Fuzzy Restricted Boltzmann Machine based Probabilistic Linear Discriminant Analysis for Noise-Robust Text-Dependent Speaker Verification on Short Utterances

Sung-Hyun Yoon, Min-Sung Koh, and Ha-Jin Yu, *Member, IAENG*

***Abstract*— In the i-vector-based speaker verification system, it is important to compensate for session variability on the i-vector to improve speaker verification performance. Linear discriminant analysis (LDA) is widely used to compensate for session variability by reducing the dimensionality of the i-vector. Restricted Boltzmann machine (RBM)-based probabilistic linear discriminant analysis (PLDA) has been proposed to improve the session variability compensation ability of LDA. It can be viewed as a probabilistic approach of LDA using RBM. However, since the RBM does not consider uncertainties in obtaining the parameters, the representation capability of RBM-based PLDA is limited. For instance, many real-world speaker verifications must consider noisy environments, which make the compensated session variability uncertain. The fuzzy restricted Boltzmann machine (FRBM) was proposed to improve the capability of the RBM. It showed a more robust performance than that of the RBM. Hence, in this paper, we propose FRBM-based PLDA to improve the representation capability of RBM-PLDA by replacing all the parameters of RBM-PLDA with fuzzy numbers. An evaluation with Part 1 of Robust Speaker Recognition (RSR) 2015 was conducted. In the experimental results, the proposed algorithm shows a better compensation for phonetic variability that exists in short utterances, and a robust speaker verification performance in diverse noisy environments where phonetic and noise variabilities are challenging issues in real-world applications.**

***Index Terms*— Discriminant analysis, fuzzy restricted Boltzmann machine, i-vector, restricted Boltzmann machine, speaker verification.**

I. Introduction

In the field of speaker verification, the i-vector [1] has been widely used as a speaker representation in recent years. It is a set of factors of a fixed dimension that is extracted from an utterance of arbitrary duration. It originates from joint factor analysis (JFA) [2], [3]. In the JFA framework, an utterance is separately decomposed into speaker-dependent factors and session-dependent factors to compensate for session variability in the utterance. The session variability is the variability between the utterances of the same speaker. It may be caused by a change in the background noise, recording device, transmission channel, etc. Therefore, compensation for the session variability should be carried out to improve the robustness of the speaker verification system. However, Dehak [4] showed that the session-dependent factors estimated using JFA also contain some speaker information. In the i-vector framework, an utterance is represented by all factors (the so-called i-vector) that contain both speaker information and session information. The main purpose of estimating the i-vector is to fully exploit the speaker information. Compensating for session variability on the i-vector is carried out in most cases because the i-vector contains the session information. We can compensate for session variability in a lower-dimensional space than in the JFA framework, which enables more effective compensation.

Linear discriminant analysis (LDA) [5] is commonly used to compensate for session variability in the i-vector. It finds the lower-dimensional subspace that maximizes the between-class variability and minimizes the within-class variability. In the field of speaker verification, the between- and within-class variability correspond to speaker and session variability, respectively. We can compensate for session variability on the i-vector by projecting the i-vector into the subspace.

Probabilistic linear discriminant analysis (PLDA) was initially proposed for face recognition [6], [7]. It probabilistically models the between-class variability and within-class variability using latent variables. It can be viewed as a probabilistic approach to the LDA and has the effect of compensating the session variability. Unlike LDA, the PLDA model is a generative directed graphical model that can learn a probability distribution over its training dataset. In the i-vector-based speaker verification system, PLDA has been widely used to compute the likelihood ratio-based speaker verification score, and it has shown state-of-the-art performance [8]-[10].

The restricted Boltzmann machine (RBM) [11] is a generative neural network that can learn a probability distribution over its training dataset using hidden variables. It is an undirected graphical model and has the restriction that there are no intralayer connections. It can be used for various

Manuscript received December 9, 2019; revised May 22, 2020. This research was supported by Projects for Research and Development of Police Science and Technology under the Center for Research and Development of Police Science and Technology and Korean National Police Agency funded by the Ministry of Science, ICT and Future Planning (PA-J000001-2017-101).

Sung-Hyun Yoon is with the School of Computer Science, University of Seoul, Seoul 02504, Republic of Korea (e-mail: ysh901108@naver.com)

Min-Sung Koh is with the School of Computing and Engineering Sciences, Eastern Washington University, Cheney, WA 99004 USA (e-mail: mkoh@ewu.edu).

Ha-Jin Yu is with the School of Computer Science, University of Seoul, Seoul 02504, Republic of Korea (corresponding author to provide phone: +82-2-6490-2448; e-mail: hjyu@uos.ac.kr).

purposes in addition to dimensionality reduction.

The RBM represents the relationship between data and hidden variables using the network parameters. The representation capability of the RBM is limited because the RBM does not consider the uncertainty in obtaining the parameters. To improve the capability of RBM, the fuzzy restricted Boltzmann machine (FRBM) was proposed in [12] and [13]. FRBM is an extension of RBM that can represent the uncertainty of the parameters by replacing all the crisp parameters of the RBM with fuzzy numbers [14]. Moreover, it has been experimentally shown that the FRBM is more robust than the RBM in noisy conditions [12].

Furthermore, RBM-based probabilistic linear discriminant analysis (RBM-PLDA) was proposed [15]. It is a method that applies PLDA with RBM. The objective of RBM-PLDA is the same as that of PLDA, which is to compensate for the session variability on the i-vector. Additionally, the RBM-PLDA model and PLDA model are both generative graphical models. However, the RBM-PLDA model differs from the PLDA model. First, RBM-PLDA is used to reduce the dimensionality of the i-vector, similar to LDA; however, it is not used as the likelihood ratio-based scoring method, as is PLDA. Second, the RBM-PLDA model is considered an undirected graphical model, similar to RBM, whereas PLDA is considered a directed graphical model.

Although RBM-PLDA showed good performance results for speaker verification in [15], [16], RBM-PLDA can be extended for real-world speaker verifications that make the parameters uncertain [17]. Almost all real-world applications have several challenging issues, especially in applying/obtaining test utterances. For example:

1) It is difficult to obtain sufficiently long test utterances. The shorter the duration of the utterance is, the greater the uncertainty in estimating speaker representations, which severely degrades the speaker verification performance in many cutting-edge systems, even in i-vector-based systems [18]. Hence, it is very inconvenient to use if speakers are forced to talk for a long time. One of the most widely used methods for improving performance under short-duration conditions is to constrain all kinds of utterances to particular phrases (i.e., text-dependent ones) [19], [20].
2) It is difficult to obtain clean test utterances. There are many noisy environments (e.g., various types and levels) in real-world applications, which also increases the uncertainty in estimating speaker representations. However, it is impractical to limit the speakers to using the system only in particular environments (e.g., only in clean environments). In addition, it is impossible to perfectly recover clean utterance from a noisy utterance [21]. Therefore, noise-robust methods are important issues in real-world applications.

The RBM-PLDA model does not address these issues; rather, it considers session variability compensation in a clean speaker sample.

In this paper, we propose fuzzy RBM-based probabilistic linear discriminant analysis (FRBM-PLDA), where FRBM-PLDA is extended from RBM-PLDA by replacing the parameters of RBM-PLDA with fuzzy numbers. We consider the challenging issues that exist in real-world applications based on FRBM-PLDA so that the uncertainties associated with speaker representations can be incorporated in the proposed algorithm (i.e., FRBM-PLDA). It is herein shown that the performance for speaker verification is improved when compensating for the session variability on the i-vector using the FRBM-PLDA model, particularly in noisy speaker samples.

The remainder of this paper is organized as follows. Section II outlines the preliminaries required to elucidate the proposed methods. Section III introduces the proposed methods. Section IV describes the scoring methods used with the proposed methods. Section V describes the experimental setup and presents an analysis of the obtained results. Finally, Section VI concludes the paper.

II. Preliminaries

A. I-vector

The i-vector can be viewed as comprising the factors that explain the total variability in the Gaussian mixture model (GMM) supervector [22] from an utterance. In the i-vector framework, a speaker and session dependent supervector $\mathbf{M}$ from an utterance is represented by

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{x} \tag{1}$$

where $\mathbf{m}$ is a speaker and session-independent supervector, which is generally obtained from a universal background model (UBM) [23], $\mathbf{T}$ is a low-rank rectangular matrix that defines a total variability subspace, and $\mathbf{x}$ is a random vector that follows a standard normal distribution. The random vector $\mathbf{x}$ is composed of the factors that explain the total variability. It is thus called the i-vector.

To estimate the i-vector for a given utterance composed of a sequence of L acoustic feature frames $\{\mathbf{u}_l \in \mathbb{R}^D\}_{l=1}^{L}$, the zero and centralized first-order Baum–Welch statistics (i.e., z_c and $\mathbf{f}_c$, respectively) on a GMM-UBM θ having C mixture components are computed as follows:

$$z_c = \sum_{l=1}^{L} P(c|\mathbf{u}_l;\theta) \tag{2}$$

$$\mathbf{f}_c = \sum_{l=1}^{L} P(c|\mathbf{u}_l;\theta)(\mathbf{u}_l - \boldsymbol{\mu}_{\theta_c}) \tag{3}$$

where c is the index of the UBM mixture component, $P(c|\mathbf{u}_l;\theta)$ is the posterior probability of component c for frame $\mathbf{u}_l$ on UBM θ, and $\boldsymbol{\mu}_{\theta_c}$ is the mean of the mixture component c. The i-vector is estimated as the following equation

$$\mathbf{x} = (\mathbf{I} + \mathbf{T}^T\boldsymbol{\Sigma}^{-1}\mathbf{Z}\mathbf{T})^{-1}\mathbf{T}^T\boldsymbol{\Sigma}^{-1}\mathbf{f} \tag{4}$$

where $\mathbf{I}$ is an identity matrix, $\boldsymbol{\Sigma}$ is a $CD \times CD$ dimensional diagonal covariance matrix that models the residual variability not captured by $\mathbf{T}$, $\mathbf{Z}$ is a $CD \times CD$ dimensional block diagonal matrix whose diagonal blocks are $z_c\mathbf{I}$, and $\mathbf{f}$ is a CD dimensional supervector that is composed by concatenating all $\mathbf{f}_c$. A more detailed explanation for the computation can be found in [24].

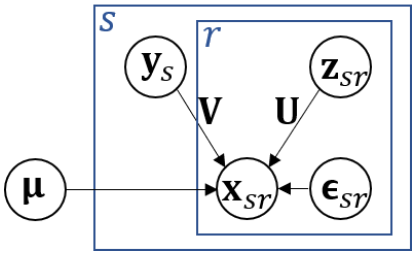

Fig. 1. The structure of PLDA.

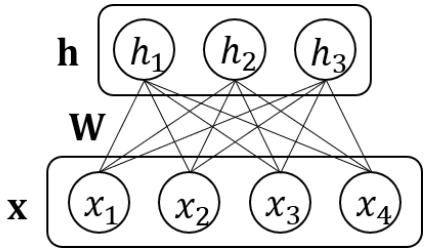

Fig. 2. The structure of RBM with $M = 4$ and $N = 3$ (all biases are omitted).

B. LDA

LDA is a renowned method in the field of machine learning for dimensionality reduction. Reducing dimensionality is performed by projecting into the lower-dimensional subspace that maximizes between-class variability and minimizes within-class variability. To find the subspace, we maximize the objective function J defined by

$$J(\mathbf{v}) = \frac{\mathbf{v}^T \mathbf{S_b} \mathbf{v}}{\mathbf{v}^T \mathbf{S_w} \mathbf{v}} \tag{5}$$

where $\mathbf{v}$ is the direction of the subspace, $\mathbf{S_b}$ is the between-class covariance matrix, and $\mathbf{S_w}$ is the within-class covariance matrix.

Maximizing (5) is equivalent to solving the eigenvalue equation defined by

$$\mathbf{S_w}^{-1} \mathbf{S_b} \mathbf{v} = \lambda \mathbf{v} \tag{6}$$

and $\mathbf{S_b}$ and $\mathbf{S_w}$ are computed as follows:

$$\mathbf{S_b} = \sum_{s=1}^{S} (\boldsymbol{\mu}_s - \boldsymbol{\mu})(\boldsymbol{\mu}_s - \boldsymbol{\mu})^T \tag{7}$$

$$\mathbf{S_w} = \sum_{s=1}^{S} \frac{1}{n_s} \sum_{r=1}^{n_s} (\mathbf{x}_{sr} - \boldsymbol{\mu}_s)(\mathbf{x}_{sr} - \boldsymbol{\mu}_s)^T \tag{8}$$

where S is the number of speakers in the training dataset, $\boldsymbol{\mu}_s$ is the mean of the i-vectors for speaker s, $\boldsymbol{\mu}$ is the global mean of i-vectors for the training dataset, n_s is the number of data points for speaker s, and $\mathbf{x}_{sr}$ is the r-th i-vector for speaker s.

The LDA projection matrix $\mathbf{A}$ is composed of a set of eigenvectors corresponding to the top-K largest eigenvalues, $\mathbf{A} = [\mathbf{v}_1 \vdots \cdots \vdots \mathbf{v}_K]$. Given an i-vector, $\mathbf{x}$, transformed feature $\mathbf{y}$ can be obtained by $\mathbf{y} = \mathbf{A}^T \mathbf{x}$.

C. PLDA

PLDA is a method that probabilistically models the between-class variability and the within-class variability using latent variables. Hence, it is a probabilistic approach to LDA. The PLDA model is a generative model and can be viewed as a directed graphical model, as depicted in Figure 1.

In PLDA, the r-th i-vector for speaker s, $\mathbf{x}_{sr}$, can be decomposed as

$$\mathbf{x}_{sr} = \boldsymbol{\mu} + \mathbf{V}\mathbf{y}_s + \mathbf{U}\mathbf{z}_{sr} + \boldsymbol{\epsilon}_{sr} \tag{9}$$

where $\boldsymbol{\mu}$ is the global mean of the i-vectors in the training dataset, $\mathbf{V}$, $\mathbf{U}$ are the matrices that define the between- and within-class variability subspaces, respectively, $\mathbf{y}_s$ and $\mathbf{z}_{sr}$ are latent vectors that explain the between- and within-class variability, respectively, and $\boldsymbol{\epsilon}_{sr}$ is the residual term. In Gaussian PLDA (GPLDA), both $\mathbf{y}_s$ and $\mathbf{z}_{sr}$ follow a standard normal distribution, and $\boldsymbol{\epsilon}_{sr}$ is assumed to follow a normal distribution with a zero mean and diagonal covariance, $\boldsymbol{\Sigma}_\epsilon$. All latent variables are assumed to be statistically independent.

Given a pair of an enrollment i-vector $\mathbf{x}_e$ and a test i-vector $\mathbf{x}_t$ for a speaker verification trial, the likelihood ratio score for GPLDA can be computed as

$$\frac{P(\mathbf{x}_e, \mathbf{x}_t | H_1)}{P(\mathbf{x}_e | H_0) P(\mathbf{x}_t | H_0)} = \frac{\mathcal{N}\left(\begin{bmatrix}\mathbf{x}_e \\ \mathbf{x}_t\end{bmatrix}; \begin{bmatrix}\boldsymbol{\mu} \\ \boldsymbol{\mu}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_\mathbf{x} & \boldsymbol{\Sigma}_\mathbf{y} \\ \boldsymbol{\Sigma}_\mathbf{y} & \boldsymbol{\Sigma}_\mathbf{x}\end{bmatrix}\right)}{\mathcal{N}\left(\begin{bmatrix}\mathbf{x}_e \\ \mathbf{x}_t\end{bmatrix}; \begin{bmatrix}\boldsymbol{\mu} \\ \boldsymbol{\mu}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_\mathbf{x} & 0 \\ 0 & \boldsymbol{\Sigma}_\mathbf{x}\end{bmatrix}\right)} \tag{10}$$

where H_0 is the hypothesis that $\mathbf{x}_e$ and $\mathbf{x}_t$ come from different speakers, H_1 is the hypothesis that $\mathbf{x}_e$ and $\mathbf{x}_t$ come from the same speaker, and $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the (multivariate) Gaussian probability density function with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}_\mathbf{x} = \mathbf{V}\mathbf{V}^T + \mathbf{U}\mathbf{U}^T + \boldsymbol{\Sigma}_\epsilon$, and $\boldsymbol{\Sigma}_\mathbf{y} = \mathbf{V}\mathbf{V}^T$.

D. RBM

RBM is an energy-based generative neural network, that consists of one layer of visible units $\mathbf{x} = \{x_i\}_{i=1}^M$ and one layer of hidden units $\mathbf{h} = \{h_j\}_{j=1}^N$. It can be expressed as a bipartite undirected graphical model [25], as depicted in Figure 2. It is a special case of the Boltzmann machine with the restriction that there are only symmetric between-layer connections and no intralayer connections. It consists of network parameters $\Theta = \{\mathbf{W}, \mathbf{a}, \mathbf{b}\}$, where $\mathbf{W} = (W_{ij}) \in \mathbb{R}^{M \times N}$ represents the connection weights between $\mathbf{x}$ and $\mathbf{h}$, and $\mathbf{a} = \{a_i\}_{i=1}^M$ and $\mathbf{b} = \{b_j\}_{j=1}^N$ are biases for $\mathbf{x}$ and $\mathbf{h}$, respectively. Note that $\mathbf{x}$ corresponds to observed data (e.g., i-vector), and $\mathbf{h}$ corresponds to latent variables. An original RBM assumes that the values of both $\mathbf{x}$ and $\mathbf{h}$ are binary. In this paper, however, real-valued $\mathbf{x}$ and $\mathbf{h}$ are applied to the Gaussian–Gaussian RBM (GGRBM) [26], [27] because the GGRBM assumes real values in both $\mathbf{x}$ and $\mathbf{h}$.

RBM can model a probability distribution over $\mathbf{x}$ using $\mathbf{h}$. The probability distribution is defined through an energy function. For the GGRBM, the energy function is

$$E(\mathbf{x}, \mathbf{h}; \Theta) = \frac{\|\mathbf{x} - \mathbf{a}\|^2}{2} + \frac{\|\mathbf{h} - \mathbf{b}\|^2}{2} - \mathbf{x}^T \mathbf{W} \mathbf{h} \tag{11}$$

where $\|\cdot\|$ denotes the Euclidean norm. Note that the standard deviations for $\mathbf{x}$ and $\mathbf{h}$ are set to one and are omitted from our notations without loss of generality (see the original form in Appendix A). Since an enormous computational load is required to train RBM, a simplified training algorithm called contrastive divergence (CD) [28] is used. The practical details for the training process are introduced in [29].

The joint probability distribution of $\mathbf{x}$ and $\mathbf{h}$ is defined by

$$P(\mathbf{x},\mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{x},\mathbf{h})} \tag{12}$$

where normalizing constant $Z = \iint e^{-E(\mathbf{x},\mathbf{h})}\, d\mathbf{x}\, d\mathbf{h}$ is defined by integrating over all possible pairs of $\mathbf{x}$ and $\mathbf{h}$. A marginal probability of $\mathbf{x}$ can be obtained by marginalizing $P(\mathbf{x},\mathbf{h})$ over all possible values of $\mathbf{h}$, as follows:

$$P(\mathbf{x}) = \frac{1}{Z} \int e^{-E(\mathbf{x},\mathbf{h})}\, d\mathbf{h}. \tag{13}$$

RBM has no intralayer connections; therefore, the units of $\mathbf{x}$ are conditionally independent given $\mathbf{h}$, and vice versa. For the GGRBM, the conditional probabilities are

$$P(\mathbf{h}|\mathbf{x}) = \prod_{j}^{N} P(h_j|\mathbf{x}) = \mathcal{N}(\mathbf{h}; \mathbf{b} + \mathbf{W}^T\mathbf{x}, \mathbf{I}) \tag{14}$$

$$P(\mathbf{x}|\mathbf{h}) = \prod_{i}^{M} P(x_i|\mathbf{h}) = \mathcal{N}(\mathbf{x}; \mathbf{a} + \mathbf{W}\mathbf{h}, \mathbf{I}) \tag{15}$$

where $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the (multivariate) Gaussian probability density function with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and $\mathbf{I}$ is the identity matrix.

The RBM is trained to minimize the negative loglikelihood using stochastic gradient-based optimization methods. Given data $\mathbf{x}$, the gradient of the negative loglikelihood is

$$\begin{aligned} &-\frac{\partial \log P(\mathbf{x})}{\partial \theta} \\ &= \int p(\mathbf{h}|\mathbf{x}) \frac{\partial E(\mathbf{x},\mathbf{h})}{\partial \theta} d\mathbf{h} \\ &- \iint p(\mathbf{x}^*,\mathbf{h}) \frac{\partial E(\mathbf{x}^*,\mathbf{h})}{\partial \theta} d\mathbf{x}^*\, d\mathbf{h}. \end{aligned} \tag{16}$$

The first term of RHS of (16) is estimated using the given data. However, it is difficult to obtain an unbiased sample that is used to estimate the second term of the RHS of (16). One step of CD (denoted by CD-1) is generally used to approximate the sample. CD-1 starts with the given data $\mathbf{x}^{(0)}$ and performs the following procedures:

$$\mathbf{x}^{(0)} \xrightarrow{P(\mathbf{h}|\mathbf{x}^{(0)})} \tilde{\mathbf{h}}^{(0)} \xrightarrow{P(\mathbf{x}|\tilde{\mathbf{h}}^{(0)})} \mathbf{x}^{(1)} \xrightarrow{P(\mathbf{h}|\mathbf{x}^{(1)})} \mathbf{h}^{(1)} \tag{17}$$

where the notation $\boldsymbol{\alpha} \xrightarrow{P(\boldsymbol{\beta}|\boldsymbol{\alpha})} \tilde{\boldsymbol{\beta}}$ means that $\tilde{\boldsymbol{\beta}}$ is the sample obtained from the conditional probability $P(\boldsymbol{\beta}|\boldsymbol{\alpha})$, the notation $\boldsymbol{\alpha} \xrightarrow{P(\boldsymbol{\beta}|\boldsymbol{\alpha})} \boldsymbol{\beta}$ means that $\boldsymbol{\beta}$ is the expectation of $P(\boldsymbol{\beta}|\boldsymbol{\alpha})$, and the superscript $\cdot^{(n)}$ denotes the index of the sampling step. It is worth noting that we do not sample $\tilde{\mathbf{x}}^{(1)}$ and $\tilde{\mathbf{h}}^{(1)}$ from $P(\mathbf{x}|\tilde{\mathbf{h}}^{(0)})$ and $P(\mathbf{h}|\mathbf{x}^{(1)})$; rather, we use the expectations of those $\mathbf{x}^{(1)}$ and $\mathbf{h}^{(1)}$, as discussed in [29]. The approximation of (16) with respect to $\mathbf{W}$ is

$$\begin{aligned} -\frac{\partial \log P(\mathbf{x}^{(0)})}{\partial \mathbf{W}} &\approx \frac{\partial E(\mathbf{x}^{(0)},\mathbf{h}^{(0)})}{\partial \mathbf{W}} - \frac{\partial E(\mathbf{x}^{(1)},\mathbf{h}^{(1)})}{\partial \mathbf{W}} \\ &= \mathbf{x}^{(1)}\mathbf{h}^{(1)^T} \\ &\quad - \mathbf{x}^{(0)}\mathbf{h}^{(0)^T}. \end{aligned} \tag{18}$$

Note that the expectations are used in (18), instead of the samples.

E. Fuzzy RBM

We consider fuzzy numbers with triangular membership functions (see Figure 3). The membership function for triangular fuzzy number $\bar{\theta}$ is

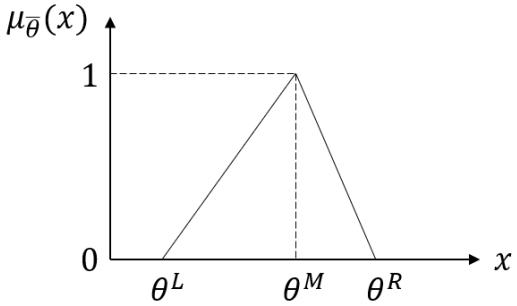

Fig. 3. The example of membership function of triangular fuzzy number.

$$\mu_{\bar{\theta}}(x) = \begin{cases} \dfrac{x-\theta^L}{\theta^M-\theta^L} & ,\theta^L \le x < \theta^M \\ \dfrac{x-\theta^R}{\theta^M-\theta^R} & ,\theta^M \le x \le \theta^R \end{cases} \tag{19}$$

where θ_L and θ_R are the left and right bounds, respectively, and θ_M is the center of the fuzzy number, $\bar{\theta}$. For the symmetric triangular fuzzy number (STFN), $\theta^M = (\theta^L + \theta^R)/2$.

The FRBM extends the RBM by replacing all the crisp parameters with fuzzy numbers. The fuzzy energy function for a Gaussian–Gaussian FRBM (GGFRBM) is extended from (11), as follows:

$$\bar{E}(\mathbf{x},\mathbf{h};\bar{\Theta}) = \frac{\|\mathbf{x}-\bar{\mathbf{a}}\|^2}{2} + \frac{\|\mathbf{h}-\bar{\mathbf{b}}\|^2}{2} - \mathbf{x}^T\bar{\mathbf{W}}\mathbf{h} \tag{20}$$

where $\bar{\Theta} = \{\bar{\mathbf{W}}, \bar{\mathbf{a}}, \bar{\mathbf{b}}\}$ is the set of fuzzy parameters. However, (20) cannot be directly used to define the probability distribution because the probability defined directly by (20) is a complex nonlinear function, which is difficult to optimize. To train the FRBM, (20) must be defuzzified. After defuzzification, we can treat the fuzzy energy as the weighted sum of the energy defined by the parameters of each bound, as shown further below. All the probabilities are defined through the defuzzified free energy.

For FRBM based on STFN (FRBM-STFN) [12], [13], (20) is defuzzified as follows:

$$\bar{E}(\mathbf{x},\mathbf{h};\bar{\Theta}) \approx \frac{E(\mathbf{x},\mathbf{h};\Theta^L)+E(\mathbf{x},\mathbf{h};\Theta^R)}{2} \tag{21}$$

where $\Theta^L=\{\mathbf{W}^L,\mathbf{a}^L,\mathbf{b}^L\}$ and $\Theta^R=\{\mathbf{W}^R,\mathbf{a}^R,\mathbf{b}^R\}$ are the left and right bounds of $\bar{\Theta}$, respectively. With STFN, the center of $\bar{\Theta}$ is not taken as comprising the parameters because it can be determined as $(\Theta^L+\Theta^R)/2$. The conditional probabilities for the FRBM-STFN are

$$\begin{aligned} P^L(\mathbf{h}|\mathbf{x}) &= \mathcal{N}(\mathbf{h};\mathbf{b}^L+\mathbf{W}^{L^T}\mathbf{x},\mathbf{I}) \\ P^R(\mathbf{h}|\mathbf{x}) &= \mathcal{N}(\mathbf{h};\mathbf{b}^R+\mathbf{W}^{R^T}\mathbf{x},\mathbf{I}) \end{aligned} \tag{22}$$

$$\begin{aligned} P^L(\mathbf{x}|\mathbf{h}) &= \mathcal{N}(\mathbf{x};\mathbf{a}^L+\mathbf{W}^L\mathbf{h},\mathbf{I}) \\ P^R(\mathbf{x}|\mathbf{h}) &= \mathcal{N}(\mathbf{x};\mathbf{a}^R+\mathbf{W}^R\mathbf{h},\mathbf{I}), \end{aligned} \tag{23}$$

which are extended from (14) and (15) with respect to each bound. The sampling procedure, which is based on CD-1, is as follows:

$$\begin{aligned} \mathbf{x}^{(0)} &\xrightarrow{P^L(\mathbf{h}|\mathbf{x}^{(0)})} \tilde{\mathbf{h}}^{L(0)} \xrightarrow{P^L(\mathbf{x}|\tilde{\mathbf{h}}^{L(0)})} \mathbf{x}^{L(1)} \xrightarrow{P^L(\mathbf{h}|\mathbf{x}^{L(1)})} \mathbf{h}^{L(1)} \\ &\xrightarrow{P^R(\mathbf{h}|\mathbf{x}^{(0)})} \tilde{\mathbf{h}}^{R(0)} \xrightarrow{P^R(\mathbf{x}|\tilde{\mathbf{h}}^{R(0)})} \mathbf{x}^{R(1)} \xrightarrow{P^R(\mathbf{h}|\mathbf{x}^{R(1)})} \mathbf{h}^{R(1)}. \end{aligned} \tag{24}$$

The approximated gradients of the log probability with respect to $\mathbf{W}^L$ and $\mathbf{W}^R$ are

$$\begin{aligned} -\frac{\partial \log P(\mathbf{x}^{(0)})}{\partial \mathbf{W}^L} &\approx \frac{1}{2}\left(\mathbf{x}^{L(1)}\mathbf{h}^{L(1)^T}-\mathbf{x}^{(0)}\mathbf{h}^{L(0)^T}\right) \\ -\frac{\partial \log P(\mathbf{x}^{(0)})}{\partial \mathbf{W}^R} &\approx \frac{1}{2}\left(\mathbf{x}^{R(1)}\mathbf{h}^{R(1)^T}-\mathbf{x}^{(0)}\mathbf{h}^{R(0)^T}\right). \end{aligned} \tag{25}$$

For FRBM based on the asymmetric triangular fuzzy number (FRBM-ATFN) [13], (20) is defuzzified as follows:

$$\begin{aligned} &\bar{E}(\mathbf{x},\mathbf{h};\bar{\Theta}) \\ &\approx \frac{E(\mathbf{x},\mathbf{h};\Theta^L)+4E(\mathbf{x},\mathbf{h};\Theta^M)+E(\mathbf{x},\mathbf{h};\Theta^R)}{6} \end{aligned} \tag{26}$$

where $\Theta^M=\{\mathbf{W}^M,\mathbf{a}^M,\mathbf{b}^M\}$ comprise the center of $\bar{\Theta}$. With ATFN, the additional parameters in Θ^M must also be trained. The conditional probabilities for the FRBM-ATFN are (22), (23), and additionally:

$$P^M(\mathbf{h}|\mathbf{x}) = \mathcal{N}\left(\mathbf{h};\mathbf{b}^M+\mathbf{W}^{M^T}\mathbf{x},\mathbf{I}\right) \tag{27}$$

$$P^M(\mathbf{x}|\mathbf{h}) = \mathcal{N}(\mathbf{x};\mathbf{a}^M+\mathbf{W}^M\mathbf{h},\mathbf{I}), \tag{28}$$

which are also extended from (14) and (15). The sampling procedure is as follows:

$$\begin{aligned} &\xrightarrow{P^L(\mathbf{h}|\mathbf{x}^{(0)})} \tilde{\mathbf{h}}^{L(0)} \xrightarrow{P^L(\mathbf{x}|\tilde{\mathbf{h}}^{L(0)})} \mathbf{x}^{L(1)} \xrightarrow{P^L(\mathbf{h}|\mathbf{x}^{L(1)})} \mathbf{h}^{L(1)} \\ \mathbf{x}^{(0)} &\xrightarrow{P^M(\mathbf{h}|\mathbf{x}^{(0)})} \tilde{\mathbf{h}}^{M(0)} \xrightarrow{P^M(\mathbf{x}|\tilde{\mathbf{h}}^{M(0)})} \mathbf{x}^{M(1)} \xrightarrow{P^M(\mathbf{h}|\mathbf{x}^{M(1)})} \mathbf{h}^{M(1)} \\ &\xrightarrow{P^R(\mathbf{h}|\mathbf{x}^{(0)})} \tilde{\mathbf{h}}^{R(0)} \xrightarrow{P^R(\mathbf{x}|\tilde{\mathbf{h}}^{R(0)})} \mathbf{x}^{R(1)} \xrightarrow{P^R(\mathbf{h}|\mathbf{x}^{R(1)})} \mathbf{h}^{R(1)}. \end{aligned} \tag{29}$$

The approximated gradients of the log probability with respect to $\mathbf{W}^L$, $\mathbf{W}^M$ and $\mathbf{W}^R$ are

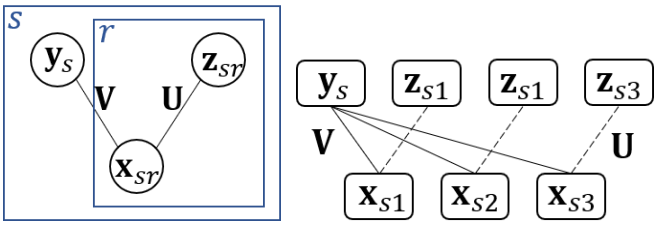

Fig. 4. (Left) The structure of RBM-PLDA. (Right) An example of the structure of RBM-PLDA with $n_s=3$. The solid and dashed lines represent the connection weights **V** and **U** respectively.

$$\begin{aligned} -\frac{\partial \log P(\mathbf{x}^{(0)})}{\partial \mathbf{W}^L} &\approx \frac{1}{6}\left(\mathbf{x}^{L(1)}\mathbf{h}^{L(1)^T}-\mathbf{x}^{(0)}\mathbf{h}^{L(0)^T}\right) \\ -\frac{\partial \log P(\mathbf{x}^{(0)})}{\partial \mathbf{W}^M} &\approx \frac{2}{3}\left(\mathbf{x}^{M(1)}\mathbf{h}^{M(1)^T}-\mathbf{x}^{(0)}\mathbf{h}^{M(0)^T}\right) \\ -\frac{\partial \log P(\mathbf{x}^{(0)})}{\partial \mathbf{W}^R} &\approx \frac{1}{6}\left(\mathbf{x}^{R(1)}\mathbf{h}^{R(1)^T}-\mathbf{x}^{(0)}\mathbf{h}^{R(0)^T}\right). \end{aligned} \tag{30}$$

As a result, the defuzzified FRBM can be viewed as the RBM with the parameters considering bounds for uncertainties. Note that we apply the gradients differently for the parameters of each bound, which is the same as in [12] but different from [13]. The reason is explained later. Details for the defuzzification process are shown in [12], [13].

F. RBM-PLDA

RBM-based PLDA (RBM-PLDA) is a kind of GGRBM applied to GPLDA, as depicted in Figure 4. The hidden units $\mathbf{h}$ are divided into the speaker factors $\mathbf{y}=\{y_j\}_{j=1}^{N_y}$ and session factors $\mathbf{z}=\{z_k\}_{k=1}^{N_z}$; that is, $\mathbf{h}=[\mathbf{y}^T,\mathbf{z}^T]^T$, and the weights are divided accordingly, $\mathbf{W}=[\mathbf{V}\vdots\mathbf{U}]$, where $\mathbf{V}=(V_{ij})\in\mathbb{R}^{M\times N_y}$ is the connection weight between $\mathbf{x}$ and $\mathbf{y}$, and $\mathbf{U}=(U_{ik})\in\mathbb{R}^{M\times N_z}$ is the connection weight between $\mathbf{x}$ and $\mathbf{z}$. The RBM-PLDA consists of network parameters $\Phi=\{\mathbf{V},\mathbf{U}\}$. Since it assumes that $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ follow a standard normal distribution, all the biases and standard deviations for $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are set to zero and one respectively, and omitted from our notations.

Similar to RBM, RBM-PLDA is trained using stochastic-gradient-based optimization methods. One important difference is that each minibatch takes all i-vectors for one speaker in the training dataset, rather than a random minibatch from the entire training dataset. In the training phase, $\mathbf{y}$ is estimated per minibatch. In contrast, $\mathbf{z}$ is estimated per i-vector. Denote $\mathbf{x}_s=\{\mathbf{x}_{sr}\}_{r=1}^{n_s}$ as the whitened i-vectors for speaker s, $\mathbf{y}_s$ as the speaker factor for speaker s, and $\mathbf{z}_{sr}$ as the session factors for $\mathbf{x}_{sr}$. Given $\mathbf{x}_s^{(0)}=\{\mathbf{x}_{sr}^{(0)}\}_{r=1}^{n_s}$ as a minibatch, the sampling procedure based on CD-1 is as follows:

$$\begin{aligned} \boldsymbol{\mu}_s^{(0)} &\xrightarrow{P(\mathbf{y}_s|\boldsymbol{\mu}_s^{(0)})} \tilde{\mathbf{y}}_s^{(0)} \\ \mathbf{x}_{sr}^{(0)} &\xRightarrow{P(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{(0)})} \tilde{\mathbf{z}}_{sr}^{(0)} \end{aligned} \xRightarrow{P(\mathbf{x}_{sr}|\tilde{\mathbf{y}}_s^{(0)},\tilde{\mathbf{z}}_{sr}^{(0)})} \begin{aligned} \boldsymbol{\mu}_s^{(1)} &\xrightarrow{P(\mathbf{y}_s|\boldsymbol{\mu}_s^{(1)})} \mathbf{y}_s^{(1)} \\ \mathbf{x}_{sr}^{(1)} &\xRightarrow{P(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{(1)})} \mathbf{z}_{sr}^{(1)} \end{aligned} \tag{31}$$

where $\boldsymbol{\mu}_s^{(n)}=n_s^{-1}\sum_{r=1}^{n_s}\mathbf{x}_{sr}^{(n)}$ is the mean of $\mathbf{x}_s^{(n)}$, and a single and double arrow means that the corresponding procedure is performed one time and multiple times for $r=1,\dots,n_s$ with a minibatch, respectively. The conditional probabilities used

in (31) are

$$P(\mathbf{y}_s|\boldsymbol{\mu}_s) = \mathcal{N}(\mathbf{y}_s;\ \mathbf{V}^T\boldsymbol{\mu}_s, n_s^{-1}\mathbf{I}) \tag{32}$$

$$P(\mathbf{z}_{sr}|\mathbf{x}_{sr}) = \mathcal{N}(\mathbf{z}_{sr};\ \mathbf{U}^T\mathbf{x}_{sr}, \mathbf{I}) \tag{33}$$

$$P(\mathbf{x}_{sr}|\mathbf{y}_s, \mathbf{z}_{sr}) = \mathcal{N}(\mathbf{x}_{sr};\ \mathbf{V}\mathbf{y}_s + \mathbf{U}\mathbf{z}_{sr}, \mathbf{I}). \tag{34}$$

The gradients of the negative loglikelihood of $\mathbf{x}_s^{(0)}$, $\mathcal{L}(\mathbf{x}_s^{(0)}) = \sum_{r=1}^{n_s} \log P\left(\mathbf{x}_{sr}^{(0)}\right)$ with respect to $\mathbf{V}$ and $\mathbf{U}$ are approximated by

$$-\frac{\partial \mathcal{L}(\mathbf{x}_s^{(0)})}{\partial \mathbf{V}} \approx n_s\left(\boldsymbol{\mu}_s^{(1)}\mathbf{y}_s^{(1)^T} - \boldsymbol{\mu}_s^{(0)}\mathbf{y}_s^{(0)^T}\right) \tag{35}$$

$$-\frac{\partial \mathcal{L}(\mathbf{x}_s^{(0)})}{\partial \mathbf{U}} \approx \sum_{r=1}^{n_s}\left(\mathbf{x}_{sr}^{(1)}\mathbf{z}_{sr}^{(1)^T} - \mathbf{x}_{sr}^{(0)}\mathbf{z}_{sr}^{(0)^T}\right). \tag{36}$$

In the evaluation phase, RBM-PLDA is used to reduce the dimensionality, similar to LDA, using only $\mathbf{V}$ as the projection matrix. Given an i-vector, $\mathbf{x}$, transformed feature $\mathbf{y}$ can be obtained by $\mathbf{y} = \mathbf{V}^T\mathbf{x}$.

III. Fuzzy RBM-based PLDA

We propose the GGFRBM-based PLDA (simply denoted by FRBM-PLDA), which is an extension of RBM-PLDA. The proposed FRBM-PLDA applies fuzzy theory to all the crisp parameters of $\Phi = \{\mathbf{V}, \mathbf{U}\}$ so that the uncertainties of the session variability can be incorporated by fuzzy parameters of $\bar{\Phi} = \{\bar{\mathbf{V}}, \bar{\mathbf{U}}\}$. As in FRBM, the fuzzy numbers of FRBM-PLDA, $\bar{\Phi}$, must be defuzzified. After defuzzification, FRBM-PLDA can be viewed as RBM-PLDA with the parameters considering the bounds of uncertainties. As mentioned in Section II-*E*, we apply the gradients differently for the parameters of each bound. Although it does not guarantee that the left bound is always smaller than the right bound, we observe that applying the gradients equally shows a larger mean squared error (MSE) than applying the gradients differently (see Appendix B).

A. FRBM-PLDA with STFN

For FRBM-PLDA with STFN, the parameters are $\Phi^L = \{\mathbf{V}^L, \mathbf{U}^L\}$ and $\Phi^R = \{\mathbf{V}^R, \mathbf{U}^R\}$. Φ^L and Φ^R are initialized as negative and positive small random numbers, respectively. The conditional probabilities are extended from (32), (33), and (34) with respect to each bound:

$$\begin{aligned} P^L(\mathbf{y}_s|\boldsymbol{\mu}_s) &= \mathcal{N}(\mathbf{y}_s;\ \mathbf{V}^{L^T}\boldsymbol{\mu}_s, n_s^{-1}\mathbf{I}) \\ P^R(\mathbf{y}_s|\boldsymbol{\mu}_s) &= \mathcal{N}(\mathbf{y}_s;\ \mathbf{V}^{R^T}\boldsymbol{\mu}_s, n_s^{-1}\mathbf{I}) \end{aligned} \tag{37}$$

$$\begin{aligned} P^L(\mathbf{z}_{sr}|\mathbf{x}_{sr}) &= \mathcal{N}(\mathbf{z}_{sr};\ \mathbf{U}^{L^T}\mathbf{x}_{sr}, \mathbf{I}) \\ P^R(\mathbf{z}_{sr}|\mathbf{x}_{sr}) &= \mathcal{N}(\mathbf{z}_{sr};\ \mathbf{U}^{R^T}\mathbf{x}_{sr}, \mathbf{I}) \end{aligned} \tag{38}$$

$$\begin{aligned} P^L(\mathbf{x}_{sr}|\mathbf{y}_s, \mathbf{z}_{sr}) &= \mathcal{N}(\mathbf{x}_{sr};\ \mathbf{V}^L\mathbf{y}_s + \mathbf{U}^L\mathbf{z}_{sr}, \mathbf{I}) \\ P^R(\mathbf{x}_{sr}|\mathbf{y}_s, \mathbf{z}_{sr}) &= \mathcal{N}(\mathbf{x}_{sr};\ \mathbf{V}^R\mathbf{y}_s + \mathbf{U}^R\mathbf{z}_{sr}, \mathbf{I}). \end{aligned} \tag{39}$$

In the training phase, $\mathbf{x}_s^{(0)}$ is given in a minibatch. The sampling procedure is as follows:

Algorithm 1 Update FRBM-PLDA with STFN

Input
$\mathbf{x}_s^{(0)} = \{\mathbf{x}_{sr}^{(0)}\}_{r=1}^{n_s}$: the prewhitened i-vectors for speaker s
$\mathbf{V}^L$, $\mathbf{V}^R$: Left and right bound of the weights between $\mathbf{x}$ and $\mathbf{y}$
$\mathbf{U}^L$, $\mathbf{U}^R$: Left and right bound of the weights between $\mathbf{x}$ and $\mathbf{z}$

1. compute $\boldsymbol{\mu}_s^{(0)} = n_s^{-1}\sum_{r=1}^{n_s}\mathbf{x}_{sr}^{(0)}$
2. compute $\mathbf{y}_s^{L(0)} = \mathbf{V}^{L^T}\boldsymbol{\mu}_s^{(0)}$ and $\mathbf{y}_s^{R(0)} = \mathbf{V}^{R^T}\boldsymbol{\mu}_s^{(0)}$
3. sample $\tilde{\mathbf{y}}_s^{L(0)} \sim P^L(\mathbf{y}_s|\boldsymbol{\mu}_s^{(0)})$ and $\tilde{\mathbf{y}}_s^{R(0)} \sim P^R(\mathbf{y}_s|\boldsymbol{\mu}_s^{(0)})$
4. **for all** $r = 1, \ldots, n_s$ **do**
5. compute $\mathbf{z}_{sr}^{L(0)} = \mathbf{U}^{L^T}\mathbf{x}_{sr}^{(0)}$ and $\mathbf{z}_{sr}^{R(0)} = \mathbf{U}^{R^T}\mathbf{x}_{sr}^{(0)}$
6. sample $\tilde{\mathbf{z}}_{sr}^{L(0)} \sim P^L(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{(0)})$ and $\tilde{\mathbf{z}}_{sr}^{R(0)} \sim P^R(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{(0)})$
7. compute $\mathbf{x}_{sr}^{L(1)} = \mathbf{V}^L\tilde{\mathbf{y}}_s^{L(0)} + \mathbf{U}^L\tilde{\mathbf{z}}_{sr}^{L(0)}$ and $\mathbf{x}_{sr}^{R(1)} = \mathbf{V}^R\tilde{\mathbf{y}}_s^{R(0)} + \mathbf{U}^R\tilde{\mathbf{z}}_{sr}^{R(0)}$
8. **end for**
9. compute $\boldsymbol{\mu}_s^{L(1)} = n_s^{-1}\sum_{r=1}^{n_s}\mathbf{x}_{sr}^{L(1)}$ and $\boldsymbol{\mu}_s^{R(1)} = n_s^{-1}\sum_{r=1}^{n_s}\mathbf{x}_{sr}^{R(1)}$
10. compute $\mathbf{y}_s^{L(1)} = \mathbf{V}^{L^T}\boldsymbol{\mu}_s^{L(1)}$ and $\mathbf{y}_s^{R(1)} = \mathbf{V}^{R^T}\boldsymbol{\mu}_s^{R(1)}$
11. **for all** $r = 1, \ldots, n_s$ **do**
12. compute $\mathbf{z}_{sr}^{L(1)} = \mathbf{U}^{L^T}\mathbf{x}_{sr}^{L(1)}$ and $\mathbf{z}_{sr}^{R(1)} = \mathbf{U}^{R^T}\mathbf{x}_{sr}^{R(1)}$
13. **end for**
14. $\Delta\mathbf{V}_L = \frac{n_s}{2}\left(\boldsymbol{\mu}_s^{L(1)}\mathbf{y}_s^{L(1)^T} - \boldsymbol{\mu}_s^{(0)}\mathbf{y}_s^{L(0)^T}\right)$
15. $\Delta\mathbf{V}_R = \frac{n_s}{2}\left(\boldsymbol{\mu}_s^{R(1)}\mathbf{y}_s^{R(1)^T} - \boldsymbol{\mu}_s^{(0)}\mathbf{y}_s^{R(0)^T}\right)$
16. $\Delta\mathbf{U}_L = \frac{1}{2}\sum_{r=1}^{n_s}\left(\mathbf{x}_{sr}^{L(1)}\mathbf{z}_{sr}^{L(1)^T} - \mathbf{x}_{sr}^{(0)}\mathbf{z}_{sr}^{L(0)^T}\right)$
17. $\Delta\mathbf{U}_R = \frac{1}{2}\sum_{r=1}^{n_s}\left(\mathbf{x}_{sr}^{R(1)}\mathbf{z}_{sr}^{R(1)^T} - \mathbf{x}_{sr}^{(0)}\mathbf{z}_{sr}^{R(0)^T}\right)$
18. update $\mathbf{V}_L$, $\mathbf{V}_R$, $\mathbf{U}_L$, $\mathbf{U}_R$ using $\Delta\mathbf{V}_L$, $\Delta\mathbf{V}_R$, $\Delta\mathbf{U}_L$, $\Delta\mathbf{U}_R$

$$\begin{array}{llll} \xrightarrow{P^L(\mathbf{y}_s|\boldsymbol{\mu}_s^{(0)})} \tilde{\mathbf{y}}_s^{L(0)} & \xRightarrow{P^L(\mathbf{x}_{sr}|\tilde{\mathbf{y}}_s^{L(0)}, \tilde{\mathbf{z}}_{sr}^{L(0)})} & \boldsymbol{\mu}_s^{L(1)} \xrightarrow{P^L(\mathbf{y}_s|\boldsymbol{\mu}_s^{L(1)})} \mathbf{y}_s^{L(1)} & \\ \xRightarrow{P^L(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{(0)})} \tilde{\mathbf{z}}_{sr}^{L(0)} & & \mathbf{x}_{sr}^{L(1)} \xRightarrow{P^L(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{L(1)})} \mathbf{z}_{sr}^{L(1)} & \\ \xrightarrow{P^R(\mathbf{y}_s|\boldsymbol{\mu}_s^{(0)})} \tilde{\mathbf{y}}_s^{R(0)} & \xRightarrow{P^R(\mathbf{x}_{sr}|\tilde{\mathbf{y}}_s^{R(0)}, \tilde{\mathbf{z}}_{sr}^{R(0)})} & \boldsymbol{\mu}_s^{R(1)} \xrightarrow{P^R(\mathbf{y}_s|\boldsymbol{\mu}_s^{R(1)})} \mathbf{y}_s^{R(1)} & \\ \xRightarrow{P^R(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{(0)})} \tilde{\mathbf{z}}_{sr}^{R(0)} & & \mathbf{x}_{sr}^{R(1)} \xRightarrow{P^R(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{R(1)})} \mathbf{z}_{sr}^{R(1)}. & \end{array} \tag{40}$$

The gradients of the negative log likelihood of $\mathbf{x}_s^{(0)}$ with respect to each parameter are approximated by

$$\begin{aligned} -\frac{\partial \mathcal{L}(\mathbf{x}_s^{(0)})}{\partial \mathbf{V}^L} &\approx \frac{n_s}{2}\left(\boldsymbol{\mu}_s^{L(1)}\mathbf{y}_s^{L(1)^T} - \boldsymbol{\mu}_s^{(0)}\mathbf{y}_s^{L(0)^T}\right) \\ -\frac{\partial \mathcal{L}(\mathbf{x}_s^{(0)})}{\partial \mathbf{V}^R} &\approx \frac{n_s}{2}\left(\boldsymbol{\mu}_s^{R(1)}\mathbf{y}_s^{R(1)^T} - \boldsymbol{\mu}_s^{(0)}\mathbf{y}_s^{R(0)^T}\right) \end{aligned} \tag{41}$$

$$\begin{aligned} -\frac{\partial \mathcal{L}(\mathbf{x}_s^{(0)})}{\partial \mathbf{U}^L} &\approx \frac{1}{2}\sum_{r=1}^{n_s}\left(\mathbf{x}_{sr}^{L(1)}\mathbf{z}_{sr}^{L(1)^T} - \mathbf{x}_{sr}^{(0)}\mathbf{z}_{sr}^{L(0)^T}\right) \\ -\frac{\partial \mathcal{L}(\mathbf{x}_s^{(0)})}{\partial \mathbf{U}^R} &\approx \frac{1}{2}\sum_{r=1}^{n_s}\left(\mathbf{x}_{sr}^{R(1)}\mathbf{z}_{sr}^{R(1)^T} - \mathbf{x}_{sr}^{(0)}\mathbf{z}_{sr}^{R(0)^T}\right). \end{aligned} \tag{42}$$

The pseudocode for updating FRBM-PLDA with STFN is shown in Algorithm 1.

In the evaluation phase, $\mathbf{V}^L$ and $\mathbf{V}^R$ are used to reduce the dimensionality for a given i-vector $\mathbf{x}$. We can obtain two transformed features, $\mathbf{y}^L = \mathbf{V}^{L^T}\mathbf{x}$ and $\mathbf{y}^R = \mathbf{V}^{R^T}\mathbf{x}$.

B. FRBM-PLDA with ATFN

For FRBM-PLDA with ATFN, the parameters are Φ^L, Φ^R and the centers of $\bar{\Phi}$, $\Phi^M = \{\mathbf{V}^M, \mathbf{U}^M\}$. As in STFN, Φ^L and Φ^R are initialized as negative and positive small random numbers, respectively. In contrast, Φ^M is initialized as

follows:

$$\begin{aligned}\mathbf{V}^M &= \rho_1\mathbf{V}^L + (1-\rho_1)\mathbf{V}^R \\ \mathbf{U}^M &= \rho_2\mathbf{U}^L + (1-\rho_2)\mathbf{U}^R\end{aligned} \tag{43}$$

where ρ_1 and ρ_2 are random numbers in the interval (0, 1). The conditional probabilities are (37), (38), and (39). In addition:

$$P^M(\mathbf{y}_s|\boldsymbol{\mu}_s) = \mathcal{N}(\mathbf{y}_s;\ \mathbf{V}^{M^T}\boldsymbol{\mu}_s, n_s^{-1}\mathbf{I}) \tag{44}$$

$$P^M(\mathbf{z}_{sr}|\mathbf{x}_{sr}) = \mathcal{N}(\mathbf{z}_{sr};\ \mathbf{U}^{M^T}\mathbf{x}_{sr}, \mathbf{I}) \tag{45}$$

$$P^M(\mathbf{x}_{sr}|\mathbf{y}_s, \mathbf{z}_{sr}) = \mathcal{N}(\mathbf{x}_{sr};\ \mathbf{V}^M\mathbf{y}_s + \mathbf{U}^M\mathbf{z}_{sr}, \mathbf{I}), \tag{46}$$

which are also extended from (32), (33), and (34). The sampling procedure is as follows:

$$\begin{array}{ccccc}
\xrightarrow{P^L(\mathbf{y}_s|\boldsymbol{\mu}_s^{(0)})} \tilde{\mathbf{y}}_s^{L(0)} & & & & \\
 & \xrightarrow{P^L(\mathbf{x}_{sr}|\tilde{\mathbf{y}}_s^{L(0)},\tilde{\mathbf{z}}_{sr}^{L(0)})} & \boldsymbol{\mu}_s^{L(1)} & \xrightarrow{P^L(\mathbf{y}_s|\boldsymbol{\mu}_s^{L(1)})} & \mathbf{y}_s^{L(1)} \\
\xRightarrow{P^L(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{(0)})} \tilde{\mathbf{z}}_{sr}^{L(0)} & & \mathbf{x}_{sr}^{L(1)} & \xRightarrow{P^L(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{L(1)})} & \mathbf{z}_{sr}^{L(1)} \\
\xrightarrow{P^M(\mathbf{y}_s|\boldsymbol{\mu}_s^{(0)})} \tilde{\mathbf{y}}_s^{M(0)} & & & & \\
 & \xrightarrow{P^M(\mathbf{x}_{sr}|\tilde{\mathbf{y}}_s^{M(0)},\tilde{\mathbf{z}}_{sr}^{M(0)})} & \boldsymbol{\mu}_s^{M(1)} & \xrightarrow{P^M(\mathbf{y}_s|\boldsymbol{\mu}_s^{M(1)})} & \mathbf{y}_s^{M(1)} \\
\xRightarrow{P^M(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{(0)})} \tilde{\mathbf{z}}_{sr}^{M(0)} & & \mathbf{x}_{sr}^{M(1)} & \xRightarrow{P^M(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{M(1)})} & \mathbf{z}_{sr}^{M(1)} \\
\xrightarrow{P^R(\mathbf{y}_s|\boldsymbol{\mu}_s^{(0)})} \tilde{\mathbf{y}}_s^{R(0)} & & & & \\
 & \xrightarrow{P^R(\mathbf{x}_{sr}|\tilde{\mathbf{y}}_s^{R(0)},\tilde{\mathbf{z}}_{sr}^{R(0)})} & \boldsymbol{\mu}_s^{R(1)} & \xrightarrow{P^R(\mathbf{y}_s|\boldsymbol{\mu}_s^{R(1)})} & \mathbf{y}_s^{R(1)} \\
\xRightarrow{P^R(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{(0)})} \tilde{\mathbf{z}}_{sr}^{R(0)} & & \mathbf{x}_{sr}^{R(1)} & \xRightarrow{P^R(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{R(1)})} & \mathbf{z}_{sr}^{R(1)}.
\end{array} \tag{47}$$

The gradients of the negative loglikelihood of $\mathbf{x}_s^{(0)}$ with respect to each bound are approximated by

$$\begin{aligned}
-\frac{\partial\mathcal{L}(\mathbf{x}_s^{(0)})}{\partial\mathbf{V}^L} &\approx \frac{n_s}{6}\left(\boldsymbol{\mu}_s^{L(1)}\mathbf{y}_s^{L(1)^T} - \boldsymbol{\mu}_s^{(0)}\mathbf{y}_s^{L(0)^T}\right) \\
-\frac{\partial\mathcal{L}(\mathbf{x}_s^{(0)})}{\partial\mathbf{V}^M} &\approx \frac{2n_s}{3}\left(\boldsymbol{\mu}_s^{M(1)}\mathbf{y}_s^{M(1)^T} - \boldsymbol{\mu}_s^{(0)}\mathbf{y}_s^{M(0)^T}\right) \\
-\frac{\partial\mathcal{L}(\mathbf{x}_s^{(0)})}{\partial\mathbf{V}^R} &\approx \frac{n_s}{6}\left(\boldsymbol{\mu}_s^{R(1)}\mathbf{y}_s^{R(1)^T} - \boldsymbol{\mu}_s^{(0)}\mathbf{y}_s^{R(0)^T}\right)
\end{aligned} \tag{48}$$

$$\begin{aligned}
-\frac{\partial\mathcal{L}(\mathbf{x}_s^{(0)})}{\partial\mathbf{U}^L} &\approx \frac{1}{6}\sum_{r=1}^{n_s}\left(\mathbf{x}_{sr}^{L(1)}\mathbf{z}_{sr}^{L(1)^T} - \mathbf{x}_{sr}^{(0)}\mathbf{z}_{sr}^{L(0)^T}\right) \\
-\frac{\partial\mathcal{L}(\mathbf{x}_s^{(0)})}{\partial\mathbf{U}^M} &\approx \frac{2}{3}\sum_{r=1}^{n_s}\left(\mathbf{x}_{sr}^{M(1)}\mathbf{z}_{sr}^{M(1)^T} - \mathbf{x}_{sr}^{(0)}\mathbf{z}_{sr}^{M(0)^T}\right) \\
-\frac{\partial\mathcal{L}(\mathbf{x}_s^{(0)})}{\partial\mathbf{U}^R} &\approx \frac{1}{6}\sum_{r=1}^{n_s}\left(\mathbf{x}_{sr}^{R(1)}\mathbf{z}_{sr}^{R(1)^T} - \mathbf{x}_{sr}^{(0)}\mathbf{z}_{sr}^{R(0)^T}\right).
\end{aligned} \tag{49}$$

The pseudocode for updating FRBM-PLDA with ATFN is shown in Algorithm 2.

In the evaluation phase, $\mathbf{V}^L$, $\mathbf{V}^M$ and $\mathbf{V}^R$ are used to reduce the dimensionality for a given i-vector $\mathbf{x}$. We can obtain three transformed features, $\mathbf{y}^L = \mathbf{V}^{L^T}\mathbf{x}$, $\mathbf{y}^M = \mathbf{V}^{M^T}\mathbf{x}$ and $\mathbf{y}^R = \mathbf{V}^{R^T}\mathbf{x}$.

Algorithm 2 Update FRBM-PLDA with ATFN

Input

$\mathbf{x}_s^{(0)} = \{\mathbf{x}_{sr}^{(0)}\}_{r=1}^{n_s}$: the prewhitened i-vectors for speaker s

$\mathbf{V}^L$, $\mathbf{V}^M$, $\mathbf{V}^R$: Left bound, center, and right bound of the weights between $\mathbf{x}$ and $\mathbf{y}$

$\mathbf{U}^L$, $\mathbf{U}^M$, $\mathbf{U}^R$: Left bound, center, and right bound of the weights between $\mathbf{x}$ and $\mathbf{z}$

1. compute $\boldsymbol{\mu}_s^{(0)} = n_s^{-1}\sum_{r=1}^{n_s}\mathbf{x}_{sr}^{(0)}$
2. compute $\mathbf{y}_s^{L(0)} = \mathbf{V}^{L^T}\boldsymbol{\mu}_s^{(0)}$, $\mathbf{y}_s^{M(0)} = \mathbf{V}^{M^T}\boldsymbol{\mu}_s^{(0)}$, and $\mathbf{y}_s^{R(0)} = \mathbf{V}^{R^T}\boldsymbol{\mu}_s^{(0)}$
3. sample $\tilde{\mathbf{y}}_s^{L(0)}\sim P^L(\mathbf{y}_s|\boldsymbol{\mu}_s^{(0)})$, $\tilde{\mathbf{y}}_s^{M(0)}\sim P^M(\mathbf{y}_s|\boldsymbol{\mu}_s^{(0)})$, and $\tilde{\mathbf{y}}_s^{R(0)}\sim P^R(\mathbf{y}_s|\boldsymbol{\mu}_s^{(0)})$
4. **for all** $r = 1, \dots, n_s$ **do**
5. compute $\mathbf{z}_{sr}^{L(0)} = \mathbf{U}^{L^T}\mathbf{x}_{sr}^{(0)}$, $\mathbf{z}_{sr}^{M(0)} = \mathbf{U}^{M^T}\mathbf{x}_{sr}^{(0)}$, and $\mathbf{z}_{sr}^{R(0)} = \mathbf{U}^{R^T}\mathbf{x}_{sr}^{(0)}$
6. sample $\tilde{\mathbf{z}}_{sr}^{L(0)}\sim P^L(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{(0)})$, $\tilde{\mathbf{z}}_{sr}^{M(0)}\sim P^M(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{(0)})$, and $\tilde{\mathbf{z}}_{sr}^{R(0)}\sim P^R(\mathbf{z}_{sr}|\mathbf{x}_{sr}^{(0)})$
7. compute $\mathbf{x}_{sr}^{L(1)} = \mathbf{V}^L\tilde{\mathbf{y}}_s^{L(0)} + \mathbf{U}^L\tilde{\mathbf{z}}_{sr}^{L(0)}$, $\mathbf{x}_{sr}^{M(1)} = \mathbf{V}^M\tilde{\mathbf{y}}_s^{M(0)} + \mathbf{U}^M\tilde{\mathbf{z}}_{sr}^{M(0)}$, and $\mathbf{x}_{sr}^{R(1)} = \mathbf{V}^R\tilde{\mathbf{y}}_s^{R(0)} + \mathbf{U}^R\tilde{\mathbf{z}}_{sr}^{R(0)}$
8. **end for**
9. compute $\boldsymbol{\mu}_s^{L(1)} = n_s^{-1}\sum_{r=1}^{n_s}\mathbf{x}_{sr}^{L(1)}$, $\boldsymbol{\mu}_s^{M(1)} = n_s^{-1}\sum_{r=1}^{n_s}\mathbf{x}_{sr}^{M(1)}$, and $\boldsymbol{\mu}_s^{R(1)} = n_s^{-1}\sum_{r=1}^{n_s}\mathbf{x}_{sr}^{R(1)}$
10. compute $\mathbf{y}_s^{L(1)} = \mathbf{V}^{L^T}\boldsymbol{\mu}_s^{L(1)}$, $\mathbf{y}_s^{M(1)} = \mathbf{V}^{M^T}\boldsymbol{\mu}_s^{M(1)}$, and $\mathbf{y}_s^{R(1)} = \mathbf{V}^{R^T}\boldsymbol{\mu}_s^{R(1)}$
11. **for all** $r = 1, \dots, n_s$ **do**
12. compute $\mathbf{z}_{sr}^{L(1)} = \mathbf{U}^{L^T}\mathbf{x}_{sr}^{L(1)}$, $\mathbf{z}_{sr}^{M(1)} = \mathbf{U}^{M^T}\mathbf{x}_{sr}^{M(1)}$, and $\mathbf{z}_{sr}^{R(1)} = \mathbf{U}^{R^T}\mathbf{x}_{sr}^{R(1)}$
13. **end for**
14. $\Delta\mathbf{V}_L = \frac{n_s}{6}\left(\boldsymbol{\mu}_s^{L(1)}\mathbf{y}_s^{L(1)^T} - \boldsymbol{\mu}_s^{(0)}\mathbf{y}_s^{L(0)^T}\right)$
15. $\Delta\mathbf{V}_M = \frac{2n_s}{3}\left(\boldsymbol{\mu}_s^{M(1)}\mathbf{y}_s^{M(1)^T} - \boldsymbol{\mu}_s^{(0)}\mathbf{y}_s^{M(0)^T}\right)$
16. $\Delta\mathbf{V}_R = \frac{n_s}{6}\left(\boldsymbol{\mu}_s^{R(1)}\mathbf{y}_s^{R(1)^T} - \boldsymbol{\mu}_s^{(0)}\mathbf{y}_s^{R(0)^T}\right)$
17. $\Delta\mathbf{U}_L = \frac{1}{6}\sum_{r=1}^{n_s}\left(\mathbf{x}_{sr}^{L(1)}\mathbf{z}_{sr}^{L(1)^T} - \mathbf{x}_{sr}^{(0)}\mathbf{z}_{sr}^{L(0)^T}\right)$
18. $\Delta\mathbf{U}_M = \frac{2}{3}\sum_{r=1}^{n_s}\left(\mathbf{x}_{sr}^{M(1)}\mathbf{z}_{sr}^{M(1)^T} - \mathbf{x}_{sr}^{(0)}\mathbf{z}_{sr}^{M(0)^T}\right)$
19. $\Delta\mathbf{U}_R = \frac{1}{6}\sum_{r=1}^{n_s}\left(\mathbf{x}_{sr}^{R(1)}\mathbf{z}_{sr}^{R(1)^T} - \mathbf{x}_{sr}^{(0)}\mathbf{z}_{sr}^{R(0)^T}\right)$
20. update $\mathbf{V}_L$, $\mathbf{V}_M$, $\mathbf{V}_R$, $\mathbf{U}_L$, $\mathbf{U}_M$, $\mathbf{U}_R$ using $\Delta\mathbf{V}_L$, $\Delta\mathbf{V}_M$, $\Delta\mathbf{V}_R$, $\Delta\mathbf{U}_L$, $\Delta\mathbf{U}_L$, $\Delta\mathbf{U}_R$

IV. Scoring for FRBM-PLDA

As shown in Section III, we can obtain two features, $\mathbf{y}^L$ and $\mathbf{y}^R$, with STFN or three features $\mathbf{y}^L$, $\mathbf{y}^M$ and $\mathbf{y}^R$ with ATFN. However, traditional scoring methods take only one enrollment and one test feature as input. To exploit all features from FRBM-PLDA, we propose modified scoring methods.

A. Cosine similarity

Given a pair of an enrollment feature $\mathbf{y}_e$ and a test feature $\mathbf{y}_t$, cosine similarity [1] is computed as follows:

$$\cos(\mathbf{y}_e, \mathbf{y}_t) = \frac{\mathbf{y}_e{}^T\mathbf{y}_t}{\|\mathbf{y}_e\|\|\mathbf{y}_t\|}. \tag{50}$$

We define the modified cosine scoring method for STFN as:

$$\cos(\mathbf{y}_e^L, \mathbf{y}_t^L) + \cos(\mathbf{y}_e^R, \mathbf{y}_t^R) \tag{51}$$

and the modified scoring method for ATFN as:

$$\cos(\mathbf{y}_e^L, \mathbf{y}_t^L) + \cos(\mathbf{y}_e^M, \mathbf{y}_t^M) + \cos(\mathbf{y}_e^R, \mathbf{y}_t^R). \tag{52}$$

Equations (51) and (52) are simple score fusions.

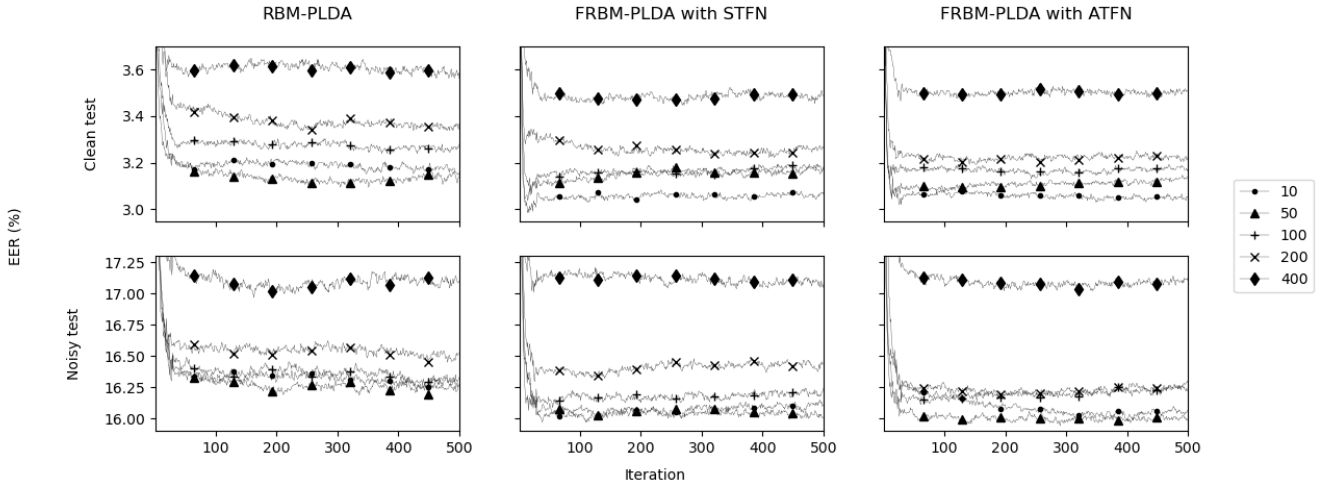

Fig. 5. The EERs according to the number of session factors (distinguished by line marker) of RBM-PLDA (left), FRBM-PLDA with STFN (middle) and with ATFN (right) on *development* trials during 500 iterations. All models were trained using clean utterances only (*CO*). All test utterances were clean (top) or contained *subway* noise at 5 dB (bottom).

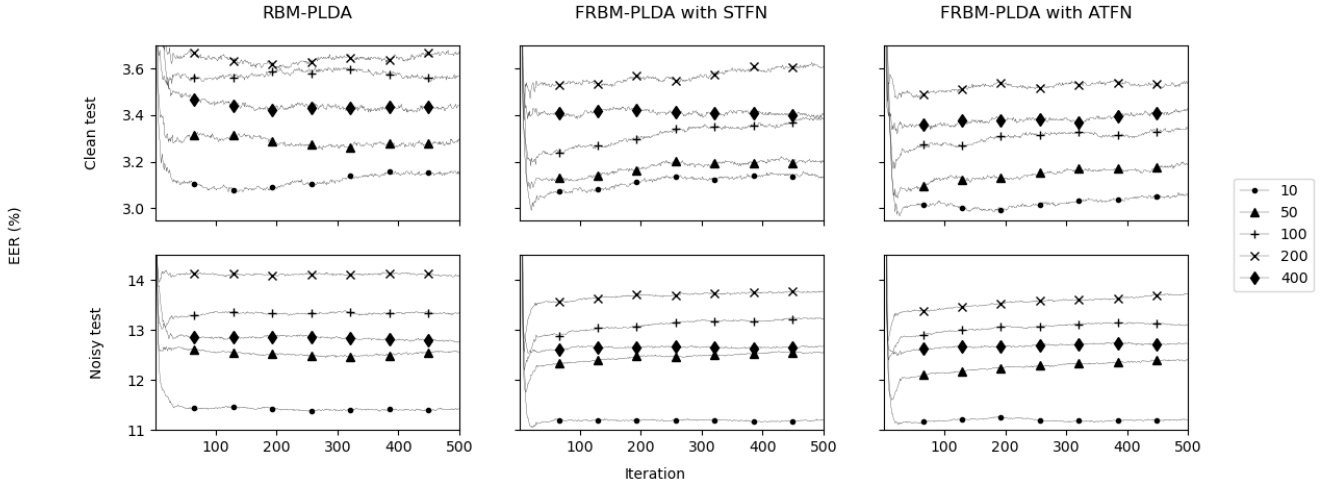

Fig. 6. EERs according to the number of session factors (distinguished by line marker) of RBM-PLDA (left), FRBM-PLDA with STFN (middle) and with ATFN (right) on *development* trials during 500 iterations. All models were trained using clean and noisy utterances together (*AN*). All test utterances were clean (top) or contained *subway* noise at 5 dB (bottom).

B. Negative-squared Euclidean distance

Stafylakis [15] used the negative-squared Euclidean distance between $\mathbf{y}_e$ and $\mathbf{y}_t$ as a scoring method for RBM-PLDA, as follows:

$$-d^2(\mathbf{y}_e,\mathbf{y}_t) = -(\mathbf{y}_e - \mathbf{y}_t)^T(\mathbf{y}_e - \mathbf{y}_t). \quad (53)$$

Note that it has the same effect as (50) if the lengths of both $\mathbf{y}_e$ and $\mathbf{y}_t$ are normalized.

For STFN, we first concatenate all features $\mathbf{y} = \left[\mathbf{y}^{L^T} \vdots \mathbf{y}^{R^T}\right]^T$, and then use the concatenated features to (53) as input. If the lengths of both $\mathbf{y}^L$ and $\mathbf{y}^R$ are normalized before concatenation, it has the same effect as (51). Therefore, we do not normalize the lengths of $\mathbf{y}^L$ and $\mathbf{y}^R$. For ATFN, we concatenate all features $\mathbf{y} = \left[\mathbf{y}^{L^T} \vdots \mathbf{y}^{M^T} \vdots \mathbf{y}^{R^T}\right]^T$ and then use the same approach shown for STFN. It has the same effect as (52) if all lengths of $\mathbf{y}^L$, $\mathbf{y}^M$ and $\mathbf{y}^R$ are normalized before concatenation. Therefore, we do not normalize the lengths of $\mathbf{y}^L$, $\mathbf{y}^M$ and $\mathbf{y}^R$.

C. PLDA

As shown in Section II-*C*, PLDA also takes only one enrollment and one test feature as input. We concatenate all features before using it as an input to PLDA, as described in Section IV-*B*. Unlike the case of using the i-vector; however, we observe that it shows slightly better performance without length normalization. Therefore, we do not normalize the lengths of all features obtained from FRBM-PLDA.

V. Experiments

A. Database

We used Part 1 of the Robust Speaker Recognition (RSR) 2015 [30] dataset. It consists of clean utterances from 300 speakers. It is divided into the *background* (50 male, 47 female), *development* (50 male, 47 female), and *evaluation* (57 male, 49 female) sets. Each of the speakers utters 30 kinds of phrases (average voiced duration of 1.25 s) in nine different sessions. Actually, the number of all utterances is lower than 81,000 (300×30×9) because a few utterances do not exist.

TABLE I
EERs ON RSR 2015 MALE EVALUATION TRIALS. ALL BACKGROUND MODELS WERE TRAINED USING CLEAN UTTERANCES (*CO*).

		i-vector	LDA	RBM-PLDA	FRBM-PLDA	
					STFN	ATFN
Clean test	C	1.7571	2.9188	2.2745	2.1183	2.1769
	D			2.2159	2.0597	2.0402
	P	1.4155	2.1476	2.0695	1.5814	**1.4057**
Bus 0dB test	C	25.4198	26.9133	22.9988	22.9793	**22.7548**
	D			25.6052	25.1562	25.3221
	P	26.3471	26.4155	26.064	25.8981	26.3764
Bus 5dB test	C	14.428	17.0246	13.1687	13.1394	**12.9539**
	D			14.5841	14.1351	14.2132
	P	15.082	16.1948	15.5506	14.838	15.0918
Bus 10dB test	C	7.624	10.4451	7.253	7.009	7.1359
	D			7.1749	6.9602	**6.9406**
	P	7.6728	9.7618	9.0199	7.7997	7.663
cafe 0dB test	C	27.5381	29.0316	25.3524	25.3417	**25.0879**
	D			27.6162	27.3819	27.46
	P	28.9926	28.8852	28.524	28.8071	29.0414
cafe 5dB test	C	16.8196	19.7872	15.9313	15.902	**15.6872**
	D			17.0246	16.8489	16.722
	P	18.2546	19.2698	18.6646	18.1765	18.2546
cafe 10dB test	C	9.303	12.6415	9.2152	9.1761	**9.0199**
	D			9.2542	9.098	9.059
	P	10.1035	12.2413	11.4701	10.1718	10.0742

C: cosine similarity / D: negative Euclidean distance / P: PLDA.
The lowest EERs are highlighted in bold.

TABLE II
EERs ON RSR 2015 FEMALE EVALUATION TRIALS. ALL BACKGROUND MODELS WERE TRAINED USING CLEAN UTTERANCES (*CO*).

		i-vector	LDA	RBM-PLDA	FRBM-PLDA	
					STFN	ATFN
Clean test	C	**1.7253**	4.8695	3.4279	3.2804	3.3258
	D			2.2134	2.1793	2.0091
	P	1.8388	3.6663	3.3485	2.1453	1.8275
Bus 0dB test	C	26.8281	30.5675	25.1759	25.0738	**24.7446**
	D			27.5369	26.9467	26.9807
	P	30.7151	32.0545	31.1918	30.6129	30.6924
Bus 5dB test	C	15.2327	20.6129	15.6527	15.3348	15.2327
	D			15.7775	15.3121	**14.9262**
	P	17.8888	20.6583	20.0341	18.2406	17.9342
Bus 10dB test	C	7.6617	12.8036	8.933	8.8876	8.706
	D			7.8774	7.4461	**7.3212**
	P	8.2747	11.294	10.9989	8.8649	8.3314
cafe 0dB test	C	27.7866	32.6107	27.798	**27.3553**	27.4234
	D			29.3871	29.1714	28.8649
	P	32.1339	33.6209	33.042	32.4064	32.1226
cafe 5dB test	C	**16.5494**	22.9966	17.7412	17.4688	17.4915
	D			17.8547	17.412	17.1396
	P	20.2497	22.4291	22.168	20.7946	20.2384
cafe 10dB test	C	**8.5585**	14.7333	10.5675	10.2497	10.2611
	D			9.4211	9.2168	8.8082
	P	10.1703	13.0647	12.8944	10.7832	10.1362

C: cosine similarity / D: negative Euclidean distance / P: PLDA.
The lowest EERs are highlighted in bold. The underline means the lowest EERs for the cases of comparing FRBM-PLDA with RBM-PLDA only.

To evaluate the performance in noisy environments, we obtained seven types of noise from [31]: *babble*, *bus*, *cafe*, *car*, *metro*, *station*, and *subway*. *Babble*, *car* and *metro* noises were added to all utterances of the *background* set at 0, 5 and 10 dB. *Subway* noise was added to the test utterances of the *development* set at 5 dB. *Bus* and *cafe* noises were added to the test utterances of the *evaluation* set at 0, 5 and 10 dB. Note that for the *development* and *evaluation* sets, we added noise to the test utterance only for simulating real-world environments, wherein it was relatively difficult to obtain clean utterances in testing speaker verification algorithms. We used the filtering and noise adding tool (FaNT) [32] to add noise.

The *background* set was used to build gender-independent models: a GMM-UBM, an i-vector extractor, LDAs, RBM-PLDAs, FRBM-PLDAs, and PLDAs. The GMM-UBM and i-vector extractor were trained using clean utterances only. Each of LDA, RBM-PLDA, FRBM-PLDA, and PLDA was trained in two ways: 1) using clean utterances only (referred to as *CO*) and 2) using clean and noise utterances together (referred to as *AN*, which used 13 times more utterances than *CO*).

The *development* set was used to validate RBM-PLDAs and FRBM-PLDAs during training. To determine the number of training iterations and select hyperparameters (e.g., the number of session factors), we monitored the MSE of RBM-PLDAs and FRBM-PLDAs, and equal error rates (EERs) on gender-independent trials from the *development* set.

The *evaluation* set was used to evaluate the performance of our proposed methods on gender-dependent trials. The number of evaluation trials was 583,566 for males and 431,739 for females. Each (speaker + phrase) model was enrolled with 3 utterances and was tested using the other 6 utterances of the same phrase.

B. Experimental setup

The acoustic feature vectors were 60 dimensional: 19 dimensional mel-frequency cepstral coefficients (MFCCs) + energy + delta + acceleration. These were extracted using a 25-ms long Hamming window with a 10-ms shift. Mean normalization was applied using a 300 frame (approximately 3 s) sliding window. Silence frames were removed according to energy-based voice activity detection (VAD). In the case of noisy utterances, simple energy-based VAD is prone to incorrectly detecting an unvoiced frame as a voiced frame, which affects the speaker verification performance. Since we are not interested in the VAD performance in this paper, we used the VAD results obtained from clean utterances. A gender-independent GMM-UBM consists of 1,024 mixture components with diagonal covariance. It was trained for ten iterations of the expectation-maximization (EM) algorithm. A gender-independent 400-dimensional i-vector extractor was trained for five iterations of the EM algorithm. Length normalization [9] was applied to the i-vectors. The Kaldi speech recognition toolkit [33] was used to extract the acoustic features and i-vectors, and to build all background models (i.e., GMM-UBM and i-vector extractor).

Regardless of the means of training (i.e., *CO* or *AN*), LDAs were applied to reduce the dimensionality of the i-vector to 150. They were implemented using Kaldi.

RBM-PLDAs and FRBM-PLDAs share 400 visible units and 150 speaker factors. The Adam [34] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ was used and L2 regularization with $\lambda = 0.1$ was applied for training. $\mathbf{V}$ and $\mathbf{U}$ were initialized to random numbers generated from $\mathcal{N}(\mathbf{0}, 10^{-3}\mathbf{I})$. $\mathbf{V}^L$, $\mathbf{V}^R$, $\mathbf{U}^L$ and $\mathbf{U}^R$ were initialized to random numbers generated from $\mathcal{N}(\mathbf{0}, 10^{-3}\mathbf{I})$ first. To make all values of $\mathbf{V}^L$ and $\mathbf{U}^L$ negative, we employed the absolute values of $\mathbf{V}^L$ and $\mathbf{U}^L$ and then multiplied them by -1. Then, $\mathbf{V}^M$ and $\mathbf{U}^M$ were initialized as shown in (43). For *CO*, we set the learning rate of 10^{-4} until 30 iterations and 10^{-5} after 30 iterations (if the number of iterations was greater than 30). For *AN*, we set the learning rate 5×10^{-5} until 30 iterations and 5×10^{-6} after 30 iterations. RBM-PLDA-*CO* had 50 session factors and was trained for 200 iterations. RBM-PLDA-*AN* ten session factors and was trained for 100

TABLE III
EERs ON RSR 2015 MALE EVALUATION TRIALS. ALL BACKGROUND MODELS WERE TRAINED USING CLEAN AND NOISY UTTERANCES (*AN*).

		i-vector	LDA	RBM-PLDA	FRBM-PLDA	
					STFN	ATFN
Clean test	C	**1.7571**	2.7333	2.4795	2.4502	2.3233
	D			2.5647	2.4209	2.3819
	P	1.7962	2.704	2.5576	1.8743	1.7962
Bus 0dB test	C	25.4198	18.5865	15.6872	15.4529	15.2382
	D			17.2589	16.4779	16.4389
	P	14.7306	17.0441	15.5994	**14.5744**	14.916
Bus 5dB test	C	14.428	10.4842	8.6099	8.5318	8.1804
	D			9.0102	8.6002	8.3073
	P	**7.1164**	9.3421	8.4733	7.370	7.4678
Bus 10dB test	C	7.624	6.1597	5.3202	5.2811	5.0078
	D			5.4959	5.164	5.0664
	P	**3.6607**	5.5838	4.9492	3.9731	3.7095
cafe 0dB test	C	27.5381	22.8817	19.6896	19.2405	19.1234
	D			21.1343	20.2265	20.4315
	P	19.7677	22.2765	19.5529	**19.0258**	19.9043
cafe 5dB test	C	16.8196	14.3889	11.8606	11.5189	11.5092
	D			12.6904	11.8996	11.8801
	P	10.6209	13.403	11.3725	**10.4647**	10.7966
cafe 10dB test	C	9.303	8.4537	7.2921	6.9309	6.7357
	D			7.5264	6.9797	6.8626
	P	**5.4666**	7.9071	6.6576	5.5838	5.574

C: cosine similarity / D: negative Euclidean distance / P: PLDA.
The lowest EERs are highlighted in bold. The underline means the lowest EERs for the cases of comparing FRBM-PLDA with RBM-PLDA only.

TABLE IV
EERs ON RSR 2015 FEMALE EVALUATION TRIALS. ALL BACKGROUND MODELS WERE TRAINED USING CLEAN AND NOISY UTTERANCES (*AN*).

		i-vector	LDA	RBM-PLDA	FRBM-PLDA	
					STFN	ATFN
Clean test	C	**1.7253**	4.143	3.1555	3.042	3.1442
	D			2.7355	2.5993	2.6334
	P	2.7242	4.0409	3.6549	2.7015	2.6901
Bus 0dB test	C	26.8281	21.5778	17.5482	17.3439	17.1964
	D			17.6504	17.0715	16.7537
	P	16.6061	19.5119	17.5596	**16.4245**	16.8899
Bus 5dB test	C	15.2327	13.4166	10.0341	10.0908	9.9659
	D			9.4779	9.3757	9.0352
	P	**8.8763**	11.8161	10.3292	8.9217	9.0125
Bus 10dB test	C	7.6617	8.3087	6.3678	6.277	6.2997
	D			5.8456	5.6527	5.4257
	P	**5.1986**	7.4347	6.6175	5.3916	5.21
cafe 0dB test	C	27.7866	26.1017	21.3734	20.9762	21.0102
	D			21.9977	21.4188	21.1237
	P	21.714	24.7446	22.0204	**20.9535**	21.8842
cafe 5dB test	C	16.5494	16.6402	12.8036	12.5085	12.4404
	D			12.3723	11.8956	**11.6799**
	P	12.1566	15.0851	13.4279	11.9069	12.2134
cafe 10dB test	C	8.5585	10.2043	7.4574	7.2191	7.4234
	D			6.9012	**6.4813**	6.5494
	P	6.6175	9.2736	8.1271	6.8104	6.6402

C: cosine similarity / D: negative Euclidean distance / P: PLDA.
The lowest EERs are highlighted in bold. The underline means the lowest EERs for the cases of comparing FRBM-PLDA with RBM-PLDA only.

iterations. FRBM-PLDA-*CO* with STFN had ten session factors and was trained for 80 iterations. FRBM-PLDA-*AN* with STFN had ten session factors and was trained for 20 iterations. FRBM-PLDA-*CO* with ATFN had 50 session factors and was trained for 30 iterations. FRBM-PLDA-*AN* with ATFN had ten session factors and was trained for 20 iterations. The numbers of iterations and session factors for each model were determined based on EER (Figures 5 and 6) on *development* trials. Note in Figures 5 and 6 that FRBM-PLDAs converged to an EER in fewer iterations than RBM-PLDA. It was also observed that the MSE is an unsuitable factor to determine hyperparameters. These were implemented using TensorFlow [35].

The PLDA models were trained for each type of feature (i.e., raw i-vector, LDA transformed, the speaker factors from RBM-PLDA and FRBM-PLDA). The raw i-vector and LDA transformed i-vector were length-normalized. However, the speaker factors from RBM-PLDA and FRBM-PLDA were not normalized, as mentioned in Section IV-*C*. All PLDAs were trained for ten iterations of the EM algorithm and implemented using Kaldi.

All LDAs, PLDAs, RBM-PLDAs, and FRBM-PLDAs were trained on 2,910 classes (97 speakers×30 phrases) from the *background* set.

C. Results

Our main objective was to compare the performance differences when all the parameters of the RBM-PLDA were replaced with fuzzy numbers with the ultimate goal of considering parameter uncertainties motivated by real-world applications. We focused on a comparison of EERs of RBM-PLDA with those of FRBM-PLDA.

Tables I and II show the EERs of the male and female evaluation trials, respectively. All models were trained using only clean utterances. Except in a clean test environment, the PLDA scoring method shows higher EERs than both cosine similarity and negative Euclidean distance. It seems that PLDAs trained by clean utterances only cannot compensate for noise variability properly. In the male trials, the FRBM-PLDA, especially with ATFN, showed the lowest EER in all environments. In female trials, however, FRBM-PLDA, with both STFN and ATFN, shows a slightly higher EER than raw i-vector/cosine similarity in clean, *cafe* noise at 5 and 10 dB environments. It is thought that female utterances are more sensitive than male utterances. However, FRBM-PLDAs are still competitive even in clean, cafe noise at 5 and 10 dB. FRBM-PLDA shows lower EERs than RBM-PLDA in all cases. It can be inferred that FRBM-PLDA is more noise-robust than both RBM-PLDA and PLDA because it more effectively considers the uncertainty than the latter models, even if the noise variability is not learned.

Tables III and IV show the EERs of the male and female evaluation trials, respectively. All models were trained using clean and noisy utterances together; therefore, the noise variability is known. Note that the results of raw i-vector/cosine similarity in Tables III and IV are the same as those in Tables I and II, respectively. In clean test environments, the i-vector/cosine similarity shows the lowest EERs, which seems that all models are better able to compensate for the noise variability, but are less able to compensate for the variability in a clean environment. FRBM-PLDA shows the lowest EERs in the environments of *bus* noise at 0 dB, *cafe* noise at 0 and 5 dB on male trials, and in the environments of *bus* noise at 0 dB, *cafe* noise at 0, 5 and 10 dB on female trials. It is apparent that PLDA alone can adequately compensate for noise variability if the noise variability is relatively small (i.e., a high signal-to-noise ratio (SNR) or the type of noise that has a narrow bandwidth). If the noise variability is relatively large (i.e., a low SNR or the type of noise that has a wide bandwidth); however, PLDA alone cannot adequately compensate for the noise variability. In these cases, using it with FRBM (i.e., FRBM-PLDA) can compensate for the variability that is not captured by PLDA.

Regardless of scoring methods, LDA-*CO* (see Tables I and II) shows the highest EERs in all cases except only one case of LDA for *cafe* noise at 0 dB in Table I. It is thought that the phonetic variability is relatively larger than other variabilities on short durations, as mentioned in [36]. RBM-PLDA shows considerably lower EERs than LDA, and FRBM-PLDA shows lower EERs than RBM-PLDA in all cases. It is evident that both RBM-PLDA and FRBM-PLDA, unlike LDA, compensate for the phonetic variability to some degree without a phrase-dependent model [36]. Additionally, FRBM-PLDA can compensate for the variability, which is not captured by RBM-PLDA.

VI. Conclusion

In this paper, we proposed FRBM-PLDA, which extends RBM-PLDA by applying fuzzy theory to crisp parameters. In particular, we applied FRBM to treat real-valued data (e.g., i-vector) rather than binary data, which results in GGFRBM-PLDA for the proposed algorithm. We also proposed the modified scoring methods for FRBM-PLDA to exploit all features from FRBM-PLDA. Evaluations were conducted on Part 1 of the RSR 2015 evaluation trials, which are used for text-dependent speak verification on short utterance conditions. We used noise data to evaluate the performance in situations considering real-world applications. We observed that FRBM-PLDA showed lower EERs than RBM-PLDA for all our experiments and the lowest EERs in low SNR conditions. Additionally, both RBM-PLDA and FRBM-PLDA showed considerably lower EERs than LDA. We conclude that applying fuzzy theory to RBM (i.e., FRBM) and applying it for PLDA (i.e., FRBM-PLDA) provides more robust performance on short-duration text-dependent tasks by well compensating for phonetic and noise variabilities.

Appendix

The original energy function of the GGRBM is defined by

$$E(\mathbf{x},\mathbf{h};\Theta)=\sum_i^M\frac{(x_i-a_i)^2}{2\sigma_i^2}+\sum_j^N\frac{(h_j-b_j)^2}{2\rho_j^2}-\sum_{i,j}^{M,N}\frac{x_i}{\sigma_i}\frac{h_j}{\rho_j}W_{ij}$$
$$=\frac{1}{2}(\mathbf{x}-\mathbf{a})^T\boldsymbol{\Lambda}^{-1}(\mathbf{x}-\mathbf{a})+\frac{1}{2}(\mathbf{h}-\mathbf{b})^T\boldsymbol{\Omega}^{-1}(\mathbf{h}-\mathbf{b})$$
$$-\mathbf{x}^T\boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{W}\boldsymbol{\Omega}^{-\frac{1}{2}}\mathbf{h} \qquad (53)$$

where σ_i and ρ_j are the standard deviations for x_i and h_j, respectively, and $\boldsymbol{\Lambda}=\mathrm{diag}(\sigma_1^2,\dots,\sigma_M^2)$ and $\boldsymbol{\Omega}=\mathrm{diag}(\rho_1^2,\dots,\rho_N^2)$ are the diagonal covariance matrices for $\mathbf{x}$ and $\mathbf{h}$, respectively. The original conditional probabilities for the GGRBM are

$$P(\mathbf{h}|\mathbf{x})=\prod_j^N P(h_j|\mathbf{x})=\prod_j^N\mathcal{N}(h_j;\ b_j+\rho_j\sum_i^M\frac{x_i}{\sigma_i}W_{ij},\rho_j^2)$$
$$=\mathcal{N}\Big(\mathbf{h};\mathbf{b}+\boldsymbol{\Omega}^{\frac{1}{2}}\mathbf{W}^T\boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{x},\boldsymbol{\Omega}\Big) \qquad (54)$$

$$P(\mathbf{x}|\mathbf{h})=\prod_i^M P(x_i|\mathbf{h})=\prod_i^M\mathcal{N}(x_i;\ a_i+\sigma_i\sum_j^N\frac{h_j}{\rho_j}W_{ij},\sigma_i^2)$$
$$=\mathcal{N}\Big(\mathbf{x};\mathbf{a}+\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{W}\boldsymbol{\Omega}^{-\frac{1}{2}}\mathbf{h},\boldsymbol{\Lambda}\Big). \qquad (55)$$

If all σ_i and ρ_j are assumed to be 1, then $\boldsymbol{\Lambda}$ and $\boldsymbol{\Omega}$ become identity matrix $\mathbf{I}$ and can be omitted.

Appendix B

For FRBM-PLDA with STFN, the gradients are applied equally. The gradients with respect to $\mathbf{W}^L$ and $\mathbf{W}^R$ are

$$-\frac{\partial\log P(\mathbf{x}^{(0)})}{\partial\mathbf{W}^L}=-\frac{\partial\log P(\mathbf{x}^{(0)})}{\partial\mathbf{W}^R}\approx\frac{1}{2}\Big(\mathbf{x}^{(1)}\mathbf{h}^{(1)^T}-\mathbf{x}^{(0)}\mathbf{h}^{(0)^T}\Big) \qquad (56)$$

where

$$\mathbf{x}^{(k)}=\frac{1}{2}\big(\mathbf{x}^{L(k)}+\mathbf{x}^{R(k)}\big)$$
$$\mathbf{h}^{(k)}=\frac{1}{2}\big(\mathbf{h}^{L(k)}+\mathbf{h}^{R(k)}\big). \qquad (57)$$

For FRBM-PLDA with ATFN, we applied the gradients with $\mathbf{W}^L$, $\mathbf{W}^M$ and $\mathbf{W}^R$ as follows:

$$-\frac{\partial\log P(\mathbf{x}^{(0)})}{\partial\mathbf{W}^L}=-\frac{\partial\log P(\mathbf{x}^{(0)})}{\partial\mathbf{W}^R}\approx\frac{1}{6}\Big(\mathbf{x}^{(1)}\mathbf{h}^{(1)^T}-\mathbf{x}^{(0)}\mathbf{h}^{(0)^T}\Big) \qquad (58)$$

$$-\frac{\partial\log P(\mathbf{x}^{(0)})}{\partial\mathbf{W}^M}\approx\frac{4}{6}\Big(\mathbf{x}^{(1)}\mathbf{h}^{(1)^T}-\mathbf{x}^{(0)}\mathbf{h}^{(0)^T}\Big) \qquad (59)$$

where

$$\mathbf{x}^{(k)}=\frac{1}{6}\big(\mathbf{x}^{L(k)}+4\mathbf{x}^{L(k)}+\mathbf{x}^{R(k)}\big)$$
$$\mathbf{h}^{(k)}=\frac{1}{6}\big(\mathbf{h}^{L(k)}4\mathbf{h}^{L(k)}+\mathbf{h}^{R(k)}\big). \qquad (60)$$

To monitor the training progress, we used MSE between the input, $\mathbf{x}^{(0)}$ and the reconstruction, $\mathbf{x}^{(1)}$. Figure 7 shows the MSEs of RBM-PDA and FRBM-PLDA with STFN and ATFN. The gradients of FRBM-PLDAs were applied differently. Figure 8 shows the MSEs of FRBM-PLDA with STFN and ATFN. The results of Figure 8 are significantly higher than those of Figure 7, especially with STFN. This means that if the gradients are applied equally, the convergence speed is very slow or may not converge to the level of applying the gradients differently.

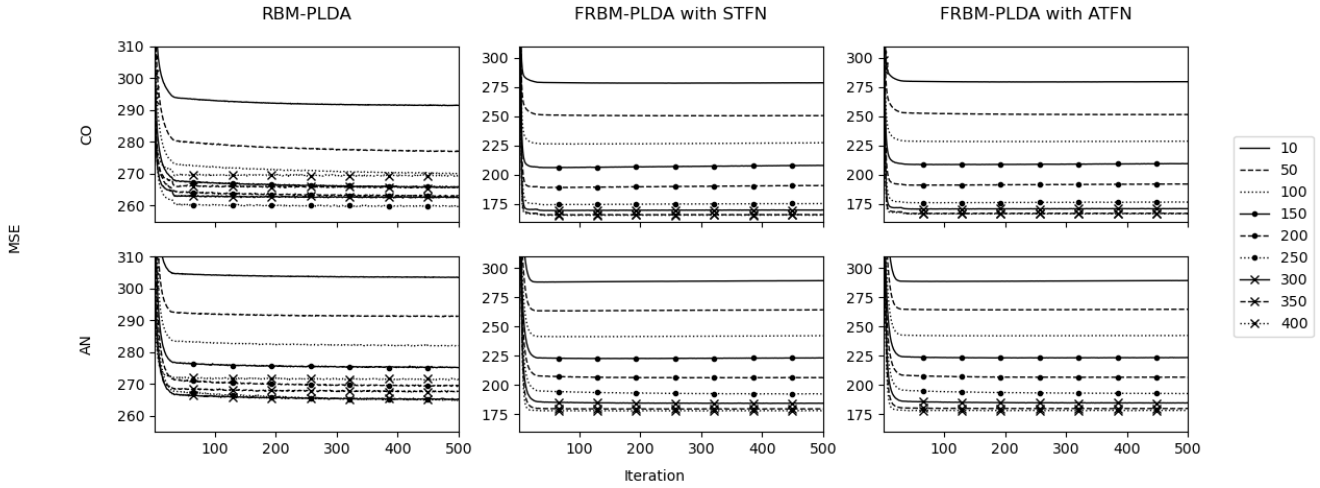

Fig. 7. MSEs according to the number of session factors (distinguished by line marker and style) of RBM-PLDA (left), FRBM-PLDA with STFN (middle) and with ATFN (right) on all clean utterances from the *development* set during 500 iterations when applying gradients differently. The top row shows the MSEs of the models trained using clean utterances only (*CO*). The bottom row shows the MSEs of the models trained using clean and noisy utterances together (*AN*).

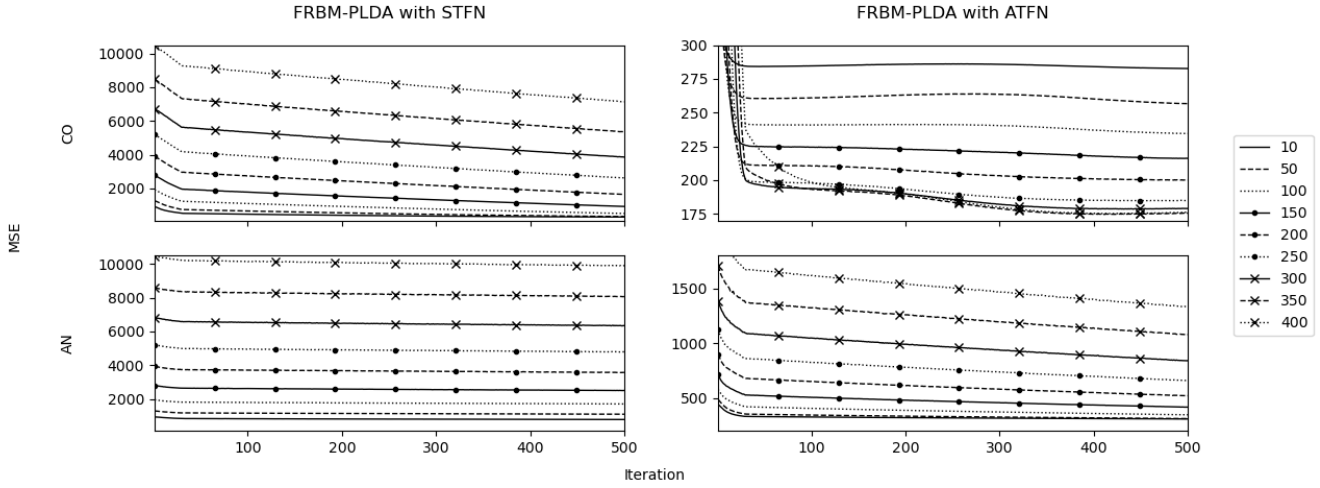

Fig. 8. MSEs according to the number of session factors (distinguished by line marker and style) of FRBM-PLDA with STFN (left) and with ATFN (right) on all clean utterances from *development* set during 500 iterations when applying gradients equally. The top row shows the MSEs of the models trained using clean utterances only (*CO*). The bottom row shows the MSEs of the models trained using clean and noisy utterances together (*AN*).

ACKNOWLEDGMENT

This research was supported by Projects for Research and Development of Police Science and Technology under the Center for Research and Development of Police Science and Technology and Korean National Police Agency funded by the Ministry of Science, ICT and Future Planning (PA-J000001-2017-101).

REFERENCES

[1] 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, no. 4, pp788–798, May 2011.

[2] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp1435–1447, May 2007.

[3] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in GMM-based speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp1448–1460, May 2007.

[4] N. Dehak, "Discriminative and generative approaches for long- and short-term speaker characteristics modeling: Application to speaker verification," Ph.D. dissertation, École de Technologie Supérieure, Montreal, QC, Canada, 2009.

[5] S. Balakrishnama and A. Ganapathiraju, "Linear discriminant analysis-A brief tutorial," *Institute for Signal and Information Processing*, vol. 18, pp1-8, 1998.

[6] S. Ioffe, "Probabilistic linear discriminant analysis," European Conference on Computer Vision, 2006, pp531-542.

[7] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inference about identity," in *Proceedings of 11th International Conference on Computer Vision*, 2007, pp1-8.

[8] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proceedings of Odyssey Speaker and Language Recognition Workshop*, Jul. 2010, p14.

[9] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proceedings of Interspeech*, Aug. 2011, pp249–252."

[10] N. Brummer and E. de Villiers, "The speaker partitioning problem," in Proceedings of Odyssey, Jul. 2010, p34.

[11] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, pp194-281, Cambridge, MA: MIT Press, 1986.

[12] C. L. P. Chen, C. Y. Zhang, L. Chen, and M. Gan, "Fuzzy restricted Boltzmann machine for the enhancement of deep learning," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 6, pp.2163-2173, 2015.

[13] S. Feng and C. L. P. Chen, "A fuzzy restricted Boltzmann machine: Novel learning algorithms based on crisp possibilistic mean value of

fuzzy numbers," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 1, pp117-130, Feb. 2018.
[14] G. J. Klir and B. Yuan, "Fuzzy sets and fuzzy logic: Theory and applications," Englewood Cliffs, NJ, USA: Prentice-Hall, 1995.
[15] T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel, "PLDA using Gaussian restricted Boltzmann machines with application to speaker verification," in *Proceedings of Interspeech*, Sep. 2012, pp1692-1696.
[16] A. Nautsch, H. Hao, T. Stafylakis, C. Rathgeb, and C. Busch, "Towards PLDA-RBM based speaker recognition in mobile environment: Designing stacked/deep PLDA-RBM systems," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp5055-5059.
[17] P. Melin, J. Urias, D. Solano, M. Soto, M. Lopez, and O. Castillo, "Voice recognition with neural networks, type-2 fuzzy logic and genetic algorithms," Engineering Letters, vol. 13, no. 2, pp108-116, 2006.
[18] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, and M. Mason, "I-vector based speaker recognition on short utterances," in *Proceedings of Interspeech*, Aug. 2011, pp2341-2344.
[19] M. Hebert, "Text-dependent speaker recognition," *Springer handbook of speech processing*, Springer Berlin Heidelberg, pp743-762, 2008.
[20] A. Larcher, P. M. Bousquet, K. A. Lee, D. Matrouf, H. Li, and J. F. Bonastre, "I-vectors in the context of phonetically-constrained short utterances for speaker verification," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2012, pp4773-4776.
[21] L. Lei and S. Kun, "Speaker recognition using wavelet packet entropy, i-vector, and cosine distance scoring," Journal of Electrical and Computer Engineering, 2017.
[22] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp308–311, 2006.
[23] D. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1, pp19–41, 2000.
[24] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp345–354, May 2005.
[25] A. S. Asratian, T. M. Deneley, and R. Haggkvist, "Bipartite graphs and their applications," Cambridge university press, 1998.
[26] T. Yamasita, M. Tanaka, E. Yoshida, Y. Yamauchi, and H. Fujiyoshi, "To be Bernoulli or to be Gaussian, for a restricted Boltzmann machine," in Proceedings of IEEE International Conference on Pattern Recognition, Aug. 2014, pp1520-1525.
[27] S. Ogawa and H. Mori, "A Gaussian-Gaussian-restricted-Boltzmann-machine-based deep neural network technique for photovoltaic system generation forecasting," *IFAC-PapersOnLine*, vol. 52, no. 4, pp87-92, 2019.
[28] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp1711-1800, 2002.
[29] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," Technical Report UTML TR 2010-003, Department of Computer Science, University of Toronto, 2010.
[30] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Communication*, vol. 60, pp56-77, 2014.
[31] E. Fonseca, J. Pons, X. favory, F. Font, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: A platform for the creation of open audio datasets," in *Proceedings of International Society for Music Information Retrieval (ISMIR)*, Oct. 2017, pp486-493.
[32] H. G. Hirsch, "FaNT-Filtering and noise adding tool," *Niederrhein University of Applied Science*, [Online]. Available: http://dnt.kr.hsnr.de/download/fant_manual.pdf, 2005
[33] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proceedings of the Automatic Speech Recognition & Understanding (ASRU) Workshop*, Dec. 2011.
[34] D. P. Kingma and B. Jimmy, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[35] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for large-scale machine learning," in *Proceedings of 12th USENIX symposium on Operating System Design and Implementation (OSDI)*, Nov. 2016, pp265-283.
[36] T. Stafylakis, P. Kenny, P. Ouellet, J. Perez, and M. Kockmann, "Text-dependent speaker recognition using PLDA with uncertainty propagation," in *Proceedings of Interspeech*, Aug. 2013, pp3684-3688.

Sung-Hyun Yoon received the B.S. degree in computer science from the University of Seoul, Republic of Korea, in 2015, where he is currently pursuing the combined M.S. and Ph.D. degrees with the School of Computer Science. His research interests include speaker recognition, spoofing detection for automatic speaker verification, and machine learning.

Min-Sung Koh received the Ph.D. degree in electrical engineering from Washington State University, Pullman, WA, USA. He was with Korea Electric Power Corporation (KEPCO) for nine years before enrolling in the Ph.D. program. In 2002, he joined the Department of Engineering and Design, Eastern Washington University, Cheney, WA, USA, where he is currently a Full Professor of electrical engineering. His research interests include speech/image signal processing, machine learning, adaptive signal processing, matrix theories, multivariate statistical analysis, signal processing on graphs, brain–computer interface, biomedical signal processing, hardware implementation of signal processing algorithms, and signal processing in communication systems.

Ha-Jin Yu received the B.S., M.S., and Ph.D. degrees in computer science from KAIST, Republic of Korea, in 1990, 1992, and 1997, respectively. From 1997 to 2000, he was a Senior Researcher with LG Electronics. From 2000 to 2002, he was the Director with SL2 Ltd. Since 2002, he has been a Professor with the Computer Science Department, University of Seoul. His research interests include speech and speaker recognition, acoustic scene classification, and machine learning. He is an Editor of the *Journal of the Korean Society of Speech Sciences*.