Neural Processing Letters **3**: 81–89, 1996.

Text-Dependent Speaker Identification Based on Input/Output HMMs: An Empirical Study

KE CHEN, DAHONG XIE and HUISHENG CHI
National Laboratory of Machine Perception and Center for Information Science, Peking University, Beijing 100871, China
E-mail: chen@cis.pku.edu.cn

Key words: speaker identification, input/output HMM, EM algorithm, temporal processing

Abstract. In this paper, we explore the *Input/Output HMM* (IOHMM) architecture for a substantial problem, that of text-dependent speaker identification. For subnetworks modeled with generalized linear models, we extend the IRLS algorithm to the M-step of the corresponding EM algorithm. Experimental results show that the improved EM algorithm yields significantly faster training than the original one. In comparison with the multilayer perceptron, the dynamic programming technique and hidden Markov models, we empirically demonstrate that the IOHMM architecture is a promising way to text-dependent speaker identification.

1. Introduction

Speaker identification task is to classify an unlabeled voice token as belonging to one of a set of N reference speakers [1]. It is a very hard problem since a speaker's voice changes in time. There have been extensive studies in this field based upon conventional techniques of speech signal processing [2]. Recently, the neural computing techniques have been investigated to improve the classification performance [3]. It is well known that the temporal information or sequence effect plays a crucial role in speech processing. In previous researches, a dynamic programming technique called *Dynamic Time Warping* (DTW) [4] was proposed to handle the sequence effects with a template matching method in speech processing. But the performance of the DTW is unsatisfactory in speaker identification since a speaker's voice greatly changes in different time and environments so that the testing data may be rather different from the templates. In neural network community, some temporal processing techniques have been developed such as recurrent networks and time-delay techniques etc. Unfortunately, most of those techniques either cannot capture long-term temporal information [5] or suffer from high computational burdens [6].

In the recent research of neural computing, Bengio et al. proposed a recurrent architecture having a modular structure based upon the *Mixtures of Experts* (ME) architecture and *Hidden Markov Model* (HMM). *Expectation-Maximization* (EM) algorithm with the supervised learning paradigm is also employed in the model for training. Unlike standard HMMs which only learn the output sequence distribution and trained by an unsupervised EM algorithm, the model can be used to learn map input sequences to output sequences using the same processing style as recurrent neural networks. So it is called *Input/Output HMM* (IOHMM) [7]. An utterance of speaker may be viewed as a time-series. In text-dependent speaker identification, the text in both training and testing is the same or is known. Thus, the utterance of a fixed text naturally becomes a sequence consisting of successive feature vectors after preprocessing and feature extraction, which results in that text-dependent speaker identification becomes a problem of sequence recognition. In previous researches, indeed, this idea has already been used to attack problems of text-dependent speaker recognition [2, 8] based upon different temporal processing techniques, such as HMMs and the DTW. In this paper, based upon the idea, we explore the application of an alternative technique, IOHMM, to text-dependent speaker identification. In comparison with HMMs and the DTW, the IOHMM allows to take into account inter-speaker information for classification. Different structures of IOHMMs have been empirically investigated and experimental results indicate that the IOHMM is a promising technique to attack the text-dependent speaker identification problem in comparison with other techniques such as HMMs, the DTW and the Multi-Layer Perceptron (MLP). On the other hand, we also show that the M-step of the EM algorithm in the IOHMM is still an Iterative Reweighted Least Square (IRLS) problem [9, 10] when the statistical structure of its subnetworks can be modeled by *Generalized Linear Model* (GLIM) theory [10]. Accordingly, we extend the IRLS algorithm in [9] to the M-step of the EM algorithm for the IOHMM instead of the original *gradient ascent* method in [7]. Experiments demonstrate that the improved EM algorithm yields significantly faster training than the original one.

The remainder of the paper is organized as follows. Section 2 briefly reviews the IOHMM architecture and introduces the IRLS algorithm to the EM algorithm in the IOHMM for training. Section 3 presents experimental results and conclusions are drawn in the final section.

2. IOHMM Architecture and the Improved EM Algorithm

2.1. The IOHMM Architecture

The IOHMM can be modeled by a discrete state dynamical system based upon the state space description: $\mathbf{y}_t = g(x_t, \mathbf{u}_t)$, $x_t = f(x_{t-1}, \mathbf{u}_t)$ where $\mathbf{u}_t \in R^m$ is the input vector at time t, $\mathbf{y}_t \in R^q$ is the output vector, and $x_t \in \{1, 2, \cdots, n\}$ is a discrete state. Moreover, admissible state transitions will be specified by a directed graph $\mathcal{G}$ whose vertices correspond to the model's states and the set of successors of state j in $\mathbf{S}_j$. Bengio et al. model such a system as the recurrent architecture [7] illustrated in Figure 1. The architecture consists of a set of state networks N_j, $j = 1, \cdots, n$ and a set of output networks O_j, $j = 1, \cdots, n$. Each one of the state and output networks is uniquely associated to one of the states, and all networks share the same input $\mathbf{u}_t$. Each state network N_j has the task of predicting

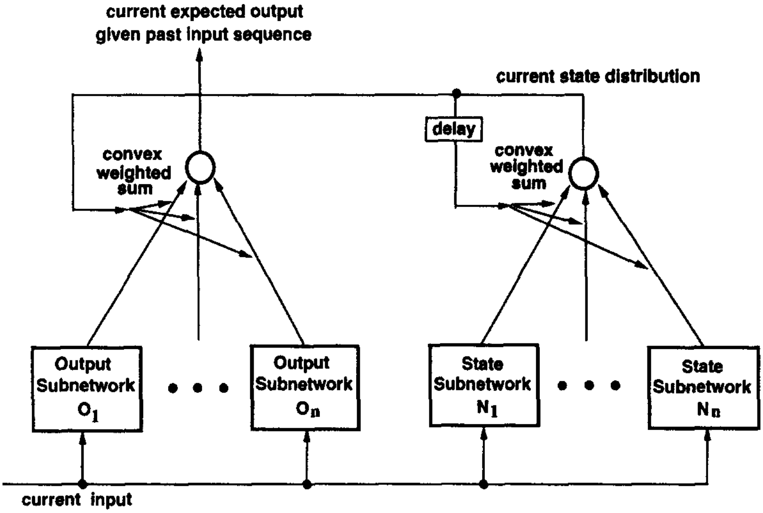

Figure 1. The input/output HMM architecture.

the next state distribution, based on the current input and given that $x_{t-1} = j$. Similarly, each output network O_j predicts the output of the system, given the current state and input. All the subnetworks are assumed to be static. At time t, each output $\varphi_{ij,t}$ of the state subnetwork N_j on the input $\mathbf{u}_t$ is associated with one of the successor i of state j as $\varphi_{ij,t} = e^{a_{ij,t}} / \sum_{k \in \mathbf{S}_j} e^{a_{kj,t}}$, $j = 1, \cdots, n, i \in \mathbf{S}_j$ where $a_{ij,t}$ are intermediate variables that can be thought of as the activations of the output units of subnetwork N_j. In addition, the condition $\varphi_{ij,t} = 0$ is also imposed for each $i \notin \mathbf{S}_j$. In this way, $\sum_{i=1}^{n} \varphi_{ij,t} = 1 \, \forall j, t$. Let vector $\vec{\zeta_t} \in R^n$ denote the internal state of the model and it can be interpreted as the current state distribution. It is computed through the previously computed internal state as $\vec{\zeta_t} = \sum_{j=1}^{n} \zeta_{j,t-1} \vec{\varphi}_{j,t}$ where $\vec{\varphi}_{j,t} = [\varphi_{1j,t}, \cdots, \varphi_{nj,t}]^T$ when the $\mathbf{u}_t$ is input. Output networks compete to predict the global output of the system $\vec{\eta}_t \in R^q$ on the input $\mathbf{u}_t$: $\vec{\eta}_t = \sum_{j=1}^{n} \zeta_{j,t} \vec{\eta}_{j,t}$ where $\vec{\eta}_{j,t} \in R^q$ is the output of subnetwork O_j. This connectionist architecture can be also interpreted as a probability model. Let us assume a multinomial distribution for the x_t and initialize the vector $\vec{\zeta_0}$ to positive numbers summing to one. The probabilistic interpretation of N_j is as follows:

$$\begin{aligned} &P(x_t = i | \mathbf{u}_1^t) = \textstyle\sum_{j=1}^{n} P(x_t = i | x_{t-1} = j, \mathbf{u}_t) P(x_{t-1} = j | \mathbf{u}_1^{t-1}), \\ &\varphi_{ij,t} = P(x_t = i | x_{t-1} = j, \mathbf{u}_t) \end{aligned} \tag{1}$$

Accordingly, the output $\vec{\eta}_t$ of this architecture can be interpreted as a 'position parameter' for the probability distribution of the output $\mathbf{y}_t$. However, in additional to being conditional on an input $\mathbf{u}_t$, this expectation is also conditional on the state x_t: $\eta_t = E[\mathbf{y}_t | x_t = i, \mathbf{u}_t]$. The actual form of the output distribution, denoted $f_Y(\mathbf{y}_t; \vec{\eta}_t)$, will be chosen according to the given task.

2.2. The Improved EM Algorithm

For the recurrent architecture, Bengio et al. have proposed an EM algorithm using a supervised paradigm for parameter estimation [7]. Consider the training data are a set of P pairs of input/output sequences of length T_p: $D = \{(\mathbf{u}_1^{T_p}(p), \mathbf{y}_1^{T_p}(p));\ p = 1, \cdots, P\}$. Let Θ denote the set of all parameters in the architecture. The likelihood function is

$$L(\Theta; D) = \prod_{p=1}^{P} P(\mathbf{y}_1^{T_p}(p)|\mathbf{u}_1^{T_p}(p); \Theta) \tag{2}$$

To derive the learning equations with EM algorithm, let us define the *complete data* through introducing hidden state paths $\mathcal{X} = \{x_1^{T_p}(p);\ p = 1, \cdots, P\}$ (describing a path in state space for each sequence) as $D_c = \{(\mathbf{u}_1^{T_p}(p), \mathbf{y}_1^{T_p}(p), x_1^{T_p}(p));\ p = 1, \cdots, P\}$. The corresponding complete data log-likelihood is

$$l_c(\Theta; D_c) = \sum_{p=1}^{P} \log P(\mathbf{y}_1^{T_p}(p), x_1^{T_p}(p)|\mathbf{u}_1^{T_p}(p); \Theta) \tag{3}$$

To simplify the presentation, we omit p below from Equation (3) such that we have

$$l_c(\Theta; D_c) = \sum_{p=1}^{P} \log P(\mathbf{y}_1^{T_p}, x_1^{T_p}|\mathbf{u}_1^{T_p}; \Theta)$$

As a result, EM algorithm is given by introducing the auxiliary function $Q(\Theta, \Theta^{(k)})$ and iterating the following two steps for $k = 1, 2, \cdots$:

E-step: Compute $Q(\Theta, \Theta^{(k)}) = E_{\mathcal{X}}[l_c(\Theta; D_c)|D, \Theta^{(k)}]$
M-step: Update the parameters as $\Theta^{(k+1)} \leftarrow \arg\max_{\Theta} Q(\Theta, \Theta^{(k)})$

In the E-step, for a sequence consisting of T components, computing $Q(\Theta, \Theta^{(k)})$ is the equivalent to computing *a posteriori* probabilities $h_{ij,t}$ as

$$h_{ij,t} = P(x_t = i, x_{t-1} = j|\mathbf{y}_1^T, \mathbf{u}_1^T) = \frac{\beta_{i,t}\alpha_{j,t-1}\varphi_{ij,t}}{\sum_i \alpha_{i,T}}$$

where

$$\alpha_{j,t-1} = P(\mathbf{y}_1^{t-1}, x_{t-1} = j|\mathbf{u}_1^{t-1}) = f_Y(\mathbf{y}_{t-1}; \vec{\eta}_{j,t-1}) \sum_k \varphi_{jk,t-1}\alpha_{k,t-2},$$

and

$$\beta_{i,t} = P(\mathbf{y}_t^T, x_t = i|\mathbf{u}_t^T) = f_Y(\mathbf{y}_t; \vec{\eta}_{i,t}) \sum_k \varphi_{ki,t+1}\beta_{k,t+1}.$$

In the M-step, in general, it can be completed by the *gradient ascent* method [7]. Unfortunately, it suffers from rather slow training. In practice, distributions in the exponential family can cover most of problems. The IRLS algorithm is an iterative algorithm for computing the maximum likelihood estimates of the parameters of a GLIM [9, 10]. Therefore, we can adopt the IRLS algorithm in the M-step if the statistical structure of all output subnetworks is modeled as the GLIM [9, 10], i.e. $f_Y(\mathbf{y}_t; \vec{\eta}_{i,t})$ is a distribution in the exponential family. Let $\Theta_i^{(s)}$ and $\Theta_i^{(o)}$, $i = 1, \cdots, n$ denote the parameters of state and output networks, respectively. Thus, the M-step becomes two separate maximization problems

$$\Theta_i^{(o)} = \arg\max_{\Theta_i^{(o)}} \sum_{p=1}^{P} \sum_{t=1}^{T_p} \sum_{i=1}^{n} \zeta_{i,t} \log f_Y(\mathbf{y}_t; \vec{\eta}_{i,t}) \tag{4}$$

$$\begin{aligned} \Theta_i^{(s)} &= \arg\max\nolimits_{\Theta_i^{(s)}} \textstyle\sum_{p=1}^{P} \sum_{t=1}^{T_p} \sum_{i=1}^{n} \sum_{j=1}^{n} h_{ij,t} \log \varphi_{ij,t} \\ &= \arg\max\nolimits_{\Theta_i^{(s)}} \textstyle\sum_{p=1}^{P} \sum_{t=1}^{T_p} \log(\prod_{i=1}^{n} \prod_{j=1}^{n} \varphi_{ij,t}^{h_{ij,t}}) \end{aligned} \tag{5}$$

Since $f_Y(\mathbf{y}_t; \vec{\eta}_{i,t})$ is a distribution of the exponential family and $\prod_{i=1}^{n} \prod_{j=1}^{n} \varphi_{ij,t}^{h_{ij,t}}$ is the multinomial distribution which is also a member of the exponential family, the IRLS algorithm can be used to solve those two problems. For the problem in Equation (4), we have

$$\Delta\Theta_{ir}^{(o)} = [\mathbf{U}^T W_{ir}^{(o)} \mathbf{U}]^{-1} \mathbf{U} \mathbf{e}^{(o)} \tag{6}$$

where $\Theta_{ir}^{(o)}$ denotes the parameter vector related to the rth output node of the i output network. The tth component of $\mathbf{e}^{(o)}$ is $e_{ir,t}^{(o)} = (y_{r,t} - \eta_{ir,t})/f'(\eta_{ir,t})$ and $f(\cdot)$ is the link function [10] of $f_Y(\mathbf{y}_t; \vec{\eta}_{i,t})$. $\mathbf{U}$ is the matrix consisting of all training data and its rows are the feature vectors of each utterances. $W_{ir}^{(o)}$ is a diagonal matrix whose tth diagonal element is $w_{ir,t} = [f'(\eta_{ir,t})]^2/Var(y_{r,t})$ and $Var(\cdot)$ is the variance function [10] of $f_Y(\mathbf{y}_t; \vec{\eta}_{i,t})$. In speaker identification, the classification is a specific multiway classification that the output is the binary vector with a single non-zero component. Therefore, we use the *generalized Bernoulli distribution* proposed in [6] as the probabilistic model of output networks. In [6], we have shown that the generalized Bernoulli distribution is a member of the exponential family. Accordingly, we have $f_Y(\mathbf{y}_t; \vec{\eta}_{i,t}) = \prod_{k=1}^{q} \eta_{ik,t}^{y_{k,t}} (1 - \eta_{ik,t})^{1-y_{k,t}}$ such that $e_{ir,t}^{(o)} = \zeta_{i,t}(y_{r,t} - \eta_{ir,t})$ and $w_{ir,t}^{(o)} = \zeta_{i,t}\eta_{ir,t}(1 - \eta_{ir,t})$. For the problem in Equation (5), using the IRLS algorithm, we have

$$\Delta\Theta_{ir}^{(s)} = [\mathbf{U}^T W_{ir}^{(s)} \mathbf{U}]^{-1} \mathbf{U} \mathbf{e}^{(s)} \tag{7}$$

where $\Theta_{ir}^{(s)}$ denotes the parameter vector related to the r output node of the i state network. The tth component of $\mathbf{e}^{(s)}$ is $e_{ir,t}^{(s)} = h_{ir,t} - \varphi_{ir,t}$. $W_{ir}^{(s)}$ is also a diagonal matrix whose tth diagonal element is $w_{ir.t}^{(s)} = \varphi_{ir,t}(1 - \varphi_{ir,t})$.

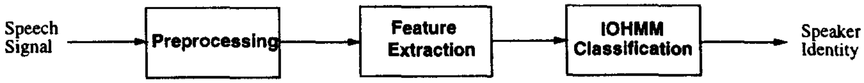

Figure 2. Scheme of the speaker identification system based on the IOHMM.

3. Experiments and Results

We have already developed a text-dependent speaker identification system in Sun Sparc II workstation. Scheme of the system is depicted in Figure 2.

The technical details of the acoustic preprocessing and feature extraction are briefly as follows: 1) 16-bit A/D-converter with 11.025 KHz sampling rate; 2) processing the data with a pre-emphasis filter $H(z) = 1 - 0.95z^{-1}$; 3) 16-order linear predictive (LP) analysis; 4) 256-point LP based FFT formed every 25.6ms using a Hamming analysis window without overlapping; 5) dividing channels from 0 Hz to 5.0125 KHz into 24 channels according to the knowledge on *critical bands* in [11], subtraction of the average from the components and normalization of the pattern vectors as follows: In each channel, the energy is accumulated and denoted as E_i, $(i = 1, 2, \cdots, 24)$. Furthermore, an entropy is also defined over each channel for producing a 24-order feature vector for each frame as follows,

$$I_i = -P_i \log P_i,\ P_i = \frac{E_i}{\sum_{j=1}^{24} E_j};\quad i = 1, 2, \cdots, 24. \tag{8}$$

In the current system, we choose 10 isolated digits from ‘0’ to ‘9’ as the fixed text. Depending upon the fixed text, as classifiers, 10 IOHMMs are established so that those 10 classifiers correspond to 10 digits from ‘0’ to ‘9’, respectively. The current acoustic database consists of 10 isolated digits from ’0’ to ‘9’ in Chinese and 10 male speakers are registered in the database. In the database, the utterances are recorded in three different sessions; each digit is uttered 10 times in each session. According to three different recording sessions, we naturally divide all data into three sets called Set-1, Set-2 and Set-3, respectively. After preprocessing and feature extraction, accordingly, we achieve three sets consisting of 24-order feature vectors. For evaluating the performance, we adopt two methods to train the IOHMM. One is the *single-session training* which simply uses data in Set-1 as training samples. The other is the *multi-session training* which uses 7 utterances in Set-1 and 3 utterances in Set-2 of each digit, respectively, for each speaker. As a result, the remainder of data in Set-2 is called Set-2*. During testing, we call results as Test-1, Test-2 and Test-1* when data in Set-2, Set-3 and Set-2* are used as testing sets, respectively.

Like the HMMs, it is still an open problem how to select an appropriate structure of the IOHMM. That is, for a given task, the number of subnetworks and the matrix of state transition in the IOHMM must be determined in advance. In the current work, we exhaustively search for an appropriate number of subnetworks from 4

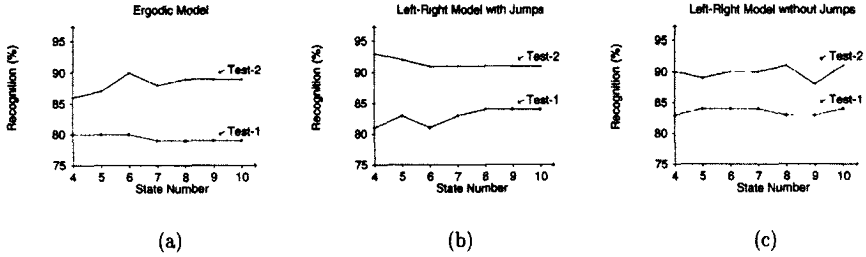

Figure 3. The results of (a) ergodic model, (b) left-right with jumps, (c) left-right without jump.

Table I. The identifying accuracies (%) of the IOHMM with jumps under single-session training.

Text	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	Averaging
Test-1	83.0	82.0	83.0	77.0	83.0	87.0	89.0	89.0	84.0	84.0	84.1
Test-2	92.0	94.0	91.0	85.0	84.0	89.0	82.0	81.0	80.0	85.0	86.3

to 10 and investigate three typical models often used in speech processing [12], i.e. *ergodic* model, *left-right* models including one without jump and one that no jumps of more than two states are allowed. For instance, we show all identifying accuracies on the digit '0' at different models and different states in Fig. 3 under the single-session training. According to averaging identifying accuracies, we find that the optimal numbers of the ergodic model, left-right with jumps and left-right without jumps are 6, 5 and 10, respectively. Using the optimal numbers for each chosen model, furthermore, we find that the performance of left-right models with jumps is better than ones of ergodic model and left-right model without jump under the single-session training. Due to the limited space here, we merely report results of 5-state IOHMM with jumps in both single-session and multi-session training. The results are shown in Table I and Table II, respectively.

It is worth pointing out that the use of IRLS algorithm in the M-step of EM algorithm yields significantly faster training than the gradient ascent method. In detail, 4 or 5 epoches (about 5 minutes) are merely needed to reach the steady state using the improved EM algorithm, while more than 800 epoches (about 5 hours) are usually necessary to reach the steady state using the original EM algorithm in [7].

Table II. The identifying accuracies (%) of the IOHMM with jumps under multi-session training.

Text	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	Averaging
Test-1*	97.1	97.1	97.1	94.3	98.6	87.2	97.1.0	98.6	92.9	94.3	95.4
Test-2	92.0	98.0	96.0	88.0	90.0	91.0	85.0	89.0	77.0	89.0	89.5

Table III. The identifying accuracies (%) of the HMM without jump under multi-session training.

Text	‘0’	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’	‘8’	‘9’	Averaging
Test-1*	97.2	95.1	87.3	86.5	86.2	89.4	87.3	99.1	77.1	88.4	91.4
Test-2	87.0	94.0	87.0	81.0	91.0	88.0	85.0	89.0	67.0	71.0	83.6

Table IV. The identifying accuracies (%) of the DTW under multi-session training.

Text	‘0’	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’	‘8’	‘9’	Averaging
Test-1*	92.2	90.1	88.3	83.5	82.8	87.7	86.4	94.7	75.9	89.2	87.1
Test-2	88.0	90.0	85.0	80.0	83.0	86.0	87.0	87.0	71.0	74.0	83.1

For the purpose of comparison, we also investigate some classic techniques in text-dependent speaker identification, i.e. HMMs [8], the DTW [4] and the MLP [3], on the same training and testing sets. For HMMs, we adopt the discrete HMM technique and the codebook consists of 256 codewords. We also investigated three kinds of HMMs, i.e. ergodic, left-right without jump and left-right with jumps [12]. As a result, the 6-state left-right HMM without jump achieved the best identifying accuracies which are shown in Table III. For the DTW technique, multiple feature vectors of utterances recorded in different sessions were employed as multiple templates of a speaker. Accordingly, for unknown utterance, the averaging score of results produced by comparing with multiple templates was used to identify the unknown speaker and the identifying accuracies are shown in Table IV. For MLPs, we used the 2-fold cross-validation technique to derive an optimal four-layer architecture with 24-20-20-40 for the given task. The identifying accuracies using such MLPs as classifiers are shown in Table V.

Based upon all aforementioned results, we could claim empirically that the IOHMM is better than those classic techniques in text-dependent speaker identification.

4. Conclusions

We have described an application of the Input/Output HMM to text-dependent speaker identification. We have also extended the IRLS algorithm [9] to the M-step of the EM algorithm in the IOHMM when the statistical structure of subsets

Table V. The identifying accuracies(%) of MLPs under multi-session training.

Text	‘0’	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’	‘8’	‘9’	Averaging
Test-1*	92.5	89.3	89.8	82.1	87.6	85.4	88.9	91.9	79.6	89.5	87.7
Test-2	88.0	90.0	87.0	85.0	82.0	83.0	84.0	83.0	77.0	89.0	84.8

could be modeled by the GLIM. Experimental results show that the IOHMM is a promising architecture which can deal with temporal information in text-dependent speaker identification. In comparison with the classic techniques, the system based on the IOHMM could achieve better performance. In our ongoing research, we shall explore the improved IOHMM architecture for capturing the long-term contextual information carried in a speaker's utterance in text-dependent speaker identification. In addition, we shall also apply the IOHMM to text-dependent speaker verification in our future work.

Acknowledgements

The authors are very grateful to two anonymous referees for their comments and also thank Z. Wu and W. Bao for their supplying results produced by HMMs. This work was partially supported by National Science Foundation of China with Grants 69571002 and 69475007 as well as the Climbing Program – National Key Project for Fundamental Research in China with Grant NSC 92097.

References

1. G.R. Doddington, "Speaker recognition – identifying people by their voices", Proc. IEEE, Vol. 73, pp. 1651–1664, 1986.
2. T. Matsui and S. Furui, "Speaker recognition technology", NTT Review, Vol. 7, No. 2, pp. 40–48, 1995.
3. Y. Bennani and P. Gallinari, "Connectionist approaches for automatic speaker recognition", Proc. ESCA Workshop on Automatic Speaker Recognition, Martigny, Switzerland, pp. 95–102, April 4–7, 1994.
4. H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for speech word recognition", IEEE Trans. Acoustics, Speech and Signal Processing, Vol. ASSP-26, No. 1, pp. 43–49, 1978.
5. Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult", IEEE Trans. Neural Networks, Vol. 5, No. 2, pp. 157–166, 1994.
6. K. Chen, D. Xie and H. Chi, "Speaker identification using time-delay HMEs", International Journal of Neural Systems, Vol. 7, No. 1 (March), pp. 29–43, 1996.
7. Y. Bengio and P. Frasconi, "An Input/Output HMM architecture", in J.D. Cowan, G. Tesauro, J. Alspector (eds) Advances in Neural Information Systems 7, MIT Press: Cambridge, MA, 1995.
8. S. Furui, "An overview of speaker recognition technology", Proc. ESCA Workshop on Automatic Speaker Recognition, Martigny, Switzerland, pp. 1–9, April 4–7, 1994.
9. M.I. Jordan and R.A. Jacobs, "Hierarchical mixtures of experts and EM algorithm", Neural Computation, Vol. 6, No. 2, pp. 181–214, 1994.
10. P. McCullagh and J.A. Nelder, Generalized Linear Models, Chapman and Hall: London, 1989.
11. K. Zwicker, "Subdivision of the audible frequency range into critical bands", J. Acoust. Soc. Amer., Vol. 35, No. 2, pp. 248–252, 1961.
12. L. Rabiner and B.H. Juang, Fundamentals of Speech Recognition, Prentice-Hall: Englewood Cliffs, NJ, 1993.