VARIATIONAL MANIFOLD LEARNING FOR SPEAKER RECOGNITION

Jen-Tzung Chien and Cheng-Wei Hsu

Department of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan

ABSTRACT

This paper presents a variational manifold learning for speaker recognition based on the probabilistic linear discriminant analysis (PLDA) using i-vectors. A latent variable model is introduced to compensate the constraints of the linearity in PLDA scoring and the high dimensionality in using i-vectors. A *deep variational learning* is formulated to jointly optimize three objectives including a regularization for variational distributions, a reconstruction based on PLDA and a manifold learning for neighbor embedding. A stochastic gradient variational Bayesian algorithm is developed to optimize the variational lower bound of log likelihood where the expectation in the objectives is estimated via a *sampling* method. Interestingly, the latent variables in the proposed variational manifold PLDA (vm-PLDA) are capable of decoding or reconstructing the i-vectors. The experiments on visualization and speaker recognition show the merits of vm-PLDA in manifold learning and classification.

Index Terms— Probabilistic linear discriminant analysis, deep learning, variational manifold learning, speaker recognition

1. INTRODUCTION

Speaker recognition system using the i-vectors [1] as the speaker features and the probabilistic linear discriminant analysis (PLDA) [2] as the scoring function has achieved state-of-the-art performance in many tasks. PLDA is seen as a linear model which is trained under the assumption that the same speaker shares a common low dimensional latent variable space where i-vectors of all speakers are represented in this space. No discriminative learning is explicitly performed. PLDA is estimated according to the expectation-maximization (EM) algorithm [3] by maximizing the likelihood using a whole set of training data. Basically, such a speaker model may be constrained due to the linearity assumption, shallow representation, high dimensionality, non-discriminative and batch learning. This study focuses on a deep manifold neural network and deals with these constraints. A deep latent variable model [4] based on the variational auto-encoder [5] is incorporated to conduct the discriminative manifold learning [6] and scoring. A variational manifold PLDA (vm-PLDA) is proposed for speaker recognition.

In general, manifold learning aims to learn a low-dimensional representation from its high-dimensional observation data, e.g. i-vector, where the objective for neighbor embedding is optimized. Speaker label can be introduced to enforce those observations in low-dimensional space to be close within the same speaker and apart across different speakers. To further strengthen the system performance, such a supervised manifold learning can be realized as a *deep latent variable model* due to twofold considerations. First, deep neural network is used to reflect the complicated characteristics within speakers and between speakers. Secondly, a latent variable model is considered to explore the latent structure and compensate the uncertainty region of a deep model via a *stochastic* back-propagation algorithm from the mini-batches of speaker utterances. In this study, we develop a deep variational manifold learning for speaker recognition. The variational inference is implemented to carry out a latent variable model for speaker recognition which tightly integrates the stochastic neighbor embedding (SNE) [7, 8] for dimensionality reduction and the PLDA scoring for speaker recognition. The variational lower bound of log likelihood, consisting of such two objectives or considerations, is jointly optimized. A stochastic gradient variational Bayesian (SGVB) [5, 9] algorithm is developed for inference of the resulting vm-PLDA. The proposed method is evaluated by the experiments on data visualization and speaker recognition.

2. BACKGROUND SURVEY

2.1. Manifold learning

SNE was developed as a nonlinear unsupervised manifold learning [7]. Suppose we are given a set of high-dimensional data $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. SNE attempts to find the low-dimensional representations $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ where $\mathbf{z}_n \in \mathbb{R}^d$ preserves the pairwise similarity to $\mathbf{x}_n \in \mathbb{R}^D$ and $d < D$. The joint probability p_{nm} of two samples $\mathbf{x}_n$ and $\mathbf{x}_m$ is expressed by a Gaussian distribution

$$p_{nm} = \frac{\exp\left(-\|\mathbf{x}_n - \mathbf{x}_m\|^2\right)}{\sum_s \sum_{t \neq s} \exp\left(-\|\mathbf{x}_s - \mathbf{x}_t\|^2\right)}. \tag{1}$$

The joint probability in low-dimensional representation q_{nm} can be also modeled by a Gaussian using the pairwise similarity $\mathbf{z}_n - \mathbf{z}_m$. In [10], a symmetric SNE was implemented by minimizing the Kullback-Leibler divergence $\sum_n \mathcal{D}_{\text{KL}}(P_n \| Q_n) = \sum_n \sum_m p_{nm} \log\left(\frac{p_{nm}}{q_{nm}}\right)$ between two sets of probability distributions $P_n = \{p_{nm}\}_{n=1}^N$ and $Q_n = \{q_{nm}\}_{n=1}^N$. Neighbor embedding of samples in two spaces is naturally preserved with this nonlinear and nonparametric transformation. In [8], the t-distributed SNE (t-SNE) was proposed by adopting the joint distribution for two low-dimensional samples $\mathbf{z}_n$ and $\mathbf{z}_m$ based on a Student's t-distribution

$$q_{nm} = \frac{\left(1 + \|\mathbf{z}_n - \mathbf{z}_m\|^2/\nu\right)^{-\frac{\nu+1}{2}}}{\sum_s \sum_{t \neq s} \left(1 + \|\mathbf{z}_s - \mathbf{z}_t\|^2/\nu\right)^{-\frac{\nu+1}{2}}} \tag{2}$$

where ν denotes the degree of freedom. The *crowding* problem in conventional SNE [7] model can be alleviated by using t-SNE where the low-dimensional representations are presented to be mutually close together. As a result, the front-end processing of finding subspace features for i-vectors using t-SNE can reduce the "curse of dimensionality" for speaker recognition based on the PLDA scoring function [6].

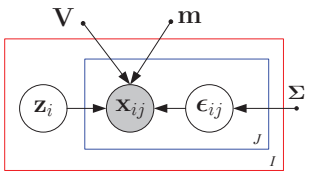


Fig. 1. Graphical model for PLDA.

2.2. Probabilistic linear discriminant analysis

PLDA [2] is a generative model which characterizes both variations from within-individuals and between-individuals via

$$\mathbf{x}_{ij} = \mathbf{m} + \mathbf{V}\mathbf{z}_i + \boldsymbol{\epsilon}_{ij} \tag{3}$$

where feature vector or i-vector $\mathbf{x}_{ij} \in \mathcal{R}^D$ of speaker i in a session j is represented by factor analysis (FA) with a mean vector $\mathbf{m}$, a factor loading matrix $\mathbf{V} \in \mathcal{R}^{D\times d}$, a common vector $\mathbf{z}_i \in \mathcal{R}^d$ and a residual vector $\boldsymbol{\epsilon}_{ij}$. FA assumes $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{\epsilon}_{ij} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with $d \times d$ identity covariance matrix $\mathbf{I}$ and $D \times D$ covariance matrix $\boldsymbol{\Sigma}$, respectively [11]. Figure 1 depicts the graphical representation for PLDA. PLDA parameters $\boldsymbol{\theta} = \{\mathbf{m}, \mathbf{V}, \boldsymbol{\Sigma}\}$ are estimated by maximizing the summation of log likelihood function over individual i-vectors

$$\begin{aligned} p(\mathbf{x}_{ij}|\boldsymbol{\theta}) &= \int \mathcal{N}(\mathbf{x}_{ij}|\mathbf{m} + \mathbf{V}\mathbf{z}_i, \boldsymbol{\Sigma})\mathcal{N}(\mathbf{z}_i|\mathbf{0}, \mathbf{I})d\mathbf{z}_i \\ &= \mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}, \mathbf{V}\mathbf{V}^\top + \boldsymbol{\Sigma}) \end{aligned} \tag{4}$$

according to the EM algorithm [3]. E-step is to calculate the posterior probability $p(\mathbf{z}_i|\mathcal{X}, \boldsymbol{\theta}^{\text{old}})$ due to latent vector $\mathbf{z}_i$ by using the training i-vectors $\mathcal{X} = \{\mathbf{x}_{ij}\}$ given old parameter estimate $\boldsymbol{\theta}^{\text{old}}$. Using this posterior probability, M-step is to estimate new PLDA parameters $\boldsymbol{\theta}^{\text{new}}$ by maximizing an auxiliary function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{\text{old}})$. This function is also viewed as the lower bound $\mathcal{L}(q, \boldsymbol{\theta})$ of the log likelihood function $p(\mathcal{X}|\boldsymbol{\theta})$ where a variational or approximate distribution $q(\mathbf{z}_i) = p(\mathbf{z}_i|\mathcal{X}, \boldsymbol{\theta}^{\text{old}})$ from E-step is merged [12]. Iterative EM steps are executed to find the converged PLDA parameters $\boldsymbol{\theta}$. In test phase, PLDA scoring is performed for speaker verification according to a *likelihood ratio test* whether a test speaker's i-vector $\mathbf{x}_s$ and target speaker's i-vector $\mathbf{x}_t$ are from the same speaker or not. The joint Gaussian distributions and individual Gaussian distributions for $\mathbf{x}_s$ and $\mathbf{x}_t$ are calculated under null hypothesis and alternative hypothesis based on PLDA, respectively. PLDA is recognized as a powerful approach to speaker recognition with different extension [13, 14].

3. VARIATIONAL MANIFOLD PLDA

This work carries out a discriminative model which conducts the supervised manifold learning of i-vectors for PLDA scoring under a deep variational learning framework.

3.1. Variational manifold learning

Variational manifold learning is performed by implementing the manifold learning in accordance with the variational auto-encoder [5, 15] which consists of an encoder for recognition model $q_\phi(\mathbf{z}|\mathbf{x})$ and a decoder for generative model $p_\theta(\mathbf{x}|\mathbf{z})$ as illustrated in Figure 2. Encoder is used to encode or transform an i-vector $\mathbf{x}_n$ into a latent low-dimensional representation $\mathbf{z}_n$ via a variational distribution given by a Gaussian

$$\mathbf{z}_n \sim q_\phi(\mathbf{z}_n|\mathbf{x}_n) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}(\mathbf{x}_n), \mathbf{C}(\mathbf{x}_n)) \tag{5}$$

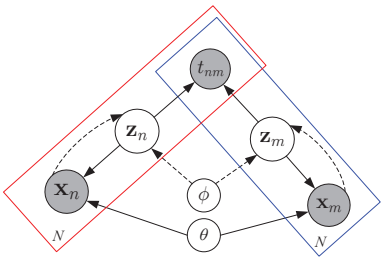


Fig. 2. Graphical model for variational manifold learning. Solid line denotes the generative model $p_\theta(\mathbf{x}|\mathbf{z})$ while dashed line denotes the recognition model using variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$.

where the mean vector $\boldsymbol{\mu}$ and the diagonal covariance matrix $\mathbf{C} = \text{diag}\{\sigma_j^2\}$ are calculated by a mapping function using the deep neural network (DNN) with weight parameters $\mathbf{W}$, i.e. $\boldsymbol{\phi} = \{\boldsymbol{\mu}(\mathbf{x}_n), \mathbf{C}(\mathbf{x}_n)\}$ or equivalently $\boldsymbol{\phi} = \mathbf{W}$ as shown in Figure 3. By sampling the latent features $\mathbf{z}$ using the estimated $q_\phi(\mathbf{z}_n|\mathbf{x}_n)$, we can reconstruct the original i-vector $\hat{\mathbf{x}}_n$. The stochastic property in latent features $\mathbf{z}$ are sufficiently reflected for estimating the PLDA parameters $\boldsymbol{\theta} = \{\mathbf{m}, \mathbf{V}, \boldsymbol{\Sigma}\}$. Importantly, the supervised manifold learning is introduced to learn low-dimensional variables $\mathbf{z}_n$ and $\mathbf{z}_m$ for i-vectors $\mathbf{x}_n$ and $\mathbf{x}_m$. The objective of neighbor embedding is optimized by using the collected class targets t_n and t_m. Considering the assumption of PLDA that i-vectors of the same speaker share the same latent variable, we avoid the explicit probability model in Eq. (1) by defining $p_{nn} = 0$ and $p_{nm} = 1$ for the case $t_n = t_m$ and $p_{nm} = 0$ for the case $t_n \neq t_m$. These pre-assigned probabilities $P = \{p_{nm}\}$ are seen as the desired values for latent variables $\mathbf{z}_n$ and $\mathbf{z}_m$. Such a supervision is seen as the observed Bernoulli target $t_{nm} \triangleq p_{nm}$ for supervised learning. In this study, the neighboring probability q_{nm} in low-dimensional space is characterized by t distribution in Eq. (2). We construct a hybrid learning objective for variational parameters $\boldsymbol{\phi}$ and model parameters $\boldsymbol{\theta}$ which fulfills the manifold learning and PLDA scoring, respectively.

3.2. Learning objective

The learning objective in variational manifold PLDA (vm-PLDA) is formed by log likelihood function $\log p(\mathcal{X}, \mathcal{T})$ of training data consisting of i-vectors $\mathcal{X} = \{\mathbf{x}_n\}$ and pairwise class targets or adjacency matrix $\mathcal{T} = \{t_{nm}\}$. This function is expressed by integrating out the pairwise latent variables $\mathcal{Z} = \{\mathbf{z}_n, \mathbf{z}_m\}$ as follows

$$\begin{aligned} &\log \int \prod_m \prod_n p(t_{nm}|\mathbf{z}_n, \mathbf{z}_m)p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n)d\mathcal{Z} \\ &\geq \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) \triangleq \sum_m \Bigg[\sum_n \underbrace{\mathbb{E}_{q_n}\mathbb{E}_{q_m}[\log p(t_{nm}|\mathbf{z}_n, \mathbf{z}_m)]}_{\text{manifold learning}} \\ &+ \underbrace{\mathbb{E}_{q_n}[\log p_\theta(\mathbf{x}_n|\mathbf{z}_n)]}_{\text{PLDA scoring}} - \underbrace{\mathcal{D}_{\text{KL}}(q_\phi(\mathbf{z}_n|\mathbf{x}_n)\|p(\mathbf{z}_n))}_{\text{model regularization}} \Bigg]. \end{aligned} \tag{6}$$

The variational lower bound $\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta})$ is obtained by applying Jensen's inequality given by the variational distribution $q_\phi(\mathbf{z}_n|\mathbf{x}_n)$. This bound consists of three objectives as seen in Eq. (6). The first term involves the manifold learning for low-dimensional features $\mathbf{z}_n$ and $\mathbf{z}_m$ which is calculated as the expectation of log Bernoulli distribution over $\mathcal{Z}$ through sampling $\mathbf{z}_{n,l}$ for $\mathbf{z}_n$ and $\mathbf{z}_{m,s}$ for $\mathbf{z}_m$

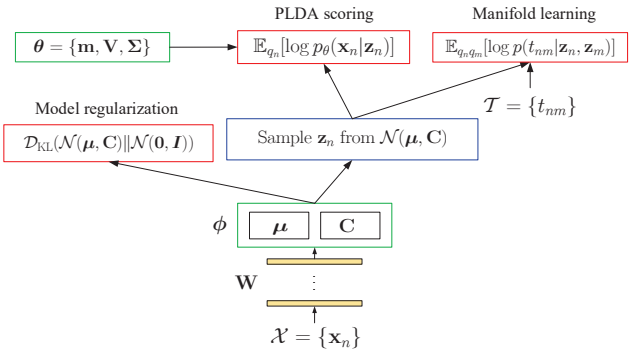


Fig. 3. System flow for constructing the objectives for vm-PLDA.

via the variational distribution $q_\phi(\mathbf{z}_n|\mathbf{x}_n)$ in Eq. (5). We have

$$\sum_n \sum_m \sum_l \sum_s \Bigg[-t_{nm} \frac{\nu+1}{2} \log\left(1 + \|\mathbf{z}_{n,l} - \mathbf{z}_{m,s}\|^2/\nu\right) + (1 - t_{nm}) \log\left(1 - \left(1 + \|\mathbf{z}_{n,l} - \mathbf{z}_{m,s}\|^2/\nu\right)^{-\frac{\nu+1}{2}}\right)\Bigg]. \tag{7}$$

This term is typically proportional to the KL divergence between $P = \{t_{nm}\}$ and $Q = \{q_{nm}\}$ in high and low-dimensional spaces, respectively. t-SNE is realized in the mv-PLDA. The second term is to measure the expectation of log likelihood which is again calculated by using the samples $\mathbf{z}_{n,l}$ and $\mathbf{z}_{m,s}$

$$-\frac{1}{2} \sum_n \sum_l \Big[\log|2\pi\boldsymbol{\Sigma}| + (\mathbf{x}_n - \mathbf{m} - \mathbf{V}\mathbf{z}_{n,l})^\top \times \boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \mathbf{m} - \mathbf{V}\mathbf{z}_{n,l}) \Big] \tag{8}$$

where $\boldsymbol{\theta} = \{\mathbf{m}, \mathbf{V}, \boldsymbol{\Sigma}\}$ denotes the PLDA parameters. The third term is a regularization term which regularizes the sampling distribution $q_\phi(\mathbf{z}_n|\mathbf{x}_n)$ for latent variable $\mathbf{z}_n$ to the prior distribution $p(\mathbf{z}_n)$ for PLDA based on a KL divergence

$$\mathcal{D}_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})\|\mathcal{N}(\mathbf{0}, \mathbf{I})) = \frac{1}{2} \sum_{j=1}^{d} \left[\mu_j^2 + \sigma_j^2 - \log\sigma_j^2 - 1\right] \tag{9}$$

where the variational parameters $\boldsymbol{\phi} = \{\boldsymbol{\mu} = \{\mu_j\}, \mathbf{C} = \mathrm{diag}\{\sigma_j^2\}\}$ correspond to the parameters $\boldsymbol{\phi} = \{\mathbf{W}\}$ in DNN mapping functions using input $\mathbf{x}_n$. Therefore, mv-PLDA is inferred by jointly optimizing a lower bound $\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta})$ containing three objectives.

3.3. Implementation and comparison

Figure 3 shows the forward pass in vm-PLDA which calculates the variational lower bound of log likelihood of the observed i-vectors and adjacency matrix $\log p(\mathcal{X}, \mathcal{T})$. Maximizing this lower bound turns out to carry out the t-SNE for dimensionality reduction of i-vectors and the PLDA scoring for latent feature representation of target speakers. At the same time, the encoder of low-dimensional features $\mathbf{z}_n$ using $q_\phi(\mathbf{z}_n|\mathbf{x}_n)$ is regularized to meet PLDA assumption. In the maximization, the gradient of these three objectives with respect to encoder parameters $\boldsymbol{\phi} = \{\boldsymbol{\mu}(\mathbf{x}_n), \mathbf{C}(\mathbf{x}_n)\}$ (or DNN parameters $\boldsymbol{\phi} = \mathbf{W}$) and decoder parameters $\boldsymbol{\theta} = \{\mathbf{m}, \mathbf{V}, \boldsymbol{\Sigma}\}$ are calculated for supervised learning of vm-PLDA. A *stochastic* back propagation algorithm is developed to learn $\mathbf{W}$ or the randomness of speaker characteristics in PLDA model. Importantly, such an inference procedure is implemented through the Monte Carlo estimate for the expectation functions in the objectives for neighbor embedding and PLDA scoring. The stochastic learning using mini-batches of i-vectors is performed in vm-PLDA while standard PLDA runs the batch training.

In this study, we apply the re-parameterization trick to implement SGVB algorithm [5] for vm-PLDA where the sampling of speaker features $\mathbf{z}_{n,l}$ is performed by re-parameterizing $\mathbf{z}_{n,l} \sim q_\phi(\mathbf{z}_n|\mathbf{x}_n)$ as $\mathbf{z}_{n,l} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_{n,l}$ where $\boldsymbol{\sigma} = \{\sigma_j\}$ and $\boldsymbol{\epsilon}_{n,l} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Sampling $\mathbf{z}_{n,l}$ is then performed through indirectly sampling $\boldsymbol{\epsilon}_{n,l}$. The problem of high variance in sampling process can be alleviated [5]. The samples of speaker features $\{\mathbf{z}_{n,l}\}_{l=1}^L$ are used to calculate the expectations in Eqs. (7)-(8). The gradients are accordingly obtained for inference of vm-PLDA.

In general, the inference via encoder and decoder in vm-PLDA is comparable with the E-step and M-step in PLDA, respectively. Encoder aims to find a posterior distribution given by variational parameters $\boldsymbol{\phi}$ while decoder reconstructs the i-vectors using model parameters $\boldsymbol{\theta}$ estimated by M-step. In addition, PLDA assumes a low-dimensional common vector $\mathbf{z}$ with the prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ but without explicitly estimating speaker factors $\mathbf{z}$. The proposed vm-PLDA is trained and regularized by fitting this assumption and running the manifold learning for finding latent features $\mathbf{z}_n$ for each i-vector $\mathbf{x}_n$. This low-dimensional representation is distributed by $\mathcal{N}(\boldsymbol{\mu}(\mathbf{x}_n), \mathbf{C}(\mathbf{x}_n))$ with the Gaussian parameters driven by DNN weights $\mathbf{W}$ and dependent on each i-vector $\mathbf{x}_n$.

Further, the speaker features $\mathbf{z}_n$ in vm-PLDA can be compared with those features extracted by the supervised t-SNE using deep model as proposed in [16]. Using the deep manifold (dm) [16] combined with the cosine scoring for speaker recognition is herein named as the *dm-Cosine*. Different from dm-Cosine using deterministic features, the so-called vm-Cosine applies the cosine scoring and conducts the variational learning to characterize stochastic speaker features according to $q_\phi(\mathbf{z}|\mathbf{x})$. Also, vm-Cosine is driven and tightly coupled with PLDA while dm-Cosine is constructed by separating t-SNE and scoring function.

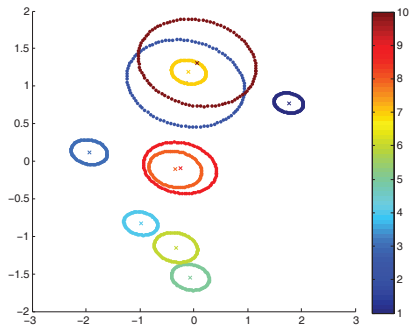
4. EXPERIMENTS

4.1. Experimental setup

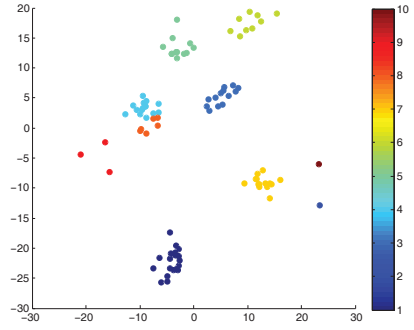
We conducted the visualization of i-vectors and the generalization of various PLDAs for speaker recognition where the equal error rate (EER) (%) was examined. We followed the experimental setup for NIST i-vector Speaker Recognition Challenge in [17]. The number of i-vectors in development set was 36,572 from 1930 males and 3028 females. The number of target speaker models was 1,306 with totally 6,530 i-vectors. The number of test i-vectors was 9,634. There were 12,582,004 trials which included all possible pairs involving a target speaker model and a single i-vector test segment. These trials were divided into a progress subset with 40% of the trials and an evaluation subset with the remaining 60% of the trials. The minimum decision cost function (minDCF) [17] was measured for comparison of different methods in two subsets.

For comparison, we carried out the PLDA [2], the dm-PLDA [16] and the proposed vm-PLDA. In dm-PLDA and vm-PLDA, the topology D-500-500-d with two 500-neuron hidden layers was adopted for manifold learning where D=600. The pre-training using the restricted Boltzmann machine and the Adam algorithm [18] with a mini-batch size of 500 were applied. Using vm-PLDA, the

initial learning rate was set to 0.001 for encoding parameters ϕ and 0.00005 for decoding parameters $\boldsymbol{\theta}$. The normalized initialization of weight parameters was performed. The i-vector length normalization was applied.



(a) PLDA



(b) vm-PLDA

Fig. 4. Two-dimensional visualizations for ten speakers using PLDA and vm-PLDA.

4.2. Experimental results

Figure 4 shows the visualization of i-vectors by using PLDA and vm-PLDA where $d = 2$ was considered. The utterances from ten speakers were selected and shown in different colors. In case of PLDA, we plot the mean and 95% confidence level using the Gaussian we estimated from the samples of the same speakers using the statistics in E-step of final EM iteration. Larger ellipse means larger variance which was basically caused by fewer samples. The features from different speakers are confusing in PLDA while vm-PLDA can generate the well distributed and separated speaker features in a wider data range. It is because that vm-PLDA conducts the distribution learning and imposes a repulsion force in the objective for separating different speakers in manifold learning.

Figure 5 illustrates the prediction capability of using dm-Cosine and vm-Cosine in case of $d = 50$. We demonstrate the EERs of test i-vectors by using different features in reduced dimension which are estimated in different learning epochs. It is shown that dm-Cosine does not generalize and converge well when compared with vm-Cosine. The features using the proposed variational manifold learning obtains a desirable performance in stochastic learning.

Table 1 reports the EER and minDCF in progress subset and evaluation subset by using different manifold learning methods

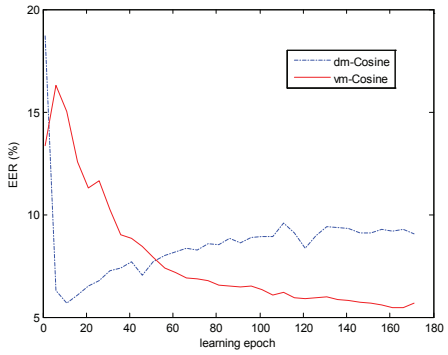


Fig. 5. EER (%) of test i-vectors versus training epochs using dm-Cosine and vm-Cosine.

Method (d)	EER	minDCF (prog)	minDCF (eval)
dm-Cosine (20)	6.1087	0.7151	0.6812
dm-Cosine (100)	3.6331	0.3661	0.3176
vm-Cosine (20)	4.7831	0.5984	0.5484
vm-Cosine (100)	3.4004	0.3406	0.2968
PLDA (20)	5.4028	0.6137	0.5862
PLDA (50)	3.4595	0.4013	0.3644
PLDA (75)	3.2878	0.3655	0.3210
PLDA (100)	3.2007	0.3591	0.3119
vm-PLDA (20)	4.5732	0.5380	0.5056
vm-PLDA (50)	3.3275	0.3962	0.3510
vm-PLDA (75)	3.1392	0.3514	0.3186
vm-PLDA (100)	2.9395	0.3314	0.2947

Table 1. Comparison of EER (%) and minDCF in two subsets using different methods.

(deep manifold - dm [16] and variational manifold - vm) and different scoring functions (Cosine and PLDA) under different reduced dimensions d. We find that PLDA scoring performs much better than cosine scoring. The proposed variational manifold learning outperforms the deep manifold learning in [16]. The vm-PLDA obtains lower EER and minDCF compared with PLDA. The larger the dimension d is selected, the better the speaker recognition is achieved but with high computation cost.

5. CONCLUSIONS

We have presented a variational manifolding learning for i-vector based PLDA scoring and speaker recognition. The means and variances of latent factors in this new PLDA, represented by DNN, were trained by maximizing the lower bound of log likelihood which guided the optimization to accomplish the t distributed SNE for subspace learning and dimensionality reduction. A shared neural network for different speakers was established but its outputs depended on the observed i-vectors. In particular, we introduced a binary variable to indicate the class information for each pair of i-vectors and used this latent variable to express the attraction and the repulsion for those low-dimensional samples within the same speaker and between two different speakers, respectively. We correspondingly built a t-SNE approach by using a neural network as encoder and a PLDA as decoder. A hybrid generative and discriminative model was constructed for deep manifold learning. This approach performed better than deterministic deep model for manifold learning and baseline PLDA for speaker recognition.

6. REFERENCES

[1] N. Dehak, P. J. 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, pp. 788–798, 2011.

[2] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. of IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977.

[4] C.-H. Lee and J.-T. Chien, "Deep unfolding inference for supervised topic model," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 2279–2283.

[5] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. of International Conference on Learning Representation*, 2014.

[6] J.-T. Chien and C.-H. Chen, "Deep discriminative manifold learning," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 2672–2676.

[7] G. E. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds., 2003, pp. 857–864.

[8] L. van der Maaten and G. E. Hinton, "Visualizing data using *t*-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[9] S. Watanabe and J.-T. Chien, *Bayesian Speech and Language Processing*, Cambridge University Press, 2015.

[10] J. Cook, I. Sutskever, A. Mnih, and G. E. Hinton, "Visualizing similarity data with a mixture of maps," in *Proc. of International Conference on Artificial Intelligence and Statistics*, 2007, pp. 67–74.

[11] J.-T. Chien and C.-W. Ting, "Factor analyzed subspace modeling and selection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 239–248, 2008.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer Science, 2006.

[13] M.-W. Mak, X. Pang, and J.-T. Chien, "Mixture of PLDA for noise robust i-vector speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 130–142, 2016.

[14] N. Li, M.-W. Mak, and J.-T. Chien, "Deep neural network supervised mixture of PLDA for robust speaker verification," in *Proc. of IEEE Workshop on Spoken Language Technology*, 2016.

[15] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. of International Conference on Machine Learning*, 2014.

[16] M. R. Min, L. Maaten, Z. Yuan, A. J. Bonner, and Z. Zhang, "Deep supervised *t*-distributed embedding," in *Proc. of International Conference on Machine Learning*, 2010, pp. 791–798.

[17] C. S. Greenberg, D. Bansé, G. R. Doddington, D. Garcia-Romero, J. J. Godfrey, T. Kinnunen, A. F. Martin, A. McCree, M. Przybocki, and D. A. Reynolds, "The NIST 2014 speaker recognition i-vector machine learning challenge," in *Proc. of Odyssey: The Speaker and Language Recognition Workshop*, 2014.

[18] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proc. of International Conference for Learning Representations*, 2015.