Siamese Neural Network with Joint Bayesian Model Structure for Speaker Verification

Xugang Lu*, Peng Shen*, Yu Tsao†,Hisashi Kawai*
* National Institute of Information and Communications Technology, Japan
E-mail: xugang.lu@nict.go.jp
† Research Center for Information Technology Innovation, Academic Sinica, Taiwan

***Abstract*—Generative probability models are widely used for speaker verification (SV). However, the generative models are lack of discriminative feature selection ability. As a hypothesis test, the SV can be regarded as a binary classification task which can be designed as a Siamese neural network (SiamNN) with discriminative training. However, in most of the discriminative training for SiamNN, only the distribution of pair-wised sample distances is considered, and the additional discriminative information in joint distribution of samples is ignored. In this paper, we propose a novel SiamNN with consideration of the joint distribution of samples. The joint distribution of samples is first formulated based on a joint Bayesian (JB) based generative model, then a SiamNN is designed with dense layers to approximate the factorized affine transforms as used in the JB model. By initializing the SiamNN with the learned model parameters of the JB model, we further train the model parameters with the pair-wised samples as a binary discrimination task for SV. We carried out SV experiments on data corpus of speakers in the wild (SITW) and VoxCeleb. Experimental results showed that our proposed model improved the performance with a large margin compared with state of the art models for SV.**

I. Introduction

Speaker verification (SV) is a task to judge whether an utterance is spoken by a registered speaker or not, it is widely used in many speech application systems for authentic or security purpose [1], [2], [3]. The conventional pipeline in constructing a SV system is composed of a front-end speaker embedding feature extraction and a back-end speaker classifier modeling. The front-end embedding feature extraction tries to extract robust and discriminative speaker features, and back-end classifier tries to model speaker features based on which the similarity or distance between two compared features vectors could be estimated. In most state of the art frameworks, i-vector [4], [5], d-vector and X-vector [5], [6], have been proposed as front-end speaker features. Particularly, the X-vector as one of the speaker embedding representations is the most widely used one [5]. Since the original front-end feature encodes various of acoustic factors, e.g., speaker factor, channel transmission factor, recording device factor, etc., before classifier modeling, a linear discriminative analysis (LDA) or local fisher discriminative analysis [7] based dimension reduction is usually applied to eliminate non-speaker specific information. Based on the robust speaker features, several back-end speaker models have been proposed, for example, the probabilistic linear discriminant analysis (PLDA) modeling [4], [8], joint Bayesian (JB) modelling [9], [10], support vector machine (SVM) [13], as well as other types of discriminative classification based modeling [14] [15], [16].

The SV problem can be defined as a hypothesis test [17]:

$$\begin{aligned} H_S &: \mathbf{x}_i, \mathbf{x}_j \text{ are spoken by the same speaker} \\ H_D &: \mathbf{x}_i, \mathbf{x}_j \text{ are spoken by different speakers,} \end{aligned} \tag{1}$$

where H_S and H_D are the two hypothesises as the same and different speaker spaces, respectively. $(\mathbf{x}_i, \mathbf{x}_j)$ is a tuple with two compared utterances indexed by i and j (as a trial in SV tasks). In most of the SV algorithms, the hypothesis test defined in Eq. (1) is finally formulated as a log-likelihood ratio (LLR) function [17]. And usually the LLR is estimated based on generative probabilistic models in a transformed speaker feature space. However, the generative models are lack of discriminative feature selection ability, and usually a discriminative feature transform is independently applied before the generative classifier modeling. As an alternative, the hypothesis test in Eq. (1) can be regarded as a binary classification task where neural network based discriminative models could be applied. In most of these discriminative models, a distance metric could be learned for the hypothesis test defined in Eq. (1). In this distance metric learning, no probability distribution assumption (e.g., Gaussian for most generative models) is required, and the feature transformed space and hypothesis test model can be optimized in a unified neural network model. However, in most of these distance metric learning algorithms, only the distribution of the distances of pair-wised samples is considered without considering the joint distribution of samples. As indicated in a joint Bayesian (JB) analysis model, considering the joint distribution of samples could introduce additional discriminative information compared with only considering the distribution of distances of the pair-wised samples [9], [10]. In this paper, we propose a novel Siamese neural network (SiamNN) based discriminative framework for SV. Although SiamNN based with pair-wise samples [11], or neural network with triple-wise samples [12], have been proposed for speaker verification, our proposed SiamNN framework is different from their studies. In the framework, the SiamNN architecture is designed to integrate the model structure of JB, and jointly optimized with a discriminative feature learning process. Due to the discriminative learning property, a direct evaluation metric for SV is easily integrated as a learning objective function. Our experiments confirmed the advantages of the proposed

SiamNN framework.

II. The proposed discriminative neural network model

The hypothesis test defined in Eq. (1) can be regarded as a Bayesian binary classification task, and a discriminative neural network model can be designed for the task. During the architecture design of the neural network model, in order to take the model structure of a generative probabilistic model into consideration, we need to explain the conventional generative model based algorithms, and their connections to the discriminative framework via the LLR estimation.

A. Log-likelihood ratio function based on generative probabilistic models

Based on a generative probability model, a log-likelihood ratio (LLR) with consideration of intra-speaker and inter-speaker distances is defined as:

$$r_{i,j} = r(\mathbf{x}_i, \mathbf{x}_j) = \log \frac{p(\Delta_{i,j}|H_S)}{p(\Delta_{i,j}|H_D)}, \tag{2}$$

where $\Delta_{i,j} = \mathbf{x}_i - \mathbf{x}_j$ is the pair-wised sample distance. Based on the Gaussian density distribution assumptions of $p(.|H_S)$ and $p(.|H_D)$, the LLR can be estimated as:

$$r_{i,j} = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) \tag{3}$$

where $\mathbf{M} = -(\Sigma_{H_S}^{-1} - \Sigma_{H_D}^{-1})$ with Σ_{H_S} and Σ_{H_D} as the covariance matrices of the pair-wised distance space $\Delta_{i,j}$ for H_S and H_D conditions, respectively. We can see that Eq. (3) is in the same form as Mahalanobis distance metric except the negativity of the $\mathbf{M}$ [18].

From the definition in Eq. (2), we can see that the learned distance metric only considers the distribution of the pair-wised sample distance space [19]. For joint Bayesian (JB) probability distribution modeling, i.e., $p(\mathbf{x}_i, \mathbf{x}_j|H_S)$ or $p(\mathbf{x}_i, \mathbf{x}_j|H_D)$, the LLR is estimated as:

$$r_{i,j} = r(\mathbf{x}_i, \mathbf{x}_j) = \log \frac{p(\mathbf{x}_i, \mathbf{x}_j|H_S)}{p(\mathbf{x}_i, \mathbf{x}_j|H_D)} \tag{4}$$

In the JB modeling, the observed speaker feature variable $\mathbf{x}$ satisfies the following formulation as:

$$\mathbf{x} = \mathbf{u} + \mathbf{n}, \tag{5}$$

where $\mathbf{u}$ is a speaker identity vector variable, and $\mathbf{n}$ represents intra-speaker variation caused by noise. In verification, for given a trial with $\mathbf{x}_i$ and $\mathbf{x}_j$ generated from Eq. (5), with zero mean Gaussian assumption (with covariance matrix $\Sigma_{\mathbf{u}}$ and $\Sigma_{\mathbf{n}}$ for $\mathbf{u}$ and $\mathbf{n}$ variables, respectively), the two terms $p(\mathbf{x}_i, \mathbf{x}_j|H_S)$ and $p(\mathbf{x}_i, \mathbf{x}_j|H_D)$ defined in Eq. (4) satisfy zero-mean Gaussian with covariances as:

$$\begin{aligned} \mathbf{S}_{H_S} &= \begin{bmatrix} \Sigma_{\mathbf{u}} + \Sigma_{\mathbf{n}} & \Sigma_{\mathbf{u}} \\ \Sigma_{\mathbf{u}} & \Sigma_{\mathbf{u}} + \Sigma_{\mathbf{n}} \end{bmatrix} \\ \mathbf{S}_{H_D} &= \begin{bmatrix} \Sigma_{\mathbf{u}} + \Sigma_{\mathbf{n}} & \mathbf{0} \\ \mathbf{0} & \Sigma_{\mathbf{u}} + \Sigma_{\mathbf{n}} \end{bmatrix} \end{aligned} \tag{6}$$

Based on Eq. (6), the LLR defined in Eq. (4) could be calculated based on:

$$r(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{A} \mathbf{x}_i + \mathbf{x}_j^T \mathbf{A} \mathbf{x}_j - 2\mathbf{x}_i^T \mathbf{G} \mathbf{x}_j, \tag{7}$$

where

$$\begin{aligned} \mathbf{A} &= (\Sigma_{\mathbf{u}} + \Sigma_{\mathbf{n}})^{-1} - [(\Sigma_{\mathbf{u}} + \Sigma_{\mathbf{n}}) - \Sigma_{\mathbf{u}}(\Sigma_{\mathbf{u}} + \Sigma_{\mathbf{n}})^{-1}\Sigma_{\mathbf{u}}]^{-1} \\ \mathbf{G} &= -(2\Sigma_{\mathbf{u}} + \Sigma_{\mathbf{n}})^{-1}\Sigma_{\mathbf{u}}\Sigma_{\mathbf{n}}^{-1} \end{aligned} \tag{8}$$

Comparing Eqs. (3) and (7), we can see that if we set $\mathbf{A} = \mathbf{G} = \mathbf{M}$, the JB model based LLR degenerates to be the same form as the Mahalanobis distance metric (except the negativity of the matrix). In this sense, we can regard the LLR in Eq. (3) as a special case in JB model based estimation. Since the LLR in Eqs. (3) and (7) are based on probabilistic modeling with Gaussian distribution assumptions, their model parameters could be estimated using EM (or EM-like) learning algorithms [9], [10].

B. Connecting log-likelihood ratio in a neural network classification model

The LLR defined either in Eqs. (2) or (4) can be derived from a generative model based classification model. Given a training data set $\{(\mathbf{x}_i, y_i)\}_{i=1,2,...,N}, y_i \in \{1, 2, ..., K\}$ with $\mathbf{x}_i$ and y_i as data feature and label, K is the number of classes, the classification model is defined as:

$$p(y = k|\mathbf{x}) = \frac{p(\mathbf{x}|y = k)\, p(y = k)}{\sum_{j=1}^{K} p(\mathbf{x}|y = j)\, p(y = j)}. \tag{9}$$

And Eq. (9) is further cast to:

$$p(y = k|\mathbf{x}) = \frac{1}{1 + \sum_{j=1, j\neq k}^{K} \exp(-r_{k,j}(\mathbf{x}, \Theta))}, \tag{10}$$

where

$$r_{k,j}(\mathbf{x}, \Theta) \triangleq \log \frac{p(\mathbf{x}|y = k)\, p(y = k)}{p(\mathbf{x}|y = j)\, p(y = j)}, \tag{11}$$

is a LLR function based on the probabilistic model with Θ as a model parameter set. In a neural network based classification model, the classification is formulated as:

$$p(y = k|\mathbf{x}) = \frac{\exp(o_k)}{\sum_{j=1}^{K} \exp(o_j)}, \tag{12}$$

where a network mapping function $o_j = \phi_j(\mathbf{x}, \Theta)$ is defined as the output corresponding to the j-th class, and Θ is the neural network parameter set. And Eq. (12) is cast to:

$$p(y = k|\mathbf{x}) = \frac{1}{1 + \sum_{j=1, j\neq k}^{K} \exp(-h_{k,j}(\mathbf{x}, \Theta))}, \tag{13}$$

where

$$h_{k,j}(\mathbf{x}, \Theta) \triangleq \phi_k(\mathbf{x}, \Theta) - \phi_j(\mathbf{x}, \Theta). \tag{14}$$

Comparing Eqs. (13), (14) with (10), (11), we can see that $h_{k,j}(\mathbf{x}, \Theta)$ can be connected to the LLR in calculation in a pair-wised neural discriminative training.

C. Pair-wised discriminative training for LLR modeling

For convenience of formulation, we define a trial as a tuple $\mathbf{z}_{i,j} = (\mathbf{x}_i, \mathbf{x}_j)$, and the two hypothesis spaces are constructed from the two data sets as:

$$\begin{aligned} S &= \{\mathbf{z}_{i,j} = (\mathbf{x}_i, \mathbf{x}_j) \in H_S\} \\ D &= \{\mathbf{z}_{i,j} = (\mathbf{x}_i, \mathbf{x}_j) \in H_D\} \end{aligned} \tag{15}$$

Given a trial with two observation variables $\mathbf{z}_{i,j} = (\mathbf{x}_i, \mathbf{x}_j)$ (X-vectors in this study), the classification task is to estimate and compare $p(H_S|\mathbf{z}_{i,j})$ and $p(H_D|\mathbf{z}_{i,j})$. As a binary discriminative learning, the label is defined as:

$$y_{i,j} = \begin{cases} 1, \mathbf{z}_{i,j} \in H_S \\ 0, \mathbf{z}_{i,j} \in H_D \end{cases} \tag{16}$$

Based on discriminative neural network model with reference to Eqs. (13) and (14), the posterior probability is estimated based on:

$$p(y_{i,j}|\mathbf{z}_{i,j}) = \begin{cases} \frac{1}{1+\exp(-h_{H_S,H_D}(\mathbf{z}_{i,j},\Theta))}; \mathbf{z}_{i,j} \in H_S \\ 1 - \frac{1}{1+\exp(-h_{H_S,H_D}(\mathbf{z}_{i,j},\Theta))}; \mathbf{z}_{i,j} \in H_D \end{cases} \tag{17}$$

As we have revealed from Eqs. (10), (11), and (4), we replace the $h_{H_S,H_D}(\mathbf{z}_{i,j}, \Theta)$ with LLR function, and define a mapping as a logistic function with scaled parameters as [20], [21]:

$$f(r_{i,j}) \triangleq \frac{1}{1 + \exp(-(\alpha r_{i,j} + \beta))} \tag{18}$$

where $r_{i,j}$ is the LLR as defined in either Eq. (2) or (4), α and β are gain and bias factors used in the regression model. In Eq. (18), we integrate the LLR score estimated from the probabilistic model in a neural discriminative training framework. The probability estimation in Eq. (17) is cast to:

$$\hat{y}_{i,j} \triangleq p(y_{i,j}|\mathbf{z}_{i,j}) = \begin{cases} f(r_{i,j}); \mathbf{z}_{i,j} \in H_S \\ 1 - f(r_{i,j}); \mathbf{z}_{i,j} \in H_D \end{cases} \tag{19}$$

The model parameters can be learned based on optimizing binary classification accuracy. Under this framework, it is easy to directly incorporate the SV evaluation metric in the neural discriminative leaning. In this study, an empirical Bayes risk (EBR) based objective function is adopted with consideration of the false alarm and miss detections, which is widely used in hypothesis test tasks for SV[22], [24].

D. Integrating the generative probabilistic model structure in the discriminative neural network

We design a Siamese neural network (SiamNN) within a pair-wised discriminative learning framework for SV. In conventional pipeline for SV, a LDA is applied before the probabilistic modeling. Correspondingly, in the SiamNN, a dense layer is designed for fulfilling the function of LDA, and another dense layer is for fulfilling the transform functions used in Eqs. (3) and (7). For more specific, the transform matrix used in Eq. (3) is factorized as:

$$\mathbf{M} = -\mathbf{P}\mathbf{P}^T. \tag{20}$$

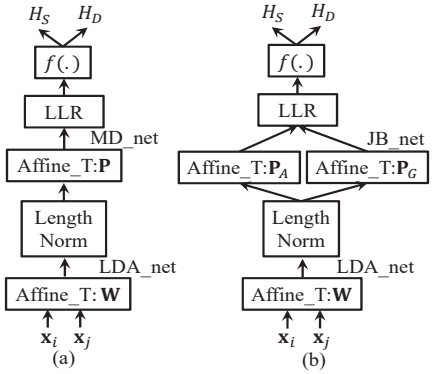

Fig. 1. The proposed SiamNN framework with: (a) MD_net (Mahalanobis net) with an affine transform matrix P, and (b) JB_net (joint Bayesian net) with two branches of affine transform matrices P_A and P_G. LDA_net with an affine transform matrix W for fulfilling the LDA transform.

And the matrices in Eq. (7) are factorized as:

$$\begin{aligned} \mathbf{A} &= -\mathbf{P}_A\mathbf{P}_A^T \\ \mathbf{G} &= -\mathbf{P}_G\mathbf{P}_G^T \end{aligned} \tag{21}$$

Based on the factorizations, the LLR function in Eq. (3) is cast to:

$$r_{i,j} = 2b_i^T b_j - b_i^T b_i - b_j^T b_j \tag{22}$$

with affine transforms as:

$$b_k = \mathbf{P}^T\tilde{\mathbf{h}}_k. \tag{23}$$

And the LLR function in Eq. (7) is cast to:

$$r_{i,j} = 2g_i^T g_j - a_i^T a_i - a_j^T a_j \tag{24}$$

with the affine transforms as:

$$\begin{aligned} a_i &= \mathbf{P}_A^T\tilde{\mathbf{h}}_i \\ g_i &= \mathbf{P}_G^T\tilde{\mathbf{h}}_i, \end{aligned} \tag{25}$$

where in Eqs. (23) and (25), $k \in \{i, j\}$, $\tilde{\mathbf{h}}_i = \frac{\mathbf{h}_i}{||\mathbf{h}_i||}$ is the length normalized vector from the LDA transform as:

$$\mathbf{h}_i = \mathbf{W}^T\mathbf{x}_i, \tag{26}$$

where $\mathbf{x}_i$ is the input X-vector feature, $\mathbf{W}$ is the transform in LDA. With these factorizations, the model architecture is designed as illustrated in Fig. 1. In this figure, "LDA_net" is for the LDA transform with the affine transform "Affine_T" defined in Eq. (26), "MD_net" is the Mahalanobis distance net with affine transform defined in Eq. (23), and "JB_net" is the JB network with two branches of affine transforms defined in Eq. (25). In training the SiamNN, we constructed "negative" and "positive" samples as we did in pair-wised discriminative training for language recognition task [23].

TABLE I
PERFORMANCE ON THE DEVELOPMENT SET OF SITW.

Methods	EER(%)	minDCF1	minDCF2
LDA+PLDA	3.00	0.332	0.520
LDA+JB	3.04	0.329	0.502
SiamNN (rand init)	4.16	0.379	0.588
SiamNN (JB init)	**2.66**	**0.297**	**0.447**

TABLE II
PERFORMANCE ON THE EVALUATION SET OF SITW.

Methods	EER (%)	minDCF1	minDCF2
LDA+PLDA	3.55	0.353	0.566
LDA+JB	3.50	0.342	0.565
SiamNN (rand init)	4.51	0.392	0.600
SiamNN (JB init)	**3.14**	**0.308**	**0.462**

TABLE III
PERFORMANCE BEFORE THE SIAMNN DISCRIMINATIVE TRAINING (WITH JB INIT) ON THE DEVELOPMENT SET OF SITW.

Methods	EER (%)	minDCF1	minDCF2
A (G=0)	47.71	1.00	1.00
G (A=0)	6.35	0.826	0.981
A, G (set G to A)	**3.12**	**0.360**	**0.584**
A, G (set A to G)	3.50	0.398	0.632

III. EXPERIMENTS AND RESULTS

A. Experimental conditions

We carried out SV experiments to test our proposed framework. The training data set is from VoxCeleb data corpus (sets 1 and 2) [25], and the test data sets are from the data corpus of speakers in the wild (SITW) [24]. We adopt a state of the art pipeline for constructing the SV baseline systems. The input speaker feature in our pipeline is X-vector which is extracted from a well trained deep time delay neural network (TDNN) neural network [5]. In training the TDNN model, the training data includes two data sets from Voxceleb corpus, i.e., the training set of Voxceleb1 corpus by removing overlapped speakers which are included in the test set of the SITW, and the training set of Voxceleb2. Moreover, data augmentation is applied to increase data diversity in TDNN model training. Input features for training the speaker embedding model are 30 Mel band bins based MFCCs with 25 ms frame length and 10 ms frame shift. The final extracted X-vector is with 512 dimensions. For removing non-speaker information, the LDA is applied to transform the 512-dimension X-vectors to 200-dimension vectors before the probabilistic modeling. Correspondingly, in the discriminative neural network model as showed in Fig. 1, a dense layer with 200 neurons is also applied in the "LDA_net".

Since the discriminative neural network architecture fits well to the conventional pipeline based on the probabilistic model structure, the dense layer parameters could be initialized with the conventional model parameters in training (according to Eqs. (26) and (25)). For comparison, the random parameter initialization method is also examined. In model training, the Adam algorithm with an initial learning rate of 0.0005 [26] was used. In order to include enough "negative" and "positive" samples, the mini-batch size was set to 4096. The training X-vectors were splitted to training and validation sets with a ratio of $9:1$. The model parameters were selected based on the best performance on the validation set.

B. Results

Two testing data sets from the SITW, i.e., development and evaluation sets are used, and each is used as an independent test set. The evaluation metrics, equal error rate (EER) and minimum decision cost function (minDCF) (with target prior 0.01 denoted as minDCF1, and prior 0.001 denoted as minDCF2) are adopted to measure the performance [22], [24]. The results are showed in tables (I) and (II). In these two tables, "LDA+PLDA" and "LDA+JB" represent the baseline systems based on probabilistic models PLDA and JB, respectively. "SiamNN" denotes the proposed system which takes the probabilistic JB model structure in designing the neural network model (as illustrated in (b) of Fig. 1), and model parameters are with random initialization ("SiamNN (rand init)") or with EM algorithm learned JB model parameters ("SiamNN (JB init)"). From these two tables, we can see that the performance of the baseline system with probabilistic JB model is comparable or a slight better than that of the PLDA based model. In the SiamNN based model, if model parameters are randomly initialized ("SiamNN (rand init)"), the performance is worse than the original baseline model based results. However, when the SiamNN parameters are initialized with the JB based baseline model parameters, the performance is significantly improved. These results indicate that the discriminative training could further enhance the discriminative power of the conventional JB based probabilistic model.

C. Effect of $\mathbf{A}$ and $\mathbf{G}$ on SV performance

In our SiamNN discriminative training, the LLR of the JB model defined in Eq. (7) is integrated. With different settings of $\mathbf{A}$ and $\mathbf{G}$ in Eq. (7), we could obtain:

$$r(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} -2\mathbf{x}_i^T \mathbf{G} \mathbf{x}_j; \text{ for } \mathbf{A} = 0 \\ \mathbf{x}_i^T \mathbf{A} \mathbf{x}_i + \mathbf{x}_j^T \mathbf{A} \mathbf{x}_j; \text{ for } \mathbf{G} = 0 \\ (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{G} (\mathbf{x}_i - \mathbf{x}_j); \text{ for } \mathbf{A} = \mathbf{G} \\ (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{A} (\mathbf{x}_i - \mathbf{x}_j); \text{ for } \mathbf{G} = \mathbf{A} \end{cases} \quad (27)$$

TABLE IV
PERFORMANCE AFTER THE SIAMNN DISCRIMINATIVE TRAINING (WITH JB INIT) ON THE DEVELOPMENT SET OF SITW.

Methods	EER (%)	minDCF1	minDCF2
A (G=0)	50.29	1.000	1.000
G (A=0)	4.78	0.421	0.634
A, G (set G to A)	**2.81**	**0.298**	0.456
A, G (set A to G)	3.08	0.313	**0.451**

TABLE V
PERFORMANCE OF THE SIAMNN DISCRIMINATIVE TRAINING WITH "MD_NET" ON THE DEVELOPMENT SET OF SITW

Methods	EER (%)	minDCF1	minDCF2
Random init $\mathbf{P}$	3.97	0.374	0.554
Init $\mathbf{P}$ with $\mathbf{P}_A$	**3.62**	**0.369**	**0.547**
Init $\mathbf{P}$ with $\mathbf{P}_G$	4.01	0.406	0.600

Based on this formulation, the two matrices $\mathbf{A}$ and $\mathbf{G}$ are connected to the two dense layers of the SiamNN model with weights $\mathbf{P}_A$ and $\mathbf{P}_G$ (refer to Fig. 1). In our model, the dense layers were first initialized with the parameters from the learned JB based baseline model, then the model was further trained with "negative" and "positive" pair-wised samples. Only in testing stage, the different parameter settings according to Eq. (27) are examined for experiments, and the results are showed in tables III and IV for the dev set of SITW. In these two tables, by comparing conditions with $\mathbf{A} = 0$ or $\mathbf{G} = 0$, we can see that the cross term contributes more to the SV performance, i.e., the dense layer with neural weight $\mathbf{P}_G$ contributes the most discriminative information in the SV task. Moreover, when keeping the cross term either by setting $\mathbf{A} = \mathbf{G}$ or $\mathbf{G} = \mathbf{A}$, the performance is better than setting any one of them to be zero. As a special case of the JB model based SiamNN, we also test the "MD_net" on dev set of SITW with different settings, and show the results in table V. From this table, we can see that when the model parameters are initialized with the $\mathbf{P}_A$ parameters, the performance is the best for this "MD_net" based model. However, no matter in what conditions, comparing results in tables I and V, we can confirm that the model structure inspired by the JB model is the best when the model parameters are initialized properly.

IV. Discussion and conclusion

In this study, we regard SV problem as a Bayesian binary classification task, and propose a SiamNN discriminative learning framework with "positive" and "negative" sample pairs (as from the same and different speakers). Rather than only considering the distributions of pair-wised intra- and inter-speaker distances, the joint distribution of samples is taken into consideration via the formulation from JB based generative modelling. With the help of matrix factorization, we reformulate the LLR estimation of the JB model to a distance metric as used in the discriminative learning framework. In particular, the linear transform matrices in the JB model are implemented as dense layers of the neural network model hence the JB based model structure is effectively connected to the SiamNN framework. Moreover, the SiamNN framework takes the speaker feature transform and classification model parameters learning in a unified optimization framework. Our experiments confirmed that the SV was benefitted from the unified discriminative learning framework.

In a discriminative learning framework, many loss functions have been investigated for speaker recognition as a distance metric learning task [27]. Our study may be further extended to integrate those different distance metrics or losses with the specially designed network architecture. In this study, the network architecture was derived from JB based generative model with a simple probability distribution assumption, i.e., a single modal Gaussian distribution assumption of speaker features and noise. In real applications, the probability distributions are much more complex. Although it is difficult for a generative probabilistic model to fit complex probability distributions in a high dimensional space, it is relatively easy for a neural network learning framework to do it. In the future, we will consider model structures for dealing with more complex probability distributions in SV tasks.

Acknowledgment

The work is partially supported by JSPS KAKENHI No. 19K12035, No. 21K17776.

References

[1] J. Hansen, T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74-99, 2015.

[2] A. Poddar, M. Sahidullah, G. Saha, "Speaker Verification with Short Utterances: A Review of Challenges, Trends and Opportunities," *IET Biometrics*, 7 (2), pp. 91-101, 2018.

[3] H. Beigi, Fundamentals of Speaker Recognition, Springer-Verlag, Berlin, 2011, ISBN 978-0-387-77591-3.

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

[5] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329-5333, 2018.

[6] E. Variani, X. Lei, E. McDermott, I. L. Moreno and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4052-4056, 2014.

[7] P. Shen, X. Lu, L. Liu, H. Kawai, "Local fisher discriminant analysis for spoken language identification," in *Proc. of ICASSP*, pp. 5825-5829, 2016.

[8] S. Prince and J. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1-8, 2007.

[9] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, "Bayesian face revisited: A joint formulation," in *European Conference on Computer Vision*, pp. 566-579, 2012.

[10] D. Chen, X. Cao, D. Wipf, F. Wen, and J. Sun, "An efficient joint formulation for Bayesian face verification," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 39, pp. 32-46, 2016.

[11] U. Khan, and J. Hernando, "Unsupervised training of siamese networks for speaker verification," *Interspeech*, 2020.

[12] C. Zhang, K. Koishida, and J. H. Hansen, "Text-independent speaker verification based on triplet convolutional neural network embeddings," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1633-1644, 2018.

[13] V. Wan, W. Campbell, "Support vector machines for speaker verification and identification," Neural Networks for Signal Processing X, in *Proceedings of the IEEE Signal Processing Society Workshop*, vol. 2, pp. 775-784, 2000.

[14] J. Villalba, N. Brummer, N. Dehak, "Tied variational autoencoder backends for i-vector speaker recognition," in *Proceeding of INTERSPEECH*, pp. 1004-1008, 2017.

[15] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matejka and N. Brummer, "Discriminatively trained Probabilistic Linear Discriminant Analysis for speaker verification," in *Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4832-4835, 2011.

[16] S. Cumani, N. Brummer, L. Burget, P. Laface, O. Plchot and V. Vasilakakis, "Pairwise Discriminative Speaker Verification in the I-Vector Space," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 6, pp. 1217-1227, June 2013.

[17] E. Lehmann, J Romano, Testing Statistical Hypotheses, Springer-Verlag New York, 2005.

[18] E. Xing, A. Ng, M. Jordan, and R. Russell, "Distance Metric Learning, with application to Clustering with side-information," *in Proceeding of Advances in Neural Information Processing Systems*, MIT Press, pp. 521-528, 2002.

[19] B. Moghaddam, T. Jebara, A. Pentland, "Bayesian face recognition," *Pattern Recognition*, vol. 33, pp. 1771-1782, 2000.

[20] J. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," *Advances in large margin classifiers*, pp. 61-74, 1999.
[21] H. Lin, C. Lin, R. Weng, "A note on Plattfs probabilistic outputs for support vector machines," *Machine Learning*, vol. 68, pp. 267-276, 2007.
[22] N. Brummer, E. Villiers, "The BOSARIS toolkit user guide: Theory, algorithms and code for binary classifier score processing," Documentation of BOSARIS toolkit, 2011.
[23] X. Lu, P. Shen, Y. Tsao, H. Kawai, "Regularization of neural network model with distance metric learning for i-vector based spoken language identification," *Computer Speech and Language*, vol.44, pp. 48-60, 2017.
[24] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The speakers in the wild (SITW) speaker recognition database," in *Proceeding of INTERSPEECH*, pp. 818-822, 2016.
[25] A. Nagrani, J. Chung, W. Xie, A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Science and Language*, vol. 60, 2020.
[26] D. P. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization," *the 3rd International Conference on Learning Representations (ICLR)*, 2014.
[27] J. Chung, J. Huh, S. Mun, M. Lee, H. Heo, S. Choe, C. Ham, S. Jung, B. Lee, I. Han. "In defence of metric learning for speaker recognition," *Interspeech*, 2020.