P}_{i,l,s} = \begin{bmatrix} \gamma_{i,l,s}^{1}\mathbf{I}_d & & \mathbf{0} \\ & \gamma_{i,l,s}^{2}\mathbf{I}_d & \\ & & \ddots \\ \mathbf{0} & & \gamma_{i,l,s}^{c}\mathbf{I}_d \end{bmatrix}_{cd \times cd},$$

$$\mathbf{H}_{i,l,s} = \begin{bmatrix} \mathbf{H}_{i,l,s}^{1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{H}_{i,l,s}^{c} \end{bmatrix}_{cd \times cdd'},$$

$$\mathbf{x}_{i,l,s} = \begin{bmatrix} \mathbf{f}_{i,l,s} \\ \vdots \\ \mathbf{f}_{i,l,s} \end{bmatrix}_{cd \times 1}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}^1 \\ \vdots \\ \mathbf{b}^c \end{bmatrix}_{cd \times 1},$$

$$\mathbf{w}_{l,s} = \begin{bmatrix} \mathbf{w}_{l,s}^{1} \\ \vdots \\ \mathbf{w}_{l,s}^{c} \end{bmatrix}_{cdd' \times 1}. \text{ (supervector)}$$

Using the matrices above, (4) can be written as

$$L(\mathbf{w}_{1,1}, \ldots, \mathbf{w}_{r(m),m}) = \sum_{s=1}^{m}\sum_{l=1}^{r(s)}\sum_{i=1}^{n(l,s)} \left[\mathbf{x}_{i,l,s} - \mathbf{b} - \mathbf{H}_{i,l,s}\mathbf{w}_{l,s}\right]^T \mathbf{P}_{i,l,s} \left[\mathbf{x}_{i,l,s} - \mathbf{b} - \mathbf{H}_{i,l,s}\mathbf{w}_{l,s}\right]. \quad (5)$$

The factor analysis model (or subspace constraint) for the supervector of last layer weights $\mathbf{w}_{l,s}$ is

$$\mathbf{w}_{l,s} \equiv \mathbf{w}_{ubm} + \mathbf{T}\,\mathbf{q}_{l,s},$$

where $\mathbf{w}_{ubm}$ represents the speaker independent (UBM) supervector of last layer weights, $\mathbf{T}$ is a matrix having fewer columns than rows representing the common low-dimensional subspace, and $\mathbf{q}_{l,s}$ is a vector of coordinates in the subspace or an i-vector associated with the l^{th} session of s^{th} speaker. By substituting this factor analysis model in (5), (note that the loss function depends only on $(\mathbf{T}, \{\mathbf{q}_{l,s}\})$)

$$L\left(\mathbf{T}, \mathbf{q}_{1,1}, \ldots, \mathbf{q}_{r(m),m}\right) = \sum_{s=1}^{m}\sum_{l=1}^{r(s)}\sum_{i=1}^{n(l,s)} \left[\mathbf{x}_{i,l,s} - \mathbf{b} - \mathbf{H}_{i,l,s}\left(\mathbf{w}_{ubm} + \mathbf{T}\mathbf{q}_{l,s}\right)\right]^T \mathbf{P}_{i,l,s} \left[\mathbf{x}_{i,l,s} - \mathbf{b} - \mathbf{H}_{i,l,s}\left(\mathbf{w}_{ubm} + \mathbf{T}\mathbf{q}_{l,s}\right)\right]. \quad (6)$$

Let us define

$$\mathbf{e}_{i,l,s} \doteq \mathbf{x}_{i,l,s} - \mathbf{b} - \mathbf{H}_{i,l,s}\mathbf{w}_{ubm}. \quad (7)$$

By substituting the expression above in (6),

$$L\left(\mathbf{T}, \mathbf{q}_{1,1}, \ldots, \mathbf{q}_{r(m),m}\right) = \sum_{s=1}^{m}\sum_{l=1}^{r(s)}\sum_{i=1}^{n(l,s)} \left[\mathbf{e}_{i,l,s} - \mathbf{H}_{i,l,s}\mathbf{T}\mathbf{q}_{l,s}\right]^T \mathbf{P}_{i,l,s} \left[\mathbf{e}_{i,l,s} - \mathbf{H}_{i,l,s}\mathbf{T}\mathbf{q}_{l,s}\right]. \quad (8)$$

Let us define the statistics

$$\mathbf{F_1}\left(l,s\right) \doteq \sum_{i=1}^{n(l,s)} \mathbf{H}_{i,l,s}^T \mathbf{P}_{i,l,s}\mathbf{e}_{i,l,s} \quad (9)$$

$$\mathbf{F_2}\left(l,s\right) \doteq \sum_{i=1}^{n(l,s)} \mathbf{H}_{i,l,s}^T \mathbf{P}_{i,l,s}\mathbf{H}_{i,l,s} \quad (10)$$

We can rewrite (8) using (9) and (10) as,

$$L\left(\mathbf{T}, \mathbf{q}_{1,1}, \ldots, \mathbf{q}_{r(m),m}\right) = \sum_{s=1}^{m}\sum_{l=1}^{r(s)} \left(\sum_{i=1}^{n(l,s)} \mathbf{e}_{i,l,s}^T \mathbf{P}_{i,l,s}\mathbf{e}_{i,l,s} \right) - 2\mathbf{q}_{l,s}^T\mathbf{T}^T\mathbf{F_1}\left(l,s\right) + \mathbf{q}_{l,s}^T\mathbf{T}^T\mathbf{F_2}\left(l,s\right)\mathbf{T}\mathbf{q}_{l,s}. \quad (11)$$

The low-dimensional subspace $\mathbf{T}$ can be learned by minimizing the loss function in (11) using the coordinate descent. In the first step, (11) is minimized with respect to $\{\mathbf{q}_{l,s}\}$ by keeping $\mathbf{T}$ fixed. In the second step, (11) is minimized with respect to $\mathbf{T}$ by keeping $\{\mathbf{q}_{l,s}\}$ fixed at the values found in step one. Note that the loss function is convex in each step, and therefore the optima is found by setting the gradient of the loss function with respect to the corresponding variable to zero. The steps are repeated until convergence.

Differentiating (11) with respect to $\mathbf{q}_{l,s}$ and setting it to zero yields,

$$\frac{\partial L}{\partial \mathbf{q}_{l,s}} = \mathbf{0} \quad \Rightarrow \quad \left[-2\mathbf{T}^T\mathbf{F_1}\left(l,s\right) + 2\mathbf{T}^T\mathbf{F_2}\left(l,s\right)\mathbf{T}\mathbf{q}_{l,s}\right] = \mathbf{0}$$

$$\mathbf{q}_{l,s} = \left[\mathbf{T}^T\mathbf{F_2}\left(l,s\right)\mathbf{T}\right]^{-1}\mathbf{T}^T\mathbf{F_1}\left(l,s\right) \quad (12)$$

Differentiating (11) with respect to $\mathbf{T}$ and setting it equal to zero yields,

$$\frac{\partial L}{\partial \mathbf{T}} = \mathbf{0} \Rightarrow \sum_{s=1}^{m}\sum_{l=1}^{r(s)} -2\mathbf{F_1}\left(l,s\right)\mathbf{q}_{l,s}^T + 2\mathbf{F_2}\left(l,s\right)\mathbf{T}\mathbf{q}_{l,s}\mathbf{q}_{l,s}^T = \mathbf{0}, \quad (13)$$

where we solve for $\mathbf{T}$ by solving a set of linear equations involving entries of the matrix $\mathbf{T}$.

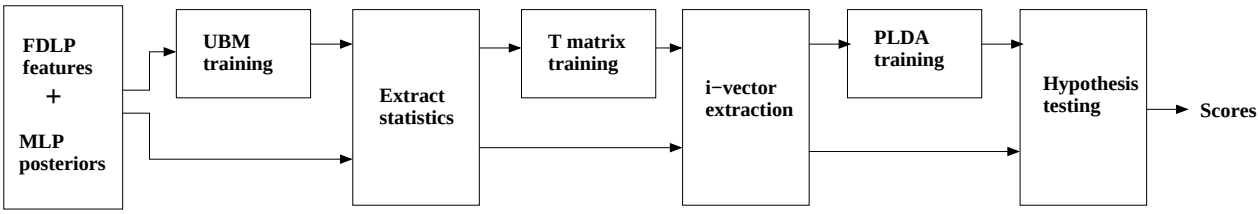

Figure 1: Block schematic of the neural network based speaker verification system.

4. Speaker Verification Systems

4.1. Proposed Neural Network System

The block diagram of the neural network based speaker verification system is shown in the Fig. 1. The description of various components of the system is provided below.

4.1.1. Acoustic Features

The acoustic features used in our experiments are 39 dimensional frequency domain linear prediction (FDLP) features [16]. In this technique, sub-band temporal envelopes of speech are first estimated in narrow sub-bands (96 linear bands). These sub-band envelopes are then gain normalized to remove reverberation and channel artifacts. After normalization, the frequency axis is warped to 37 Mel bands in the frequency range of 125-3800 Hz to derive a gain normalized mel scale energy representation of speech. This is similar to the mel spectrogram obtained in conventional mel frequency cepstral coefficients (MFCC) feature extraction. These mel band energies are converted to cepstral coefficients by applying a log and Discrete Cosine Transform (DCT). The top 13 cepstral coefficients along with derivative and acceleration components are used as features, yielding 39 dimensional feature vectors. Finally, a subset of these feature vectors corresponding to speech are selected based on the voice activity detection information.

4.1.2. Posteriors of Broad Phoneme Classes

A multilayer perceptron (MLP) is trained on 300 hours of conversational telephone speech (CTS) to estimate the posterior probabilities of 45 phonemes [17, 18]. The perceptual linear prediction (PLP) features are used for training [19]. The 45 phoneme posteriors are combined appropriately to obtain 5 broad phonetic class posteriors corresponding to vowels, fricatives, plosives, nasals and silence.

4.1.3. UBM

Mixture of AANNs based gender-specific UBMs are trained on a telephone development data set consisting of audio from the NIST 2004 speaker recognition database, the Switchboard-2 Phase III corpora and the NIST 2005 speaker recognition database. We use only 400 male and 400 female utterances each corresponding to about 17 hours of speech.

Each mixture consists of 5 AANN components corresponding to broad phoneme classes, and is trained using the FDLP features (see Section 4.1.1) to minimize the weighted reconstruction error as described in Section 2.2. The posterior probabilities of broad phoneme classes for training the mixture are obtained from an MLP described in Section 4.1.2. Each AANN component of the mixture has a linear input and a linear output

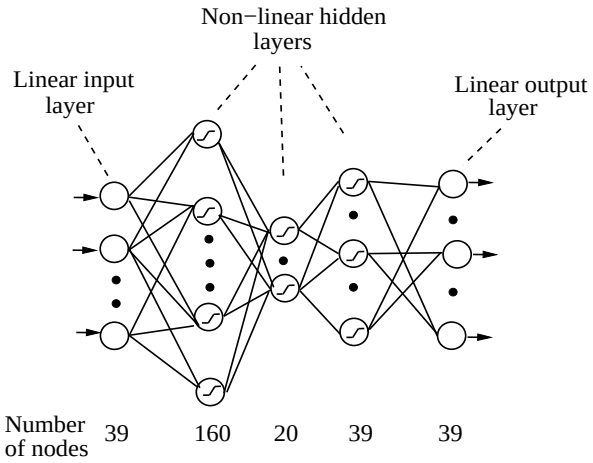

Figure 2: AANN component.

layer along with three nonlinear (tanh nonlinearity) hidden layers as shown in Fig. 2. Both input and output layers have 39 nodes corresponding to the dimensionality of the input FDLP features, 160 nodes in the first hidden layer, 20 nodes in the compression layer and 39 nodes in the third hidden layer. We have modified the Quicknet package for training the mixture of AANNs [20].

4.1.4. Statistics

The statistics in (9) and (10) are precomputed for each utterance that corresponds to a particular speaker and a session. Appropriate gender-specific UBM is used for computing the statistics. Note that we need to compute only few entries of $\mathbf{F_2}\,(l, s)$ as it is redundant. These statistics are sufficient for training the $\mathbf{T}$ matrix and extracting the i-vectors.

4.1.5. $\mathbf{T}$*-matrix Training*

Gender dependent low-dimensional subspaces ($\mathbf{T}$ matrices) are trained in part of the mixture of AANNs parameter space as described in Section 3. The development data for training the subspaces consists of Switchboard-2, Phases II and III; Switchboard Cellular, Parts 1 and 2 and NIST 2004-2005 SRE [3]. The total number of male and female utterances is 12266 and 14936 respectively. The number of columns of $\mathbf{T}$ matrix is set to be 180, and the number of rows[2] being 7605. We initialize the matrix with a Gaussian noise and learn the subspace as described in Section 3.

[2]Determined by configuration of the UBM.

4.1.6. i-vectors

Each utterance is converted to an i-vector using (12), using an appropriate gender-specific $\mathbf{T}$ matrix. All i-vectors are normalized to have unit length to reduce the mismatch during training and testing [10].

4.1.7. PLDA training

PLDA is a generative model of observations, in our case i-vectors [7, 8]. The i-vectors are assumed to be generated as

$$\mathbf{q}_{l,s} = \mu + \mathbf{\Phi}\beta_s + \epsilon_{l,s}, \tag{14}$$

where μ is an offset, $\mathbf{\Phi}$ is a matrix with fewer columns than rows, β_s is a latent identity variable having a normal distribution with mean zero and covariance matrix identity, and $\epsilon_{l,s}$ is a residual noise term assumed to be Gaussian with mean zero and full covariance matrix $\mathbf{\Sigma}$. Additionally, all latent variables are assumed to be independent.

Gender-specific PLDA models with dimension of subspace (number of columns of $\mathbf{\Phi}$) being 120 are trained using the same development data that is used for training $\mathbf{T}$ matrices (see Section 4.1.5). The maximum likelihood estimates of the model parameters $\{\mu, \mathbf{\Phi}, \mathbf{\Sigma}\}$ are obtained using an Expectation Maximization (EM) algorithm [7].

4.1.8. Hypothesis Testing

Given two i-vectors $\mathbf{q}_1$, $\mathbf{q}_2$ of a speaker verification trial, we need to test whether they belong to the same speaker ($\mathcal{H}_s$) or different speakers ($\mathcal{H}_d$). For the Gaussian PLDA of Section 4.1.7, the log-likelihood ratio can be computed in a closed-form as

$$\begin{aligned} score &= \log \frac{p\left(\mathbf{q}_1, \mathbf{q}_2 | \mathcal{H}_s\right)}{p\left(\mathbf{q}_1 | \mathcal{H}_d\right) p\left(\mathbf{q}_2 | \mathcal{H}_d\right)} \\ &= \log \frac{\mathcal{N}\left(\begin{bmatrix}\mathbf{q}_1 \\ \mathbf{q}_2\end{bmatrix}; \begin{bmatrix}\mu \\ \mu\end{bmatrix}, \begin{bmatrix}\mathbf{\Phi}\mathbf{\Phi}^T + \mathbf{\Sigma} & \mathbf{\Phi}\mathbf{\Phi}^T \\ \mathbf{\Phi}\mathbf{\Phi}^T & \mathbf{\Phi}\mathbf{\Phi}^T + \mathbf{\Sigma}\end{bmatrix}\right)}{\mathcal{N}\left(\begin{bmatrix}\mathbf{q}_1 \\ \mathbf{q}_2\end{bmatrix}; \begin{bmatrix}\mu \\ \mu\end{bmatrix}, \begin{bmatrix}\mathbf{\Phi}\mathbf{\Phi}^T + \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi}\mathbf{\Phi}^T + \mathbf{\Sigma}\end{bmatrix}\right)}, \end{aligned} \tag{15}$$

where $\mathcal{N}\left(.; \eta, \mathbf{\Lambda}\right)$ is a multivariate Gaussian density with mean η and covariance $\mathbf{\Lambda}$. The above score can be computed efficiently as described in [5, 10].

4.2. Baseline Neural Network System

The difference between the proposed system and the baseline system is in the training procedure of low-dimensional subspace or $\mathbf{T}$ matrix. For the baseline neural network system, gender dependent 180 dimensional subspaces are trained as described in [15] using the development data described in Section 4.1.5. As mentioned earlier, the idea is to retrain the supervector of weights of a mixture of AANNs based UBM for each utterance (and thus modeled as observable), and then obtain a low-dimensional subspace that preserves most of the variability of supervectors in a weighted least squares sense. Further, the i-vector of an utterance is obtained by projecting adapted supervector on to the low-dimensional subspace as described in [15]. The rest of the configuration in Fig. 1 remains unchanged for the baseline system.

4.3. GMM System

A gender-specific GMM based i-vector/PLDA system is also trained for comparison. The block diagram shown in the Fig. 1 is also applicable to the GMM system except for the MLP posteriors that are not used. Each GMM based UBM consists of 1024 mixture components with diagonal covariance. The male and female UBMs are trained using FDLP features extracted from 4324 and 5461 utterances of development data (described in Section 4.1.3) respectively. Gender-specific 400 dimensional total variability space ($\mathbf{T}$ matrix) is trained as described in [4]. The i-vectors of this space are length normalized and subsequently used for training the PLDA system with 250 dimensional subspace. Note that the development data used for training the $\mathbf{T}$ matrix and the PLDA model is same as that of the neural network system (see Section 4.1.5). Finally, the score of a given speaker verification trial is obtained using (15).

Table 1: Description of various telephone conditions of NIST-08.

C6	Telephone speech in training and test
C7	English language telephone speech in training and test
C8	English language telephone speech spoken by a native U.S. English speaker in training and test

Table 2: MIN DCF x10^3 and EER in % (shown in brackets) on conditions C6, C7 and C8 of NIST-08.

System	C6	C7	C8
Proposed 180 dim. i-vector	55.3 (10.6)	32.5 (6.1)	27.8 (4.9)
Baseline ([15]) 180 dim. i-vector	59.9 (12.0)	40.5 (8.2)	40.4 (7.7)
Baseline ([15]) 300 dim. i-vector	57.2 (11.5)	39.7 (7.6)	38.6 (7.1)
GMM 400 dim. i-vector	41.3 (7.0)	14.8 (2.8)	10.8 (2.1)

5. Experimental Results

The performance of speaker verification systems is tested on telephone conditions of NIST-2008 speaker recognition evaluation task. The description of various conditions can be found in Table 1. The minDCF and equal error rates (EER) of the baseline system (see Section 4.2) and the proposed neural network system (see Section 4.1) are shown in Table 2. It can be observed from the results that the proposed factor analysis approach yields 18% relative improvement in minDCF over the baseline neural network system.

Table 2 also lists the minDCF and EER of a baseline system ([15]) with 300 dimensional i-vectors, and a GMM system with 400 dimensional i-vectors (see Section 4.3) for comparison. We could not train the proposed system beyond 180 dimensional i-vectors due to the complexity of our training algorithm. It is to be noted that GMM based i-vector/PLDA system performs the best. However, further work on neural network based systems might close the existing performance gap, and bring forward possible advantages of this alternative nonlinear neural network based modeling in speaker verification.

6. Conclusions

This paper has developed the theory of factor analysis of the mixture of AANNs. We have applied the proposed factor analysis theory to build a neural network based speaker verification system. This system showed promising results on NIST-08 speaker recognition evaluation (SRE), and gave 18% relative improvement in minDCF over the baseline neural network system.

7. Acknowledgment

Authors would like to thank Samuel Thomas for providing the trained MLP for estimating posteriors of broad phoneme classes, Sriram Ganapathy for sharing the FDLP feature extraction code, and Daniel Garcia-Romero for sharing the PLDA software.

8. References

[1] D. Reynolds, T. Quatieri, R. Dunn, "Speaker verification using adapted Gaussian mixture models", *Digital Signal Processing*, pp. 19–41, 2000.

[2] P. Kenny, G. Boulianne, P. Oullet, P. Dumouchel, "Joint factor analysis versus eigenchannes in speaker recognition", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15(4), pp. 2072–2084, 2007.

[3] O. Glembek, L. Burget, N. Dehak, N. Brummer, P. Kenny, "Comparison of Scoring Methods used in Speaker Recognition with Joint Factor Analysis", *Proc of ICASSP 2009*.

[4] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19(4), pp. 788–798, 2010.

[5] N. Brummer, E. de Villiers, "The speaker partitioning problem", *Proc. of Odyssey 2010*.

[6] D. Garcia-Romero and C.Y. Espy-Wilson, "Joint Factor Analysis for Speaker Recognition Reinterpreted as Signal Coding Using Overcomplete Dictionaries", *Proc. of Odyssey 2010*.

[7] S.J.D. Prince, J.H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," *Proc. of ICCV 2007*.

[8] P. Kenny, "Bayesian speaker verication with heavytailed priors," *Proc. of Odyssey 2010*.

[9] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matejka, N. Brummer, "Discriminatively trained probabilistic linear discriminant analysis for speaker verification", *Proc. of ICASSP 2011*.

[10] D. Garcia-Romero, C.Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems", *Proc. of INTERSPEECH 2011*.

[11] B.V. Srinivasan, D. Garcia-Romero, D.N. Zotkin, R. Duraiswami, "Kernel partial least squares framework for speaker recognition", Proc. of INTERSPEECH 2011.

[12] B. Yegnanarayana S. Kishore, "AANN: an alternative to GMM for pattern recognition," *Neural Networks*, pp. 459–469, 2002.

[13] M.A. Kramer, "Nonlinear principal component analysis using auto-associative neural networks," *AICHE Journal*, pp. 233–243, 1991.

[14] K.S.R. Murty and B. Yegnanarayana, "Combining evidence from residual phase and MFCC features for speaker recognition," *IEEE Signal Processing Letters*, pp. 52–55, 2005.

[15] G.S.V.S. Sivaram, S. Thomas and H. Hermansky, "Mixture of Auto-Associative Neural Networks for Speaker Verification," *Proc. of INTERSPEECH-2011*.

[16] S. Ganapathy, J. Pelecanos, M.K. Omar, "Feature Normalization for Speaker Verification in Room Reverberation", *Proc. of ICASSP 2011.*

[17] M.D. Richard and R.P. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities", *Neural computation*, vol. 3(4), pp. 461–483, 1991.

[18] S. Ganapathy, S. Thomas and H. Hermansky, "Static and Dynamic Modulation Spectrum for Speech Recognition", Proc. of ISCA Interspeech, 2009.

[19] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech", *Journal of the Acoustical Society of America*, vol. 87(4), pp. 1738–1752, 1990.

[20] "The ICSI Quicknet Software Package", Available:http://www.icsi.berkeley.edu/Speech/qn.html