# Learning Speaker-Specific Characteristics with a Deep Neural Architecture

Ke Chen, *Senior Member, IEEE*, and Ahmad Salman

***Abstract*— Speech signals convey various yet mixed information ranging from linguistic to speaker-specific information. However, most of acoustic representations characterize all different kinds of information as whole, which could hinder either a speech or a speaker recognition (SR) system from producing a better performance. In this paper, we propose a novel deep neural architecture (DNA) especially for learning speaker-specific characteristics from mel-frequency cepstral coefficients, an acoustic representation commonly used in both speech recognition and SR, which results in a speaker-specific overcomplete representation. In order to learn intrinsic speaker-specific characteristics, we come up with an objective function consisting of contrastive losses in terms of speaker similarity/dissimilarity and data reconstruction losses used as regularization to normalize the interference of non-speaker-related information. Moreover, we employ a hybrid learning strategy for learning parameters of the deep neural networks: i.e., local yet greedy layerwise unsupervised pretraining for initialization and global supervised learning for the ultimate discriminative goal. With four Linguistic Data Consortium (LDC) benchmarks and two non-English corpora, we demonstrate that our overcomplete representation is robust in characterizing various speakers, no matter whether their utterances have been used in training our DNA, and highly insensitive to text and languages spoken. Extensive comparative studies suggest that our approach yields favorite results in speaker verification and segmentation. Finally, we discuss several issues concerning our proposed approach.**

***Index Terms*— Deep neural architecture, hybrid learning strategy, overcomplete representation, speaker comparison, speaker segmentation, speaker verification, speaker-specific characteristics.**

## I. Introduction

AS ONE of the most important ways for human communication, speech conveys various yet mixed information. While the major information in a speech signal is linguistic contents expressing a message to be delivered, the speech signal also contains nonverbal information such as speaker-specific and emotional information as a result of various individuals' vocal apparatus and emotional states during speech production. For human communication, all the information conveyed in speech turns out to be very useful, and people are good at making use of appropriate information for different perceptual tasks. For instance, people can recognize a speaker regardless of what is spoken for speaker recognition (SR), while they easily understand linguistic contents spoken by different speakers for speech recognition.

Although humans often perform it effortlessly, the use of appropriate information in speech for a specific perceptual task still remains one of the main hurdles to hinder automatic speech information processing systems from yielding a better performance. Since there is no effective way to distill the information of interest from speech for a given task, a generic spectral representation of speech containing different kinds of information is often employed in various speech information processing tasks including speech recognition and SR [1]–[3]. Due to the interference among different kinds of information, the use of a spectral representation often makes speech recognition and SR systems compromised with a moderate performance. Thus, exploring a task-specific representation characterizing the proper information for a given task, e.g., a speaker-specific representation for SR, has posed a challenge to speech information processing for a long time [1], [3]–[7]. To tackle this problem, many efforts have been made, ranging from exploration of prosodic acoustic features and phonetic categories sensitive to speaker variations [8] to enhanced spectral representations [5]. In addition, feature selection and data component analysis techniques, e.g., principal component analysis or independent component analysis, have been also applied to explore speaker-specific characteristics [1], [2], [9]. However, such techniques either obtain features simply overfitting to a specific dataset or fail to associate the extracted data components with speaker-specific information. Despite the limited progress mentioned above, the problem is still unsolved in general [2], [4].

It is well known that most of artificial intelligence (AI) tasks including speech information processing often get involved in a complex problem-solving process and therefore need to be fulfilled by learning highly complex functions from a machine learning (ML) perspective [10]. Although there are different varieties of ML models, recent theoretic studies in the ML community have suggested that deep architectures (DAs) become one of the best candidates for learning highly complex functions that can represent the high-level abstraction demanded by various AI tasks ranging from perception to decision making [10], [11]. A *DA* is a parametric model consisting of multiple levels of nonlinear operations, e.g., a neural network with many hidden layers, where many subformulae are repeatedly used to form complicated propositional formulae in a mathematical sense. By defining appropriate objectives, a DA learns parameters from data to optimize the objectives as required. As a result, the learned DA would produce a desired

Manuscript received August 17, 2010; revised August 25, 2011; accepted August 26, 2011. Date of publication September 26, 2011; date of current version November 2, 2011.

The authors are with the School of Computer Science, University of Manchester, Manchester M13 9PL, U.K. (e-mail: chen@cs.manchester.ac.uk; salmanaa@cs.manchester.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2011.2167240

high-level abstract representation of the input data in a flexible way [12]. Nevertheless, learning in a DA is often an extremely difficult optimization task, as learning easily gets stuck in local optima or plateaus [11], [13].

Except for few successful applications [14], to a great extent, DAs were not applied to real-world problems until Hinton and his colleagues invented a systematic solution to the optimization problem for deep belief nets [15]. This solution appears to be a hybrid learning strategy composed of greedy layerwise unsupervised learning for pretraining and global supervised learning for fine-tuning parameters [13], [15]. Such a hybrid learning strategy has turned out to be applicable to other types of DAs [16]–[18], e.g., deep neural networks. The principle behind unsupervised pretraining has been further investigated very recently [19]. In the last couple of years, DAs have been successfully applied to several difficult real-world problems including handwritten character recognition [13], [14], [20], face detection [21] and verification [22], and generic object recognition [17], [23]. Comparative studies have shown that DAs are superior to many state-of-the-art ML techniques in terms of several well-designed benchmarks [24].

Very recently, convolutional deep belief networks (CDBNs) have been applied to spectrograms for unsupervised acoustic feature learning where their objective was minimizing the reconstruction errors and regularized by a sparsity penalty term to find appropriate representations for audio classification [25]. Representations formed at different layers have been investigated and some interesting results have been achieved, e.g., gender characteristic distribution in the achieved representation. Although they apply their representations to speaker identification, their goal is using DAs to discover generic yet novel representations for various audio classification tasks other than exploring intrinsic speaker-specific characteristics, which is a problem that we focus on in this paper.

Inspired by the aforementioned successful applications of DAs [13], [14], [20]–[23], [25], we explore speaker-specific characteristics with a novel DA trained by the hybrid learning strategy [15], [16]. The contributions of this paper are summarized as follows. First, we propose a deep neural architecture (DNA) especially for learning speaker-specific characteristics from a generic spectral representation, e.g., mel-frequency cepstral coefficients (MFCCs). For the proposed DA, we come up with an objective function to ensure that our DA tends to yield a speaker-specific representation via learning intrinsic speaker similarity/dissimilarity while normalizing the interference from non-speaker-related information. Second, we apply the hybrid learning strategy [13], [15], [16] to our proposed objective function, which leads to a two-stage learning algorithm for our DA. Third, we empirically justify that the proposed DA outperforms alternative DAs and other related techniques in terms of learning speaker-specific characteristics. Finally, we demonstrate that our overcomplete representation is robust in characterizing various speakers, no matter whether their speech has been used in training our DA, and insensitive to text and languages spoken. All the contributions are verified thoroughly by means of comparative studies on four Linguistic Data Consortium (LDC) benchmarks [26] and two additional non-English corpora [27], [28], all six corpora were collected for SR. To the best of our knowledge, we are the first to apply deep learning to explore speaker-specific characteristics.

In the remainder of this paper, Section II presents our DNA and its learning algorithms. Section III describes our experimental methodology and reports experimental results in speaker-related tasks. Section IV discusses several issues concerning our approach and relates our DNA to other DAs. The last section draws conclusions.

## II. Model Description

In this section, we first propose a DNA designed especially for exploring speaker-specific characteristics. Then we present a two-stage learning algorithm by applying the hybrid learning strategy [13], [15], [16] to our proposed objective function to train the deep neural networks. Finally, we describe a fast speaker modeling method and a distance metric used in our experiments.

### A. DNA

As illustrated in Fig. 1(a), our DNA consists of two identical subnets, and each subnet is a fully connected multilayered feed-forward neural network of $2K-1$ hidden layers, where $K > 1$. $\boldsymbol{x}_i$ $(i = 1, 2)$ are input to two subnets, i.e., spectral representations of two frames after short-term speech analysis in our work, and the output of the top layers $\hat{\boldsymbol{x}}_i$ $(i = 1, 2)$ are their reconstruction of input $\boldsymbol{x}_i$ $(i = 1, 2)$ in two subsets. While the $K$th hidden layer is specified as a *code layer* whose output would be used as a new representation of input data after learning, the $k$th and the $(2K-k)$th hidden layers have the same number of neurons where $k = 1, \ldots, K-1$. Moreover, neurons in the code layer are divided into two groups: one would be used to characterize speaker-specific information, and the other is expected to encode non-speaker-related information. Two subnets are associated with each other in their code layers via a compatibility measure $E$ on those neurons characterizing speaker-specific information during learning. Let $\boldsymbol{ce}(\boldsymbol{x}_i; \Theta)$ $(i = 1, 2)$ denote output of those neurons corresponding to a speaker-specific representation in the code layer of two subnets, where $\Theta$ is a collective notation of all connection weights and biases in the DNA. We define the compatibility measure as

$$E(\boldsymbol{x}_1, \boldsymbol{x}_2; \Theta) = ||\boldsymbol{ce}(\boldsymbol{x}_1; \Theta) - \boldsymbol{ce}(\boldsymbol{x}_2; \Theta)||_1 \tag{1}$$

where $||\cdot||_1$ is the $\mathcal{L}_1$ norm. Hereinafter, we shall drop explicit parameters from $E(\boldsymbol{x}_1, \boldsymbol{x}_2; \Theta)$ to facilitate our presentation.

To learn intrinsic speaker-specific characteristics, we come up with a loss function given input $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and a binary indicator $\mathcal{I}$ with respect to the input, where $\mathcal{I} = 1$ if $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are extracted from speech of the same speaker, and $\mathcal{I} = 0$ otherwise. As a result, we define the loss function on any two frames as

$$\begin{aligned} L(\boldsymbol{x}_1, \boldsymbol{x}_2, \mathcal{I}; \Theta) = {} & [L_R(\boldsymbol{x}_1; \Theta) + L_R(\boldsymbol{x}_2; \Theta)] \\ & + L_E(\boldsymbol{x}_1, \boldsymbol{x}_2; \Theta) \end{aligned} \tag{2}$$

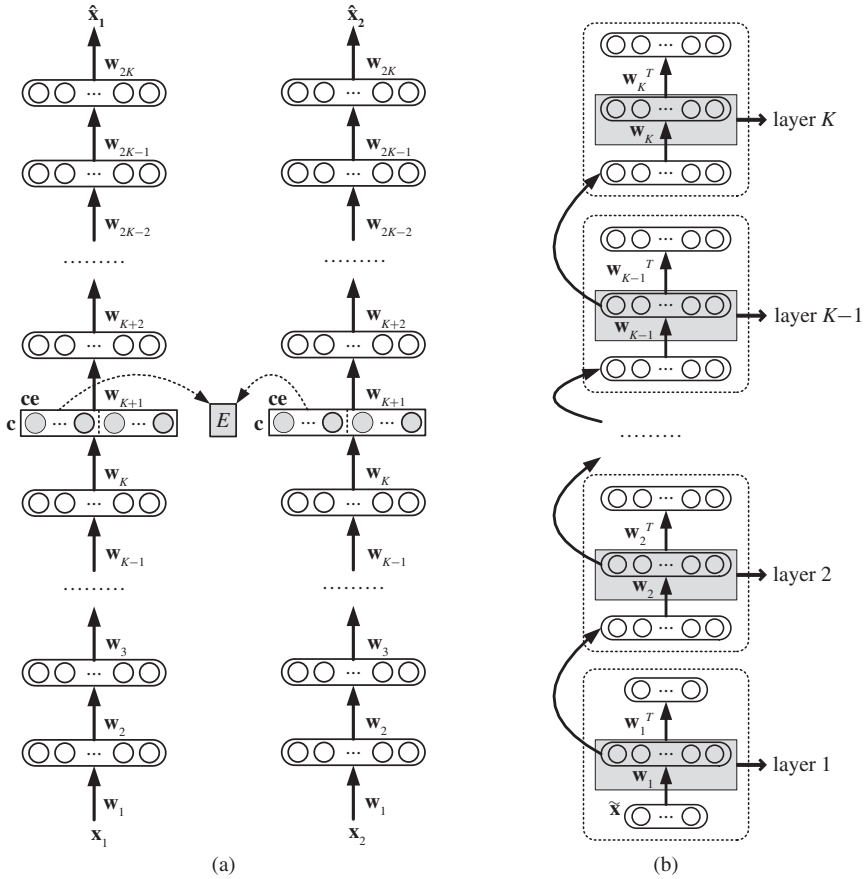


Fig. 1. Model for learning speaker-specific characteristics. (a) Deep neural architecture. (b) Greedy layerwise pretraining.

where

$$L_R(\boldsymbol{x}_i;\Theta) = \alpha||\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i||_2^2, \quad i = 1,2 \tag{3a}$$

$$L_E(\boldsymbol{x}_1,\boldsymbol{x}_2,\mathcal{I};\Theta) = (1-\alpha)\big\{\mathcal{I}[E(\boldsymbol{x}_1,\boldsymbol{x}_2;\Theta)]^2 + (1-\mathcal{I})e^{-\lambda E(\boldsymbol{x}_1,\boldsymbol{x}_2;\Theta)}\big\}. \tag{3b}$$

Here, $||\cdot||_2$ is the $\mathcal{L}_2$ norm. $L_R(\boldsymbol{x}_i;\Theta)$ are the losses incurred by data reconstruction errors for the given input of two subnets, while $L_E(\boldsymbol{x}_1,\boldsymbol{x}_2,\mathcal{I};\Theta)$ are contrastive losses caused by the incompatibility in two different situations indicated by $\mathcal{I}$ in a speaker-specific representation space. $\lambda$ in (3b) is a constant determined by the upper bound of $E(\boldsymbol{x}_1,\boldsymbol{x}_2;\Theta)$ on all training data. Note that the use of $\mathcal{L}_1$ norm in (1) and different cost functions in (3b) for two different situations is to meet conditions required by an energy-based model for discriminative learning [29]. Thus, our contrast loss in (3b) avoids the *collapse* problem that creates a dangerous plateau in a loss function and therefore leads to either a trivial solution or a failure to learn whatever is expected after optimization [29]. In general, (2) defines a multiobjective function and $\alpha$ is a constant used to trade off the contrastive loss in (3b) which hinders our architecture from learning speaker-specific characteristics against the reconstruction loss in (3a) that results in information loss.

Now we would make several remarks on our DNA shown in Fig. 1(a) in terms of relevant yet alternative DAs [13], [20], [22], which also make them self-contained to facilitate our presentation as we shall empirically justify that our proposed DNA is superior to some alternative yet relevant DAs in terms of learning speaker-specific characteristics.

*Remark 1:* Each subset itself is a deep *autoencoder (AE)* architecture originally proposed in [13]. By minimizing reconstruction errors, e.g., the one defined in (3a), the code layer tends to generate a representation that encodes all the important information underlying raw data [12], [13] and presents it in a stable way. While such an architecture was applied to yield a parsimonious representation [13], we believe that an overcomplete representation in the code layer of our subnet would better distribute and disentangle different types of speech information to facilitate learning speaker-specific characteristics. Here we emphasize that a single deep AE subnet does not generate a speaker-specific representation, which will be demonstrated in our experiments.

*Remark 2:* If we modify our DNA by removing all the layers above the code layer and associating two subnets with all neurons in their code layers via a compatibility measure $E$, then it will become a typical *Siamese (S)* architecture [30]. By minimizing the incompatibility, e.g., the one defined in (3b), the code layer tends to yield a "semantic" distance metric. While the S architecture of deep convolutional neural networks was applied to learn facial identity characteristics [22], which is the major or dominant information in facial images, the original S architecture is not appropriate to our problem as speaker-specific information is minor in comparison to lingual

information in speech, which will be empirically justified in our experiments.

*Remark 3:* Based on Remarks 1 and 2, our DNA can be viewed as a regularized Siamese (RS) architecture in which data reconstruction is used as regularization to normalize the interference from non-speaker-related information so as to avoid information loss and overfitting to training data during discriminative learning. In particular, we anticipate that splitting the code layer into speaker-specific and non-speaker-specific parts would better facilitate normalizing non-speaker-related information and isolating the spectral degradation in noisy narrow-band speech from the speaker-specific representation. We shall demonstrate this advantage by comparison with the deep AE [13] and the S architecture [22] in our experiments. In addition, a similar DA has been used to facilitate learning topological characteristics of handwritten digits in a semisupervised setting [20]. We shall discuss the difference between that architecture [20] and ours later on.

### B. Learning Algorithm

We now apply the hybrid learning strategy [13], [15], [16] to our loss function in (2) to derive a two-stage learning algorithm, *pretraining* and *discriminative learning*, for our architecture shown in Fig. 1(a). Before describing the learning algorithm, we first present notations used in our DNA.

For input $\boldsymbol{x}$ of a subnet, let $h_{kj}(\boldsymbol{x})$ denote the output of the $j$th neuron in layer $k$ for $k = 0, 1, \ldots, K, \ldots, 2K$. $\boldsymbol{h}_k(\boldsymbol{x}) = \left(h_{kj}(\boldsymbol{x})\right)_{j=1}^{|\boldsymbol{h}_k(\boldsymbol{x})|}$ is a collective notation of the output of all neurons in layer $k$, where $|\boldsymbol{h}_k(\boldsymbol{x})|$ indicates the number of neurons in layer $k$. $k = 0$ refers to the input layer with $\boldsymbol{h}_0(\boldsymbol{x}) = \boldsymbol{x}$, and $k = 2K$ refers to the top layer producing the reconstruction $\hat{\boldsymbol{x}}$. In particular, layer $K$ is specified as the code layer so that $\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{h}_K(\boldsymbol{x})$ and, moreover, $\boldsymbol{ce}(\boldsymbol{x}) = \left(h_{Kj}(\boldsymbol{x})\right)_{j=1}^{|\boldsymbol{ce}|}$ corresponding to an abstract yet new speaker-specific representation after learning. Let $W_k$ be the connection weight matrix between layers $k-1$ and $k$ and $\boldsymbol{b}_k$ denotes the bias vector of layer $k$ for $k = 1, \ldots, 2K$. Then output of layer $k$ is

$$\boldsymbol{h}_k(\boldsymbol{x}) = \sigma[\boldsymbol{u}_k(\boldsymbol{x})], \quad k = 1, \ldots, 2K-1 \tag{4}$$

where

$$\boldsymbol{u}_k(\boldsymbol{x}) = W_k \boldsymbol{h}_{k-1}(\boldsymbol{x}) + \boldsymbol{b}_k \tag{5a}$$

$$\sigma(\boldsymbol{z}) = \left((1+e^{-z_j})^{-1}\right)_{j=1}^{|\boldsymbol{z}|}. \tag{5b}$$

Given $\boldsymbol{h}_0(\boldsymbol{x}) = \boldsymbol{x}$, we have $\hat{\boldsymbol{x}} = \boldsymbol{u}_{2K}(\boldsymbol{x})$ as its reconstruction, i.e., instead of the nonlinear transfer function in (4) whose value lies in (0, 1), we use the linear transfer function in the top layer, layer $2K$, to reconstruct the original input. In other words, we do not normalize input to [0, 1], as previous studies have revealed that the normalization of a spectral representation with a whitening procedure considerably degrades the performance of SR [6], [31], [32].

*1) Pretraining:* In the hybrid learning strategy [13], [15], [16], pretraining is an unsupervised learning process that initializes weights via a greedy layerwise learning procedure. As illustrated in Fig. 1(b), pretraining starts from the input layer of a subnet. The weight matrix between layers $k$ and $k-1$ is learned via an autoassociator of one hidden layer described below where layer 0 is stipulated as the input layer of a subnet. After learning in an autoassociator, the weight matrix between the hidden and the input layers in the autoassociator is used to be the initial weight matrix between layers $k$ and $k-1$ of the subnet, and output of the hidden layer in the autoassociator will form a representation of previous input data to be used for achieving the initial weight matrix between layers $k+1$ and $k$. As a result, the greedy layerwise learning procedure leads to initial weight matrices for the first $K$ layers, i.e., $W_1, \ldots, W_K$. Then, we set $W_{K+k} = W_{K-k+1}^T$ for $k = 1, \ldots, K$ to initialize $W_{K+1}, \ldots, W_{2K}$ of the subnet.

In this paper, we apply the denoising autoassociator [33] to learn biases and the initial connection weight matrix between two adjacent layers of a subnet. A denoising autoassociator is a three-layered perceptron in which the input $\tilde{\boldsymbol{x}}$ is a distorted version of the target output $\boldsymbol{x}$. For a training example, $(\tilde{\boldsymbol{x}}, \boldsymbol{x})$, the output of the autoassociator, is the restored version $\hat{\boldsymbol{x}}$. Since the spectral representation fed to the first hidden layer and its intermediate representation input to all other hidden layers in our DNA are of continuous value, we distort either the spectral representation or its intermediate representation $\boldsymbol{x}$ by adding Gaussian noise to each feature in the representation to form a distorted version $\tilde{\boldsymbol{x}}$. The restoration learning is done by minimizing the following loss with respect to the weight matrix:

$$L_{\text{dec}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = ||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2^2. \tag{6}$$

Thus, the loss function in (6) is used to achieve biases of all hidden layers and the initial weight matrix between layers $k-1$ and $k$ for $k = 1, \ldots, K$ in a layer-by-layer way, as shown in Fig. 1(b). The appendix details the pretraining algorithm used to train each building block.

*2) Discriminative Learning:* Discriminative learning is a process of finding appropriate connection weight matrices and biases in our DNA based on the initialization to minimize our loss function defined in (2). We apply the stochastic backpropagation (SBP) algorithm [14] to fulfill this task and therefore need to derive relevant gradients from (2) via (3a) and (3b), respectively, for a training example $(\boldsymbol{x}_1, \boldsymbol{x}_2; \mathcal{I})$. To facilitate our presentation, we drop those explicit parameters from our formulae.

For losses defined in (3a), we have the following gradients. When $k = 2K$, i.e., the top layer of subnets

$$\frac{\partial L_R}{\partial \boldsymbol{u}_{2K}^i} = 2\alpha(\hat{\boldsymbol{x}}^i - \boldsymbol{x}^i). \tag{7}$$

For all hidden layers, $k = 2K-1, \ldots, 1$, applying the chain rule and (7) leads to

$$\frac{\partial L_R}{\partial \boldsymbol{u}_k^i} = \left(\frac{\partial L_R}{\partial h_{kj}^i} h_{kj}^i \left(1 - h_{kj}^i\right)\right)_{j=1}^{|\boldsymbol{h}_k^i|} \tag{8a}$$

$$\frac{\partial L_R}{\partial \boldsymbol{h}_k^i} = \left[W_{k+1}^i\right]^T \frac{\partial L_R}{\partial \boldsymbol{u}_{k+1}^i}. \tag{8b}$$

Note that superscript $i$ in (7), (8), and all the remaining equations in this section indicates subnet $i$ for $i = 1, 2$.

For losses defined in (3b), we have the following gradients. For the code layer of two subnets ($i = 1, 2$), i.e., hidden layer $K$ and its output $\boldsymbol{c} = \boldsymbol{h}_K$, we have

$$\frac{\partial L_E}{\partial \boldsymbol{u}_K^i} = \Big((1-\alpha)[2\mathcal{I}E - \lambda(1-\mathcal{I})e^{-\lambda E}]\Gamma_j^i\Big)_{j=1}^{|\boldsymbol{c}|} \tag{9}$$

where $\Gamma_j^i = \text{sign}[(1.5-i)(ce_j^1 - ce_j^2)]ce_j^i(1-ce_j^i)$ when $j = 1, \ldots, |\boldsymbol{ce}|$ and $\Gamma_j^i = 0$ when $j = |\boldsymbol{ce}|+1, \ldots, |\boldsymbol{c}|$. For hidden layer $k = K-1, \ldots, 1$, applying the chain rule and (9) results in

$$\frac{\partial L_E}{\partial \boldsymbol{u}_k^i} = \left(\frac{\partial L_E}{\partial h_{kj}^i} h_{kj}^i \left(1 - h_{kj}^i\right)\right)_{j=1}^{|\boldsymbol{h}_k^i|} \tag{10a}$$

$$\frac{\partial L_E}{\partial \boldsymbol{h}_k^i} = \left[W_{k+1}^i\right]^T \frac{\partial L_E}{\partial \boldsymbol{u}_{k+1}^i}. \tag{10b}$$

Given a training dataset, training data are randomly divided into several batches for the SBP algorithm where parameter update in our DNA is done based on each batch of $T_B$ training examples, $\left\{\left(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t); \mathcal{I}(t)\right)\right\}_{t=1}^{T_B}$. As our DNA would have two identical subnets after learning, all the weights and biases in two subnets are always kept exactly the same during discriminative learning. For layer $k = K+1, \ldots, 2K$, their parameters solely depend on the losses defined in (3a). Hence the use of (7) and (8) in the SBP algorithm immediately leads to

$$W_k^i \leftarrow W_k^i - \frac{\epsilon}{T_B}\sum_{t=1}^{T_B}\sum_{r=1}^{2}\frac{\partial L_R(t)}{\partial \boldsymbol{u}_k^r(t)}[\boldsymbol{h}_{k-1}^r(t)]^T \tag{11a}$$

$$\boldsymbol{b}_k^i \leftarrow \boldsymbol{b}_k^i - \frac{\epsilon}{T_B}\sum_{t=1}^{T_B}\sum_{r=1}^{2}\frac{\partial L_R(t)}{\partial \boldsymbol{u}_k^r(t)}. \tag{11b}$$

For layer $k = 1, \ldots, K$, their parameters are determined by (3a) and (3b). Applying (7)–(10) in the SBP algorithm results in

$$W_k^i \leftarrow W_k^i - \frac{\epsilon}{T_B}\sum_{t=1}^{T_B}\sum_{r=1}^{2}\left(\frac{\partial L_R(t)}{\partial \boldsymbol{u}_k^r(t)} + \frac{\partial L_E(t)}{\partial \boldsymbol{u}_k^r(t)}\right)[\boldsymbol{h}_{k-1}^r(t)]^T \tag{12a}$$

$$\boldsymbol{b}_k^i \leftarrow \boldsymbol{b}_k^i - \frac{\epsilon}{T_B}\sum_{t=1}^{T_B}\sum_{r=1}^{2}\left(\frac{\partial L_R(t)}{\partial \boldsymbol{u}_k^r(t)} + \frac{\partial L_E(t)}{\partial \boldsymbol{u}_k^r(t)}\right). \tag{12b}$$

Here, $\epsilon$ in (11) and (12) is a learning rate.

### C. Speaker Modeling and Comparison

After a short-term analysis, an utterance of a speaker is divided into $T_B$ frames and their spectral representations are collectively denotated as $\left\{\boldsymbol{x}(t)\right\}_{t=1}^{T_B}$. When these frames are input to one of two identical subnets in the trained DNA as presented in Section II-B, outputs of first $|\boldsymbol{ce}|$ neurons in the code layer are their new representations of $T_B$ frames, $\left\{\boldsymbol{ce}(t)\right\}_{t=1}^{T_B}$, where $ce_j(t) = h_{Kj}(t)$ and $j = 1, \ldots, |\boldsymbol{ce}|$. Then, we employ statistics on the new representations of $T_B$ frames to establish a speaker model (SM). In our experiments, we simply use the first- and second-order statistics to form an SM $S = \{\boldsymbol{\mu}, \Sigma\}$, where

$$\boldsymbol{\mu} = \frac{1}{T_B}\sum_{t=1}^{T_B}\boldsymbol{ce}(t), \quad \Sigma = \frac{1}{T_B}\sum_{t=1}^{T_B}[\boldsymbol{ce}(t) - \boldsymbol{\mu}][\boldsymbol{ce}(t) - \boldsymbol{\mu}]^T.$$

Note that the speaker modeling technique is also applicable to other representations used in our experiments for comparison.

*Speaker comparison (SC)* is a process that finds the speaker distance between two speech signals and provides an underpinning technique for many speaker-related tasks [34]. Since we have modeled speakers with the first- and the second-order statistics of speech signals based on a representation, we directly compare two SMs $S_i = \{\boldsymbol{\mu}_i, \Sigma_i\}$ $(i = 1, 2)$ with a distance metric. In our experiments, we employ

$$d(S_1, S_2) = tr\left[(\Sigma_1^{-1} + \Sigma_2^{-1})(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\right]. \tag{13}$$

Intuitively, $d(S_1, S_2)$ is expected to be large, as $S_1$ and $S_2$ belong to two different speakers, and small otherwise. The distance metric in (13) is derived from the original divergence metric for two normal distributions [1] by dropping the term concerning only covariance matrices. Doing so is based on our observation that covariance matrices often vary considerably as short utterances are used to establish SMs but (13) is fairly stable irrespective of utterance lengths.

## III. Experiment

In this section, we first describe the experimental methodology, the general experimental settings, and the corpora used in our experiments. Then we present the SC results based on MFCCs and representations yielded by our DNA and DAs (see remarks described in Section II-A for details) to empirically justify the effectiveness of our proposed DNA in learning speaker-specific characteristics. Then, we demonstrate the effectiveness of our proposed approach by applying the speaker-specific representations achieved with our DNA to speaker verification and segmentation tasks along with a comparison to relevant state-of-the-art techniques.

### A. Experimental Setting

In our experiments, we employ four LDC benchmark corpora in English [26], i.e., TIMIT, narrow-band TIMIT (NTIMIT), KING, and narrow-band KING (NKING), to train different DAs for new representations, respectively. Also we use two non-English corpora, one in Chinese (CHN) [28] and the other in Russian (RUS) [27], for test in the cross-corpora and the cross-language experiments. Table I summarizes the information on the six corpora.

In our experiments, we adopt the MFCCs to be a raw acoustic representation of speech since this representation has been widely used in various speech information processing tasks and has led to good performance. The same acoustic analysis and feature extraction procedure as used in [6], [7], [35] is applied to six corpora to extract MFCCs as follows: 1) removing silent parts in speech signals with an energy-based method; 2) pre-emphasis with the filter $H(z) = 1 - 0.95z^{-1}$; 3) Hamming windowing speech by a frame size of 20 ms with

TABLE I
INFORMATION ON THE CORPORA USED IN OUR EXPERIMENTS

| Corpus | Speaker no. | Session no. | Sampling | Bandwidth |
|---|---|---|---|---|
| TIMIT | 630 | 1 | 16 kHz | 0–8 kHz |
| NTIMIT | 630 | 1 | 16 kHz | 0.3–3.3 kHz |
| KING | 49 | 10 | 8 kHz | 0–4 kHz |
| NKING | 51 | 10 | 8 kHz | 0.3–3.3 kHz |
| CHN | 59 | 3 | 16 kHz | 0–8 kHz |
| RUS | 50 | 1 | 8 kHz | 0–4 kHz |

a frame shift of 10 ms; 4) applying 24 mel-scale triangular filters to calculate magnitude spectrum; and 5) extracting MFCCs by excluding the coefficient of order zero. Thus, 19-order MFCCs are achieved for each frame for all corpora except NTIMIT where 15-order MFCCs are only used due to channel losses (see the explanation in [7] for details). Note that the use of high-order MFCCs was suggested in [1]–[3] and [4] to avoid loss of speaker-specific information.

In our experiments, we train DAs on four LDC corpora for a thorough evaluation. To train and test a DA, we randomly divide speakers in a corpus into two groups, say A and B. For each speaker in group A, we further split his/her utterances into two subsets randomly; one used for training a DA and the other served for validation, hereinafter named *training set* and *validation set*, respectively. All utterances of speakers' in group B are reserved as a *test set* for various experiments. In the experiments reported in this paper, 50 speakers in TIMIT/NTIMIT and 20 speakers in KING/NKING are randomly chosen to constitute group A. For each of four LDC corpora, a training set is composed of five utterances of each speaker's in group A on TIMIT/NTIMIT or all the utterances of those speakers' in group A that were collected in first two sessions on KING/NKING. During the collection, there were equipment/environmental changes on KING/NKING and further a so-called "Great Divide" problem on NKING [26]. Therefore, we mainly employ all the utterances collected in sessions 1–5 in our experiments and also investigate the effect of equipment/environmental changes and the "Great Divide" with utterances collected in sessions 6–10 in speaker verification experiments. For a complete cross-corpora/cross-language experiment in speaker segmentation, we also train our DNA on the CHN corpus where 19 speakers are randomly chosen to form group A.

For learning in our DNA and two alternative DAs described in Section II-A, we have carried out empirical studies via an exhaustive search for a reasonable parameter subspace with the cross-validation method. Here, we present the best parameter values used in our experiments. For our cost function, we set $\alpha = 0.2$ in (2) and $\lambda = 1/100$ in (3b). For model selection of DAs, we look into a number of different subnet structures corresponding to $2 \leqslant K \leqslant 5$ [see Fig. 1(a)] and the number of neurons in a layer ranging from 50 to 500 where layer 0 always refers to the input layer. Our empirical studies reveal that a subnet structure of $K = 4$ yields the best performance for three DAs in general. In detail, we adopt the structure with the number of neurons in layers 1–4 as 100, 100, 100, and 200, respectively, where $|\boldsymbol{ce}|$ in the code layer, i.e., layer 4, is 100 in our DNA. Note that layer 4 in the S architecture is the output layer, while this layer is hidden and the code layer in the deep AE and our DNA (see remarks in Section II-A for details). The learning rates $\epsilon$ in pretraining and discriminative learning are 0.01 and 0.001, respectively. In pretraining, we add Gaussian noise subject to $N(0, 0.25\sigma_j)$ to feature $j$ in a representation input to a denoising autoassociator for restoration learning, where $\sigma_j$ is the standard deviation of feature $j$ estimated from the training set. To avoid overfitting, the number of epochs for discriminative learning is determined by an early stopping criterion.

### B. Speaker Comparison (SC)

SC provides a direct way of evaluating representations produced by different DAs with MFCCs as the reference point. In this experiment, we make use of a test set with known speaker identities to generate two data subsets with the same protocol used in producing a training set for learning parameters in DAs, as described in Section II-A. We first divide all test utterances into segments of a fixed length and establish a SM with a segment based on a representation of its frames as described in Section II-C. Then we exhaustively combine any two SMs to generate SM pairs. If two SMs in a pair correspond to the same speaker, they become a member of the *genuine pair* subset. Otherwise, they are a member of the *imposter pair* subset.

In our SC experiments, we use MFCCs and representations produced by the deep AE, the S, and our RS to establish an SM, hereinafter named *SM-MFCC, SM-AE, SM-S*, and *SM-RS*, respectively, as described in Section II-C. For performance evaluation, we use the *detection error tradeoff* (DET) measure [36] to show all possible errors made in decision making during SC where the area of the *operating region* enclosed by a DET curve and two error axes is generally regarded as one of the best performance indexes for a detection task. Fig. 2 depicts DET curves on test sets of four LDC benchmark corpora with MFCCs and three different representations yielded by aforementioned DAs, where speech segments of 5 s are used for speaker modeling.

It is evident from Fig. 2 that representations by our DNA outperform MFCCs and those by the deep AE and the S architectures on all four corpora given the fact that the DNA representations always lead to smaller operating regions irrespective of corpora. In particular, it appears that our DNA representation leads to an operating region of zero area on the DET plane, which indicates no errors in SC, on the TIMIT corpus of 630 speakers. In contrast, neither the deep AE nor the S yields a smaller operating region than MFCCs on TIMIT and KING, although the representations yielded by them lead to slightly smaller or similar operating regions in comparison to those of MFCCs on the two narrow-band corpora NTIMIT and NKING. Note that changing a segment length in speaker modeling could alter the shape of the DET curves or their error rates, the use of a longer length in SC results in lower error rates in general. Nevertheless, all experimental results not reported here due to the limited space reveal that the use of

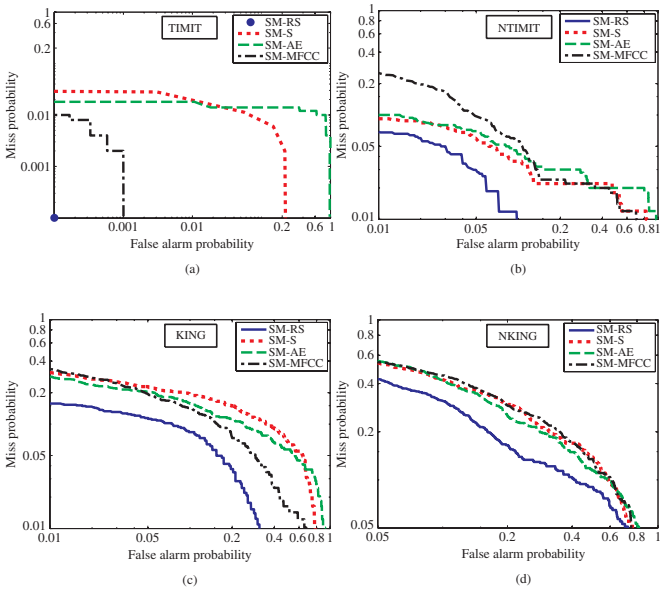


Fig. 2. SC performance for MFCCs and representations yielded by three different DAs on four LDC corpora. (a) DET curves on TIMIT. (b) DET curves on NTIMIT. (c) DET curves on KING. (d) DET curves on NKING.

different segment lengths in speaker modeling does not alter the conclusion drawn from Fig. 2. Thus, we conclude that representations by our DNA better characterize the speaker-specific information than MFCCs and those by other DAs in question. Next, we shall focus only on representations learned by our DNA and apply them to two real-world problems, i.e., speaker verification and segmentation, for further assessment.

### C. Speaker Verification (SV)

*SV* is a process that accepts or rejects the identity claim of a speaker. Typically, an SV system is composed of an SM and a decision-making mechanism [31], [37]. Since the work presented in this paper mainly focuses on speaker-specific representations, we employ the DET measure [36] for performance evaluation, which allows us to illustrate all possible errors made by an SV system via smoothly changing the thresholds in decision making. By using the DET measure, we avoid addressing other less relevant decision-making issues [31].

A *CDBN* [25] was recently applied to learning a generic representation from a spectrogram for audio classification. For a thorough evaluation, we also employ representations produced by the CDBN in our SV experiments. To make a fair comparison, we strictly follow their experimental protocols in [25] by using the same preprocessing procedure, the same CDBN structure, including the kernel size for feature maps and the neighborhood size for probabilistic maximal pooling, and the same sparsity penalty. While all other parameters are kept the same as in [25], we also exhaustively search three tunable parameter values in a broad range for the best performance with the cross-validation method. The best parameter values found for SV are as follows: the learning rate of 0.01; the sparsity parameter of 0.02; and the sparsity regularization constant of 0.1. As their CDBN structure has two hidden layers, output from either of the hidden layers and their combination by concatenating the output of two hidden layers form different representations [25], we have investigated all three representations by the CDBN in our experiments. It is worth mentioning that all features in the spectrogram input to the CDBN were normalized with a whitening procedure in their work [25]. By using both the normalized and the original spectrogram in our experiments, however, we find that representations achieved by the CDBN with the original spectrogram without the normalization always significantly outperforms those with the normalized spectrogram regardless of corpora, which is consistent with previous SR studies [6], [31], [32] and our work presented in this paper. In the sequel, we always use the best performance achieved by the CDBN to compare with others.

The Gaussian mixture model (GMM) trained on MFCCs has been a sophisticated SR technique [1], [2], [4] and, in particular, leads to the state-of-the-art SR performance on four LDC benchmark corpora used in our experiments [6], [7], [35]. Like most of existing SR studies, e.g., [1], [2], [4], [25], [31], and [32], we employ GMM with MFCCs to be a baseline system. In our experiments, we use the same experimental settings as used in [6] and [7], i.e., for a speaker we establish an SM, named *GMM-MFCC*, by training a GMM of 32 components on MFCCs with the expectation maximization (EM) algorithm and the likelihood score of an utterance is normalized with a background model in decision making. To train a GMM, we divide all utterances of a speaker into two subsets, i.e., training and test sets, respectively. To produce the training subset, we randomly choose five utterances of 14 s for TIMIT, NTIMIT, and CHN, utterances of 30 s for RUS, and utterances of 60 s, recorded in the first two sessions, for KING and NKING, respectively. The remaining utterances in each corpus form the test subset used to evaluate different SMs.

In our SV experiments, we employ different representations produced by the CDBN and our RS, to establish an SM, named *SM-CDBN* and *SM-RS*, respectively. To simulate the speaker enrolment process in an SV system, we use an utterance of a fixed length as a reference to establish an SM as described in Section II-C. For four LDC corpora, the speaker's reference speech is randomly chosen from either the validation set if his/her utterances got involved in training DAs, or the test set otherwise. For CHN and RUS corpora, all utterances of a speaker are randomly divided into training and test subsets, respectively. Then the reference utterance for enrolment on CHN/RUS is randomly chosen from the training subset. To simplify the presentation, we report the experimental results on the setting that the length of the reference speech is fixed at 5 s for TIMIT, NTIMIT, CHN, and RUS, and 10 s for KING and NKING. The use of a short reference utterance greatly facilitates the assessment of a representation to see if it characterizes the speaker-specific information well, although the use of a longer reference utterance leads to a considerably better performance. Here, we emphasize that the reference speech length for training a GMM-MFCC is significantly longer than that used to establish the SM-CDBN and the SM-RS on the same corpus given the fact that this is the minimum length that allows us to reproduce the same performance as

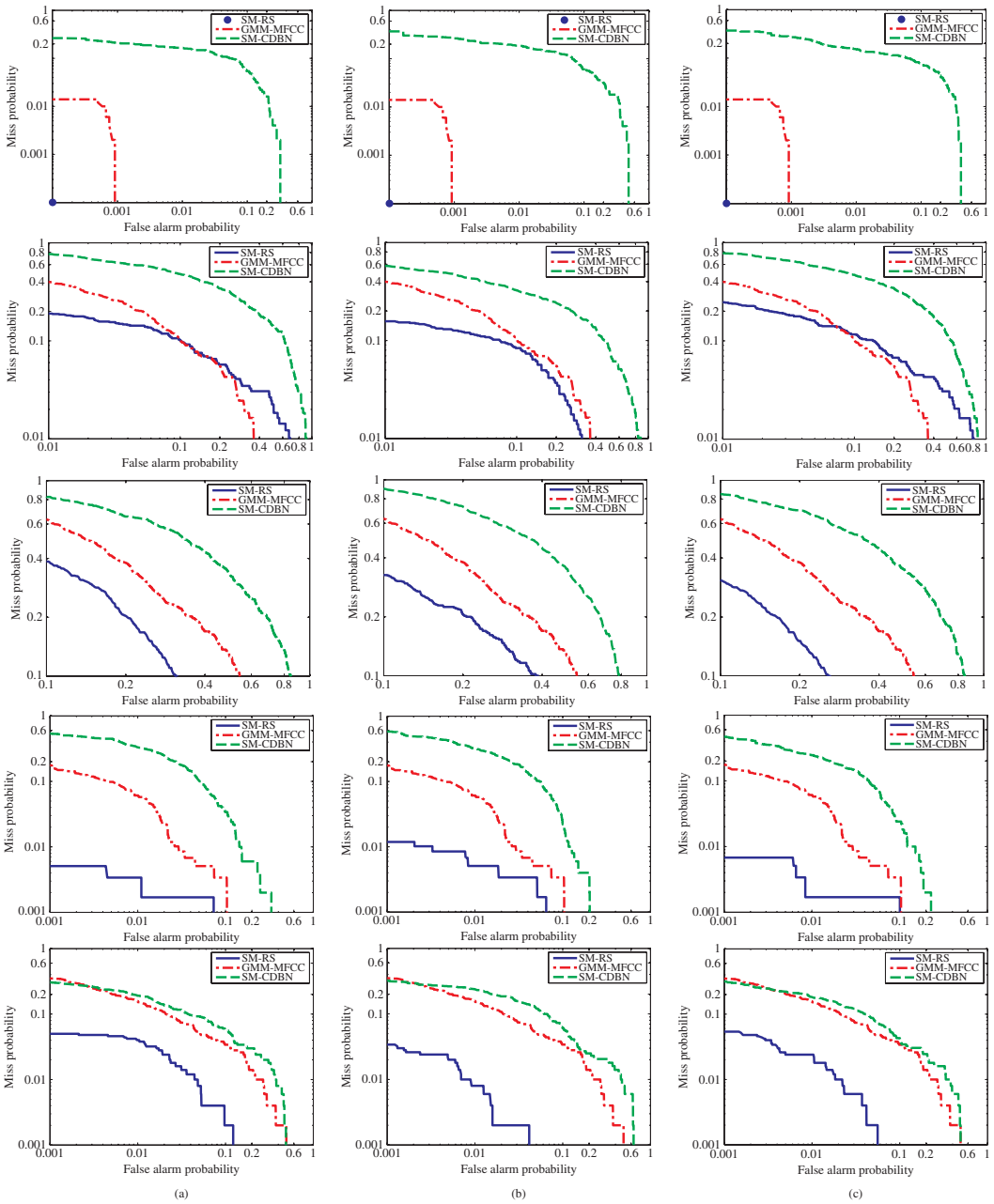


Fig. 3. SV performance of the SM-CDBM and the SM-RS with representations learned from different corpora versus the baseline performance of GMM-MFCC in cross-corpora and cross-language experiments. Five plots from top to bottom in (a)–(c) correspond to test results on TIMIT, KING, NKING, CHN, and RUS corpora, respectively. (a) DET curves for representations learned from TIMIT. (b) DET curves with representations learned from KING. (c) DET curves with representations learned from NKING.

reported in [6] and [7] by training GMM on MFCCs with the EM algorithm.

Since all speech in a corpus is often collected under the same condition, a representation learned from the corpus might overfit to only those speakers in the corpus. Hence, we first conduct *cross-corpus* and *cross-language* experiments to assess generalization, stability, and robustness of representations by the CDBN and our DNA. The cross-corpus experiments apply a representation by DAs learned from a single corpus to all corpora apart from NTIMIT, as the order of MFCCs extracted from this corpus is different from those used in other corpora as described in Section III-A, while the cross-language experiments apply a representation by DAs learned from an English corpus to two non-English corpora. Fig. 3 depicts the DET curves achieved by SMs for the test length of 5 s, where the DET curve achieved by GMM-MFCC is always illustrated in each plot although their performance remains unchanged in plots corresponding to a specific test corpus, i.e., the same DET curve appears in three plots in a row across Fig. 3(a)–(c). It is observed from Fig. 3 that representations by our DNA are stable and robust given the fact that a representation by the DNA trained on a corpus yields

not only the smallest operating regions on the same corpus but also the favorite performance on other corpora in general. In particular, our SM-RS, regardless of training corpora, yields the error-free performance on TIMIT, as shown in three plots at the top of Fig. 3, and outperforms GMM-MFCC on all test corpora apart from KING. As depicted in plots aligned in the second row of Fig. 3, DET curves on KING show that, in comparison with the baseline performance of GMM-MFCC, the use of representations learned from TIMIT and NKING in our SM-RS leads to worse performance in terms of false alarm but better performance in terms of missing, while it outperforms GMM-MFCC as the representation learned from KING itself is used. In contrast, representations by the CDBN do not capture speaker-specific characteristics well given the fact that the performance of the SM-CDBN is always significantly inferior to that of our SM-RS in all cross-corpora and cross-language experiments. In addition, the SM-CDBN also underperforms GMM-MFCC on all test corpora in general. The comparative results suggest that representations by our DNA are insensitive to the text and language spoken.

It is well known that a short length poses a challenge to an SR system [1], [2], [4], [6]–[8], [31], [34]. By means of NTIMIT, we conduct experiments on this noisy narrow-band corpus to compare different SMs in terms of short test utterance lengths. As illustrated in Fig. 4, our SM-RS significantly outperforms the SM-CDBN for short test lengths of 1–4 s, while it outperforms GMM-MFCC for test lengths of 2–4 s but its performance is similar to that of GMM-MFCC for 1 s in terms of the size of their operating regions. In contrast, the performance SM-CDBN is also worse than the baseline performance of GMM-MFCC in all four short test lengths. Thus, the comparative results on NTIMIT demonstrate that our SM-RS works well even though only a short utterance is available during test.

As described in [26], there were equipment and environmental changes after the fifth session during KING/NKING collection. Hence, sessions 1–5 and 6–10 form two different datasets in terms of the recording equipment and environment. With the same representations by the CDBN and our DNA trained on the first two sessions and the same experimental settings as used in all aforementioned experiments, we conduct two experiments to investigate the effect of recording equipment and environmental changes. One is a *within-boundary* experiment by using utterances in session 6 as speakers' references in enrolment for test on sessions 7–10 but those in session 7 as references for test on session 6, where there is no channel/environmental mismatch between the reference and the test utterances. The other is a *cross-boundary* experiment using utterances recorded in session 3 as references for test on sessions 6–10, where there is a channel/environmental mismatch between the reference and the test utterances. Again, we report only the results as the length of reference and test utterances is 10 and 5 s, respectively. Fig. 5 depicts the DET curves achieved by the SM-CDBN, our SM-RS, and GMM-MFCC in two experiments. From Fig. 5(a) and (b), it is observed that our SM-RS outperforms the SM-CDBN and GMM-MFCC on both KING and NKING in the within-boundary experiments. As shown in Fig. 5(c), however, our SM-RS is inferior to the GMM-MFCC on KING in the cross-boundary experiment although it outperforms the SM-CDBN in Fig. 5(d). In comparison to the baseline performance, the SM-CDBN outperforms GMM-MFCC on NKING in the within-boundary experiment, as shown in Fig. 5(b), but is inferior in other experiments, as illustrated in Fig. 5.

Here, we emphasize that our DNA has been trained on only a dataset without any channel and environmental mismatch so far, and hence there is no chance for learning the channel and environmental variabilities with contrastive losses, which is responsible for the poor performance in the cross-boundary experiment. From all SV experiments, however, we observe that contrastive losses work well in capturing speaker-specific characteristics when the reference and test data are of the same variability sources even for noisy narrow-band speech in NTIMIT and NKING. To verify our above analysis, we have carried out a preliminary experiment by retraining our DNA and GMM with utterances in sessions 1 and 6 instead of sessions 1 and 2, and found that the performance of our SM-RS with the newly achieved representation is comparable with that of GMM-MFCC with the previous cross-boundary experimental setting. Although we also found that the use of longer reference utterances of 30 s in our SM-RS outperforms GMMs trained on utterances of 60 s in the cross-boundary experiment, we believe that the twofold behaviors of contrast losses depending on training data raise a general issue to be discussed later on.

### *D. Speaker Segmentation (SS)*

SS is the task of detecting speaker change points in an audio stream so that the audio stream can be split into acoustically homogeneous segments where every segment contains one speaker only. As reviewed in [38], there are two typical techniques: model-based methods with prior knowledge on speakers who get involved in a conversation, and distance-based methods without the use of prior knowledge. In our SS experiments, we mainly focus on the comparison between a representation by our DNA and MFCCs with the SC technique, as described in Section II-C, by using a simple distance-based algorithm and do not consider additional performance improvement mechanisms used in more sophisticated SS algorithms [38], [39]. As the Bayesian information criterion (BIC) is widely used in SS, we also conduct comparative studies with the standard BIC technique [38], [40]. Here, we emphasize that our experiments are designed to simulate a scenario in which a representation by our DNA trained on a corpus off-line is applied to unknown audio streams for online SS with the cross-corpus and cross-language experimental protocol. Note that here we conduct a two-way cross-language experiment, as a representation learned from the CHN corpus is also used in the dataset generated from TIMIT.

Following the same experimental protocols used in the previous work, e.g., [38], [39], we make use of TIMIT and CHN corpora to create a number of audio streams. With TIMIT, we generate 25 audio streams. Each audio stream of about 40 s consists of 10 segments of variable lengths corresponding to different speakers whose utterances are randomly chosen to

TABLE II
PERFORMANCE (MEAN±STD) OF SS ON DIFFERENT AUDIO STREAM DATASETS

| Measure | TIMIT | | | | | | CHN | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BIC | MFCCs | RS-T | RS-K | RS-NK | RS-CHN | BIC | MFCCs | RS-T | RS-K | RS-NK | RS-CHN |
| FAR | 0.26±0.07 | 0.31±0.10 | **0.25±0.09** | 0.27±0.11 | 0.26±0.10 | 0.28±0.08 | 0.40±0.04 | 0.25±0.07 | 0.23±0.07 | 0.23±0.10 | 0.23±0.08 | **0.21±0.06** |
| MDR | 0.27±0.13 | 0.24±0.12 | **0.19±0.09** | 0.19±0.15 | 0.21±0.13 | 0.21±0.13 | 0.51±0.09 | 0.40±0.12 | 0.34±0.12 | 0.40±0.14 | 0.39±0.13 | **0.34±0.09** |
| $F_1$ | 0.68±0.11 | 0.67±0.11 | **0.74±0.12** | 0.73±0.10 | 0.72±0.12 | 0.71±0.12 | 0.44±0.07 | 0.60±0.12 | 0.66±0.12 | 0.61±0.15 | 0.62±0.13 | **0.68±0.08** |

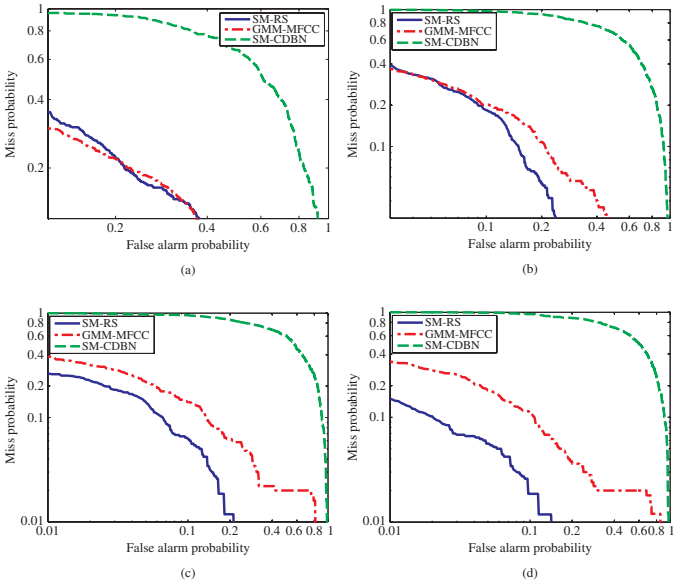


Fig. 4. SV performance of the SM-CDBM and the SM-RS versus that of GMM-MFCC on NTIMIT in terms of short test lengths. (a) DET curves for 1 s. (b) DET curves for 2 s. (c) DET curves for 3 s. (d) DET curves for 4 s.

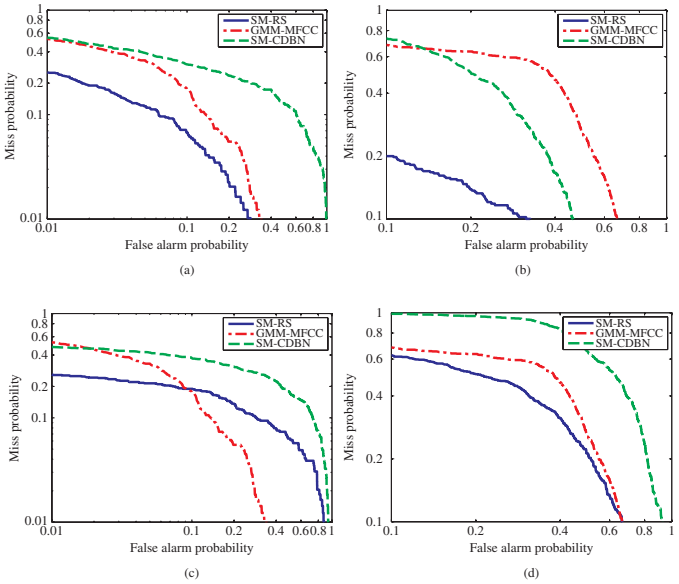


Fig. 5. SV performance of the SM-CDBM and the SM-RS versus that of GMM-MFCC on KING and NKING in the within- and cross-boundary experiments. (a) DET curves on KING (within-boundary). (b) DET curves on NKING (within-boundary). (c) DET curves on KING (cross-boundary). (d) DET curves on NKING (cross-boundary).

form segments ranging from 1.6 to 7.0 s, respectively. Totally, utterances of 250 speakers from TIMIT are used to generate 25 audio streams. Similarly, we create 15 audio streams with 50 speakers' utterances from the CHN corpus. Each audio stream of 62 s on average consists of 10 segments of variable lengths ranging from 3.0 to 9.0 s.

The distance-based algorithm [39] used in our experiments is summarized as follows. 1) Calculate the distance between two adjacent windows of a fixed size for an audio stream and forming a distance curve by sliding windows with a fixed increment through the audio stream. 2) Normalize and smoothen the distance curve with the low-pass filter to remove the sharp glitch. 3) Detect peaks from the smoothed distance curve with a threshold to find speaker change points. Note that parameters in the above algorithm are fixed for all the audio streams generated from a specific corpus no matter what representation is employed. For test, the detected speaker change points are aligned against the ground-truth speaker turn points. If a detected change point falls into the tolerance interval of a ground-truth turn point, this change point will be regarded as a correct detection. Otherwise, either a false alarm or a missing detection occurs. In our experiments, the same tolerance interval is applied in the distance-based and the BIC algorithms for all the audio streams.

For performance evaluation, we adopt three commonly used measures [38]–[40]: i.e., false alarm rate (FAR), miss detection rate (MDR), and $F_1$ measure. FAR and the MDR are defined as FAR $= N_{FA}/(N_{FA} + N_{GC})$ and MDR $= N_{MD}/N_{GC}$ where $N_{FA}$, $N_{MD}$, and $N_{GC}$ are the number of false alarms, miss detections and genuine speaker changes, respectively. Let $N_{CFC}$ and $N_{TDC}$ be the number of correctly found speaker changes and totally detected speaker changes, respectively. Based on the precision and the recall rates, the $F_1$ measure is defined as $F_1 = 2PR/(P + R)$ where the precision rate is $P = N_{CFC}/N_{TDC}$ and the recall rate is $R = N_{CRC}/N_{GC}$. Intuitively, a large FAR and MDR implies the poor performance, whilst a large $F_1$ indicates the good performance.

Table II tabulates all experimental results for audio streams generated from two corpora where the best performance corresponding to a specific corpus is highlighted by bold font. MFCCs are used in the BIC and the distance-based method for all audio streams, while four representations by our DNA trained on TIMIT, KING, NKING, and CHN, named RS-T, RS-K, RS-NK, and RS-CHN, are applied to the distance-based method for audio streams generated from TIMIT and CHN. It is observed from Table II that the use of the representation

by our DNA trained on TIMIT and CHN leads to the best performance on audio streams generated from TIMIT and CHN, respectively. Nevertheless, representations by our DNA trained on other corpora also yield satisfactory results in comparison to the BIC and MFCCs with the same SS algorithm. Thus, our experimental results suggest that representations by our DNA is promising for SS and expected to yield the improved performance by incorporating more sophisticated SS algorithms.

## IV. Discussion

In this section, we discuss issues raised by our work and relate our DNA to the previous work.

As reported in Section III, a variety of experiments on speaker-related tasks suggest that a representation by our DNA outperforms MFCCs and representations by alternative DAs. The success is attributed to the use of both contrastive and reconstruction losses. The contrastive losses tend to capture speaker-specific characteristics, while the reconstruction losses plays a critical role in regularization by avoiding information loss and normalizing non-speaker variabilities in speech signals. In particular, the splitting code mechanism in the code layer of our DNA has greatly facilitated the implementation. In contrast, our experiments reveal that the use of either contrastive or reconstruction losses only does not lead to the satisfactory performance in general. The use of contrastive losses only actually results in an aggressive supervised learning strategy on the frame level of speech. As there is an inextricable relationship between speaker-specific and non-speaker-related information, there is too much speaker and other variability on the frame level. Unlike the previous work in face verification [22], such an strategy itself is unlikely to capture the intrinsic speaker-specific characteristics with a small training set as observed from our experiments. Nevertheless, we observe that the use of contrastive losses in our DNA generally yields a better representation on noisy narrow-band corpora in contrast to the baseline performance. It is because channel noise, as a different source, was added to speech during recording, and the minimization of contrast losses on the code layers of two subnets tends to cancel out the distortion from the same variability sources along with learning speaker-specific characteristics if the input of two subsets corresponds to utterances recorded under the same condition. On the other hand, the reconstruction losses are designed to generate an overcomplete speech representation rather than capture speaker-specific characteristics in essence. Our experimental results suggest that such an overcomplete representation achieved with unsupervised learning facilitates disentangling speaker-specific from non-speaker-related information in a new representation. However, our observations suggest that the use of unsupervised learning alone, e.g., the CDBN [25] and the AE, leads to a generic yet novel representation containing all types of speech information other than a speaker-specific representation, which is evident from the comparative studies in this paper and results reported in [25]. In addition, our results, along with those not reported in this paper due to the limited space, also lend further support to the hybrid learning strategy for deep learning [13], [15] and its empirical justification [18], [19], given the fact that our DNA initialized with the pretraining considerably outperforms itself initialized randomly even though the same discriminative learning algorithm described in Section II-B.2 is applied.

Our work was limited by only a few of speech corpora available for us but reveals that learning a universal speaker-specific representation is still a challenging problem as a number of factors could affect such a learning task. First of all, as a data-driven methodology, deep learning requires sufficient training data but it is very difficult to collect them, ideally, a training dataset should cover all possible variabilities in speech including not only speaker-specific but also other non-speaker variabilities, as demonstrated in our SV experiments on KING/NKING where there are not only speaker-specific but also channel variabilities across different sessions. Next, there is a huge structure hypothesis space for our DNA and, to our knowledge, there are only very few empirical studies in model selection of DAs [18]. In particular, the hybrid learning strategy makes this problem more challenging, it is generally unclear how to determine the structure of a DA during pretraining and whether altering the structure of a DA after pretraining would compromise the pretraining effect. In addition, the role of the pretraining strategy [13], [16] is not well understood, although it practically works very well and some hypothesis were made [19] recently. Finally, we use statistics of a representation by our DNA to model speakers in our work. To improve the performance, we could modify our loss functions in (3b) to explicitly learn speaker-specific characteristics in terms of statistics, possibly including higher order ones, as required in speaker modeling along with a proper distance metric. In our ongoing research, we are studying the aforementioned problems in terms of learning speaker-specific characteristics. We anticipate that our studies will help understand DAs and the hybrid learning strategy toward discovering a universal speaker-specific representation.

Our DNA is especially motivated by the previous work [20], where a similar DA was proposed in the context of learning a nonlinear embedding to find a parsimonious representation for handwritten digits. Apart from the difference in building blocks, namely, restricted Boltzmann machine versus autoassociator, the two architectures differ in loss functions and motivations. In [20], neighborhood component analysis (NCA) [41] is employed to yield a representation that minimizes the variability of a digit class. Thanks to the probabilistic formulation of the NCA, there is no need to consider the interclass variability explicitly. However, NCA is inappropriate to our requirements as there are so many examples in one class and, instead, we employ contrastive losses to minimize both intra- and interclass variabilities simultaneously. On the other hand, the intrinsic topological structure is the major information source in a handwritten digit given the fact that the deep AE itself can yield an adequate representation [12], [13], [15], [20]. Thus, the use of the NCA in [20] simply reinforces the topological invariant by minimizing the variabilities incurred by geometric transformations and noise distortion with a small amount of labeled data [20]. In our work, however, speaker-specific information is minor in speech in comparison with

non-speaker-related information and hence a large amount of labeled data covering all kinds of variabilities in speech is required during discriminative learning despite the pretraining.

## V. Conclusion

We have proposed a deep learning architecture for learning intrinsic speaker-specific characteristics. Our architecture working on the regularized discriminative learning leads to a speaker-specific overcomplete representation. With a simple SC technique, we demonstrated that a representation learned by our DNA can capture intrinsic speaker-specific characteristics and generally outperform MFCCs by incorporating a state-of-the-art speaker modeling technique in various speaker-related tasks. Moreover, our work presented in this paper also reveals a number of challenges in discovering a universal speaker-specific representation with a DA. In our ongoing work, we are investigating those challenging problems.

In a broader sense, we argue that speech Information Component Analysis (ICA) becomes critical to overcome one of main obstacles that prevents a speech information processing system from achieving a higher performance, i.e., the use of proper speech ICA techniques would result in task-specific speech representations to improve the performance radically. This paper has demonstrated that speech ICA via learning is feasible. Moreover, deep learning could be a promising yet effective methodology for speech ICA.

## Appendix

Let $W$ denote the connection weight matrix between the input and the hidden layers in an autoassociator. Accordingly, $W^T$ is the weight matrix between the hidden and the output layers. Let $\boldsymbol{b}_h$ and $\boldsymbol{b}_o$ denote biases of the hidden and the output layers, respectively. For a distorted input $\tilde{\boldsymbol{x}}$, output of the hidden and the output layers are $\boldsymbol{h}(\tilde{\boldsymbol{x}}) = \sigma[\boldsymbol{u}_h(\tilde{\boldsymbol{x}})]$ and $\hat{\mathbf{x}} = \boldsymbol{u}_o(\tilde{\boldsymbol{x}})$ for the first hidden layer or $\hat{\mathbf{x}} = \sigma[\boldsymbol{u}_o(\tilde{\boldsymbol{x}})]$ for hidden layer $k$, $k = 2, \ldots, K$, respectively, where $\boldsymbol{u}_h(\tilde{\boldsymbol{x}}) = W\tilde{\boldsymbol{x}} + \boldsymbol{b}_h$ and $\boldsymbol{u}_o(\tilde{\boldsymbol{x}}) = W^T\boldsymbol{h}(\tilde{\boldsymbol{x}}) + \boldsymbol{b}_o$.

Given a training example $(\tilde{\boldsymbol{x}}, \boldsymbol{x})$, we have the gradient for the cost function in (6) with respect to $\boldsymbol{u}_o(\tilde{\boldsymbol{x}})$

$$\frac{\partial L_{dec}(\tilde{\boldsymbol{x}}, \boldsymbol{x})}{\partial \boldsymbol{u}_o(\tilde{\boldsymbol{x}})} = 2(\hat{\boldsymbol{x}} - \boldsymbol{x})\sigma'[\boldsymbol{u}_o(\tilde{\boldsymbol{x}})]. \tag{A.1}$$

Based on (A.1), we have the gradient

$$\frac{\partial L_{dec}(\tilde{\boldsymbol{x}}, \boldsymbol{x})}{\partial \boldsymbol{h}(\tilde{\boldsymbol{x}})} = W\frac{\partial L_{dec}(\tilde{\boldsymbol{x}}, \boldsymbol{x})}{\partial \boldsymbol{u}_o(\tilde{\boldsymbol{x}})}. \tag{A.2}$$

By using the chain rule, we achieve the gradient

$$\frac{\partial L_{dec}(\tilde{\boldsymbol{x}}, \boldsymbol{x})}{\partial \boldsymbol{u}_h(\tilde{\boldsymbol{x}})} = \left(h_j(\tilde{\boldsymbol{x}})[1 - h_j(\tilde{\boldsymbol{x}})]\frac{\partial L_{dec}(\tilde{\boldsymbol{x}}, \boldsymbol{x})}{\partial h_j(\tilde{\boldsymbol{x}})}\right)_{j=1}^{|\boldsymbol{h}(\tilde{\boldsymbol{x}})|}. \tag{A.3}$$

Here, $h_j(\tilde{\boldsymbol{x}})$ is the $j$th element of $\boldsymbol{h}(\tilde{\boldsymbol{x}})$. Similarly, we have the gradient with respect to biases

$$\frac{\partial L_{dec}(\tilde{\boldsymbol{x}}, \boldsymbol{x})}{\partial \boldsymbol{b}_o(\tilde{\boldsymbol{x}})} = \frac{\partial L_{dec}(\tilde{\boldsymbol{x}}, \boldsymbol{x})}{\partial \boldsymbol{u}_o(\tilde{\boldsymbol{x}})} \tag{A.4}$$

and

$$\frac{\partial L_{dec}(\tilde{\boldsymbol{x}}, \boldsymbol{x})}{\partial \boldsymbol{b}_h(\tilde{\boldsymbol{x}})} = \frac{\partial L_{dec}(\tilde{\boldsymbol{x}}, \boldsymbol{x})}{\partial \boldsymbol{u}_h(\tilde{\boldsymbol{x}})}. \tag{A.5}$$

Based on (A.1)–(A.5), we apply the gradient descent method and tied weights to achieve update rules as follows:

$$W \leftarrow W - \epsilon\left(\frac{\partial L_{dec}(\tilde{\boldsymbol{x}}, \boldsymbol{x})}{\partial \boldsymbol{u}_h(\tilde{\boldsymbol{x}})}\tilde{\boldsymbol{x}}^T + \boldsymbol{h}(\tilde{\boldsymbol{x}})\left[\frac{\partial L_{dec}(\tilde{\boldsymbol{x}}, \boldsymbol{x})}{\partial \boldsymbol{u}_o(\tilde{\boldsymbol{x}})}\right]^T\right) \tag{A.6}$$

$$\boldsymbol{b}_o = \boldsymbol{b}_o - \epsilon\frac{\partial L_{dec}(\tilde{\boldsymbol{x}}, \boldsymbol{x})}{\partial \boldsymbol{u}_o(\tilde{\boldsymbol{x}})} \tag{A.7}$$

and

$$\boldsymbol{b}_h = \boldsymbol{b}_h - \epsilon\frac{\partial L_{dec}(\tilde{\boldsymbol{x}}, \boldsymbol{x})}{\partial \boldsymbol{u}_h(\tilde{\boldsymbol{x}})}.$$

Here, $\epsilon$ is a learning rate.

Instead of the online learning, the SBP algorithm [14] applies the same rules as described above but updates parameters by using the averaging gradient over a batch of training examples, i.e., a speech segment of several frames in our work.

## Acknowledgment

The authors would like to thank the Associate Editor and anonymous reviewers for their valuable comments that improved the presentation of this paper and also G. Hinton and H. Larochelle for personal communications. We are also grateful to H. Lee for providing the convolutional deep belief network code and L. Wang for providing their Chinese corpus, both of which were used in our experiments.

## References

[1] J. Campbell, "Speaker recognition: A tutorial," *IEEE Proc.*, vol. 85, no. 8, pp. 1437–1462, Sep. 1997.
[2] J. P. Campbell, W. Shen, W. M. Campbell, R. Schwartz, J. F. Bonastre, and D. Matrouf, "Forensic speaker recognition," *IEEE Signal Proc. Mag.*, vol. 26, no. 2, pp. 95–103, Mar. 2009.
[3] X. Huang, A. Acero, and H. Hon, *Spoken Language Processing*. Englewood Cliffs, NJ: Prentice Hall, 2001.
[4] Q. Jin, "Robust speaker recognition," Ph.D. thesis, School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, 2007.
[5] R. Mammone, X. Zhang, and R. Ramachandran, "Robust speaker recognition: A feature-based approach," *IEEE Signal Process. Mag.*, vol. 13, no. 5, pp. 1–58, Sep. 1996.
[6] D. A. Reynolds, "Speaker Identification and verification using Gaussian mixture speaker models," *Speech Commun.*, vol. 17, nos. 1–2, pp. 91–108, Aug. 1995.
[7] D. A. Reynolds, "Speaker Identification and verification using Gaussian mixture speaker models," *MIT Lincoln Lab. J.*, vol. 8, pp. 173–191, Jan. 1995.
[8] S. Kajarekar, "Analysis of variability in speech with applications to speech and speaker recognition," Ph.D. thesis, School Sci. Eng., Oregon Graduate Institute Sci. Eng., Portland, OR, 2002.
[9] V. Laparra, G. Camps-Valls, and J. Malo, "Iterative gaussianization: From ICA to random rotations," *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 537–549, Apr. 2011.
[10] Y. Bengio and Y. LeCun, "Scaling learning algorithms toward AI," in *Large Scale Kernel Machine*. Cambridge, MA: MIT Press, 2007, pp. 321–360.
[11] Y. Bengio, "Learning deep architectures for AI," *Foundations Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
[12] G. E. Hinton, "Learning multiple layers of representation," *Trends Cogn. Sci.*, vol. 11, no. 10, pp. 428–434, Oct. 2007.
[13] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient based learning applied to document recognition," *IEEE Proc.*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[15] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.
[16] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Adv. Neural Inform. Process. Syst.*, vol. 19. 2007, pp. 1–8.
[17] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *Proc. IEEE 12th Int. Conf. Comput. Vision*, Kyoto, Japan, Sep.–Oct. 2009, pp. 2146–2153.
[18] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *J. Mach. Learn. Res.*, vol. 10, pp. 1–40, Jan. 2009.
[19] D. Erhan, Y. Bengio, A. Courville, P. Manzagol, and P. Vincent, "Why does unsupervised pre-training help deep learning?," *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, Feb. 2010.
[20] R. Salakhutdinov and G. Hinton, "Learning a non-linear embedding by preserving class neighbourhood structure," in *Proc. Art. Intell. Statis.*, vol. 2. 2007, pp. 412–419.
[21] M. Osadchy, Y. LeCun, and M. Miller, "Synergistic face detection and pose estimation with energy-based models," *J. Mach. Learn. Res.*, vol. 8, pp. 1197–1215, May 2007.
[22] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Computr. Soc. Conf. Comput. Vision Pattern Recognit.*, vol. 1. Jun. 2005, pp. 539–546.
[23] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Proc. IEEE Computr. Soc. Conf. Comput. Vision Pattern Recognit.*, vol. 2. Jul. 2004, pp. 97–104.
[24] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 473–480.
[25] H. Lee, Y. Largman, P. Pham, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Proc. Adv. Neural Informat. Process. Syst.*, vol. 22. 2010, pp. 1–9.
[26] Linguistic Data Consortium (LDC), Philadelphia, PA [Online]. Available: http://www.ldc.upenn.edu.com
[27] *Russian Speech Corpus* [Online]. Available: http://www.repository.voxforge1.org/
[28] L. Wang, "Chinese speech corpus for speaker recognition," Shenzhen Inst. Advanced Technology, Chinese Academy Science, Shenzhen, China, Tech. Rep., pp. 1–66, 2008.
[29] Y. LeCun and F. J. Huang, "Loss functions for discriminative training of energy-based models," in *Proc. Art. Intell. Statist.*, 2005, pp. 1–8.
[30] J. Bromley, I. Guyon, Y. LeCun, E. Sackinger, and R. Shah, "Signature verification using a siamese time delay neural network," in *Proc. Adv. Neural Inform. Process. Syst.*, vol. 5. 1993, pp. 1–3.
[31] K. Chen, "Toward better making a decision in speaker verification," *Pattern Recog.*, vol. 36, no. 2, pp. 329–346, Feb. 2003.
[32] L. Wang, K. Chen, and H. Chi, "Capture inter-speaker information with a neural network for speaker identification," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 436–445, Mar. 2002.
[33] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. Art. Intell. Statis.*, 2007, pp. 1–8.
[34] W. Campbell, D. E. Sturim, and Z. Karam, "Speaker comparison with inner product discriminant functions," in *Proc. Adv. Neural Inform. Process. Syst.*, vol. 22. 2010, pp. 1–9.
[35] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 1, pp. 72–83, Jan. 1995.
[36] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Pryzbocki, "The DET curve in assessment of detection task performance," in *Proc. Eurospeech*, vol. 4. 1997, pp. 1899–1903.
[37] S. Zhang and M. Mak, "Optimized discriminative kernel for SVM scoring and its application to speaker verification," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 173–185, Feb. 2011.
[38] M. Kotti, V. Moschou, and C. Kotropoulos, "Speaker segmentation and clustering," *Signal Process.*, vol. 88, no. 5, pp. 1091–1124, May 2008.
[39] P. Delacourt and C. Wellekens, "DISTBIC: A speaker-based segmentation for audio data indexing," *Speech Commun.*, vol. 32, nos. 1–2, pp. 111–126, Sep. 2000.
[40] S. S. Chen and P. S. Gopalakrishnan, "Speaker environment and channel change detection and clustering via Bayesian information criterion," in *Proc. Defense Adv. Res. Projects Agency Speech Recognit. Workshop*, 1998, pp. 127–132.
[41] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood component analysis," in *Proc. Adv. Neural Inform. Process. Syst.*, vol. 17. 2005, pp. 1–8.

**Ke Chen** (M'97–SM'00) received the B.Sc., M.Sc., and Ph.D. degrees in computer science in 1984, 1987, and 1990, respectively.

He has been with the University of Manchester, Manchester, U.K., since 2003. Earlier, he was with the University of Birmingham, Birmingham, U.K., Peking University, Beijing, China, Ohio State University, Columbus, Kyushu Institute of Technology, Fukuoka, Japan, and Tsinghua University, Beijing. He was a Visiting Professor with Microsoft Research Asia, Beijing, in 2000, and Hong Kong Polytechnic University, Kowloon, Hong Kong, in 2001. He has published more than 100 academic papers in refereed journals and conference proceedings. His current research interests include pattern recognition, machine learning, machine perception, and computational cognitive systems.

Dr. Chen is a Technical Program Co-Chair of the International Joint Conference on Neural Networks (IJCNN'12) and has been a member of the Technical Program Committees of numerous international conferences including CogSci and IJCNN. He chaired the IEEE Computational Intelligence Society (IEEE CIS) Intelligent Systems Applications Technical Committee (ISATC) and the University Curricula Subcommittee, in 2008 and 2009. He also served as Task Force Chairs and a member of NNTC, ETTC, and DMTC in IEEE CIS. He is a recipient of several academic awards including the NSFC Distinguished Principal Young Investigator Award and the JSPS Research Award. He is a member of IEEE CIS and International Neural Network Society. He has been on the editorial boards of several academic journals including the IEEE TRANSACTIONS ON NEURAL NETWORKS from 2005 to 2010, and serves as the Category Editor of *Pattern Recognition in Scholarpedia*.

**Ahmad Salman** received the B.E. and M.Sc. degrees in electrical engineering from the National University of Sciences and Technology, Rawalpindi, Pakistan, in 2003 and 2007, respectively. He is currently pursuing the Ph.D. degree with the School of Computer Science, University of Manchester, U.K.

His current research interests include speech information processing and machine learning.