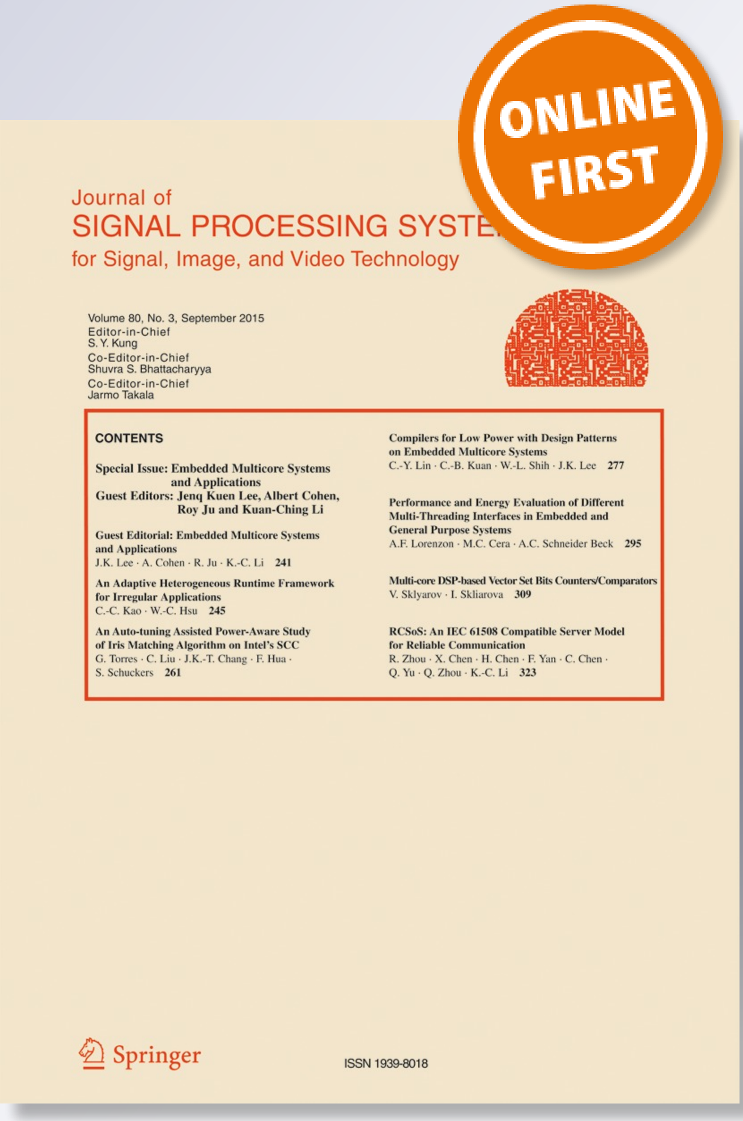
Discriminative Boosting Algorithm for Diversified Front-End Phonotactic Language Recognition

Wei-Wei Liu, Meng Cai, Wei-Qiang Zhang, Jia Liu & Michael T. Johnson

Journal of Signal Processing Systems
for Signal, Image, and Video Technology
(formerly the Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology)

ISSN 1939-8018

J Sign Process Syst
DOI 10.1007/s11265-015-1017-1



ONLINE FIRST

Journal of
SIGNAL PROCESSING SYSTE
for Signal, Image, and Video Technology

Volume 80, No. 3, September 2015
Editor-in-Chief
S. Y. Kung
Co-Editor-in-Chief
Shuvra S. Bhattacharyya
Co-Editor-in-Chief
Jarmo Takala

CONTENTS

Springer

ISSN 1939-8018

Springer

J Sign Process Syst
DOI 10.1007/s11265-015-1017-1

Discriminative Boosting Algorithm for Diversified Front-End Phonotactic Language Recognition

Wei-Wei Liu[1,2] · Meng Cai[1] · Wei-Qiang Zhang[1] · Jia Liu[1] · Michael T. Johnson[3]

Received: 14 November 2014 / Revised: 23 April 2015 / Accepted: 11 May 2015

Abstract Currently, phonotactic spoken language recognition (SLR) and acoustic SLR systems are widely used language recognition systems. Parallel phone recognition followed by vector space modeling (PPRVSM) is one typical phonotactic system for spoken language recognition. To achieve better performance, researchers assumed to extract more complementary information of the training data using phone recognizers trained for multiple language-specific phone recognizers, different acoustic models and acoustic features. These methods achieve good performance but usually compute at high computational cost and only using complementary information of the training data. In this paper, we explore a novel approach to discriminative vector space model (VSM) training by using a boosting framework to use the discriminative information of test data effectively, in which an ensemble of VSMs is trained sequentially. The effectiveness of our boosting variation comes from the emphasis on working with the high confidence test data to achieve discriminatively trained models. Our variant of boosting also includes utilizing original training data in VSM training. The discriminative boosting algorithm (DBA) is applied to the National Institute of Standards and Technology (NIST) language recognition evaluation (LRE) 2009 task and show performance improvements. The experimental results demonstrate that the proposed DBA shows 1.8 %, 11.72 % and 15.35 % relative reduction for 30s, 10s and 3s test utterances in equal error rate (EER) than baseline system.

Keywords Language recognition · Discriminative boosting algorithm (DBA)

1 Introduction

Spoken language recognition (SLR) has become an increasingly crucial technique for many applications such as search engines and language translation systems [1]. Without loss of generality, we can consider language recognition as a classification problem [2]. Given a set of training data and associated labels, the first step is to learn characteristics of languages from the training data, and then classify a speech utterance to the most probable language based on the language model. Currently, acoustic language recognition (LR) systems [3] and phonotactic LR systems [2] are both widely used.

Parallel phone recognition followed by vector space modeling (PPRVSM) is one typical phonotactic system for

This project is supported by National Natural Science Foundation of China (No.61273268, No. 61370034 and No. 61403224).

✉ Wei-Qiang Zhang
wqzhang@tsinghua.edu.cn

Wei-Wei Liu
liu-ww10@hotmail.com

[1] Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

[2] General Communication Station, Chinese General Logistics Department, Beijing 100842, China

[3] Department of Electrical and Computer Engineering, Marquette University, Milwaukee, Wisconsin, USA

Published online: 28 May 2015

spoken language recognition. In PPRVSM system, various phone recognizers are applied in parallel and fused at the confidence score level. Generally, the PPRVSM system can be developed in three ways. The first one is to use parallel phone recognizers on multiple language-specific speech data with different phone sets [2], eg. Hungarian, Czech and Russian phone-recognizers developed by the Brno University of Technology (BUT) [4]. The second way is to use phone recognizers on the same language-specific speech data with one phone set but using different acoustic models [5], like GMM-HMM, ANN-HMM [4] and DNN-HMM acoustic models. The third way is to use phone recognizers on the same language-specific speech data with one phone set but using different acoustic features, such as the Mel-frequency cepstral coefficients (MFCC) and perceptual linear prediction (PLP) features. All the three ways focuses on extract complementary information of the training data, but do not explore any information of the test data.

In fact, in actual test conditions, the training and test data are variable in speakers, background noise, channel conditions. To achieve higher robustness to variable test conditions, it is necessary to use the discriminative information of the useful resource of test data effectively. Recently, discriminative training techniques such as maximum mutual information (MMI) [6, 7], minimum phone error (MPE) [8], minimum classification error (MCE) [9] and heteroscedastic linear discriminant analysis [10] have been proposed and outperformed nondiscriminative models in language recognition task [11]. In language recognition, discriminative training focuses on defining the classification decision boundaries so that the equal error rate (EER) [12] can be decreased. But these training approaches are always computationally expensive and sometimes difficult to implement.

In this paper, we propose a boosting method using support vector machines (SVMs) as discriminative classifiers to build a regression backend for language recognition tasks. The discriminative boosting algorithm (DBA) methods uses a simple confidence criterion. The motivation of DBA is to find the high confidence test utterances and use them as training data, so that more useful discriminative information of the test database can be fully exploit and achieve better language recognition performance. Because the component classifiers have the same structure as the classifier in baseline system and are trained with the same criterion, little new implementation and additional computation are needed.

The rest of the paper is organized as follows. In Section 2 we review PPRVSM baseline language recognition system. The implementation of discriminative boosting algorithm method is introduced in Section 3. We also discuss variants and implementation of the algorithm. Experimental setup is described in Section 4. The results and discussions that DBA experimentally compared against the PPRVSM approach are in Section 5 followed by conclusions in Section 6.

2 PPRVSM Baseline System

In this work we use PPRVSM [1, 13] language recognition system as baseline system. The architecture of the system is shown in Fig. 1.

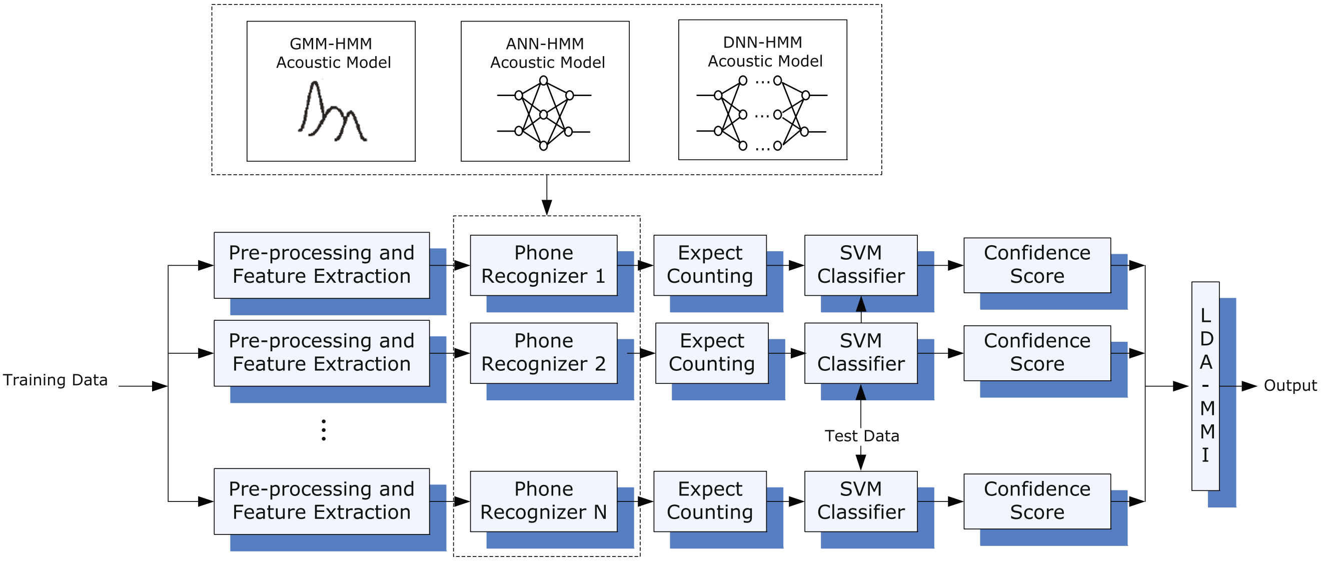


Figure 1 Architecture of phonotactic language recognition system.

Generally, the language recognition system maps the input data x to a high dimensional feature supervector as following:

$$\Phi : x \rightarrow \varphi(x). \tag{1}$$

Then the supervector $\varphi(x)$ is sent to the classifier and a decision is made based on the output of the classifier [14]. According to Fig. 1, the PPRVSM system comprises three main components: decoding, expect counting and the vector space modeling (VSM).

2.1 Decoding

In this system, phoneme recognizers are employed to convert the speech into phone lattices according to the given acoustic model, then the lattices are used to perform phonotactic analysis to classify languages in SVM. The phoneme recognizers are usually trained either on multiple language-specific speech data with different phone sets [2] or on the same language-specific speech data with one phone set but using different acoustic models [5]. In this paper, DNNs, GMMs and ANNs have been used to compute state observation probabilities for all tied states in the HMM set [15].

2.2 Expect Counting

Given the acoustic model Λ_{AM}, the expected counts over all possible hypotheses in the lattice ℓ of speech utterance X are computed as follows [14]:

$$\begin{aligned} & c_{\mathrm{E}}(h_i, ..., h_{i+N-1}|\ell) \\ = \; & E[c_{\mathrm{E}}(h_i, ..., h_{i+N-1})|X, \Lambda_{\mathrm{AM}}, M_{\mathrm{E}}] \\ = \; & \sum_{h_i...h_{i+N-1} \in \ell, h(e_i)=h_i} \left[\alpha(e_i)\beta(e_{i+N-1}) \prod_{j=i}^{i+N-1} \xi(e_j) \right], \end{aligned}$$

where acoustic model Λ_{AM} is language independent, M_{E} is the estimates of the N-gram probabilities that maximize $\sum_H f(X|H, \Lambda_{\mathrm{AM}})P(H|\mathcal{L})$ (H is an N-gram phone sequence, $H = h_i...h_{i+N-1}$, $\mathcal{L}$ is the language model of the language under consideration, $f(X|H, \Lambda_{\mathrm{AM}})$ is the likelihood of the speech utterance X given Λ_{AM} and H). $\alpha(e_i)$ is the forward probability of the starting node of edge e_i and $\beta(e_{i+N-1})$ is the backward probability of the ending node of edge e_{i+N-1}. $\xi(e_j)$ denotes the posterior probability of the edge e_j.

Then the probability of the phone sequence $h_i...h_{i+N-1}$ in the lattice is calculated as follows:

$$p(h_i...h_{i+N-1}|\ell) = \frac{c_{\mathrm{E}}(h_i...h_{i+N-1}|\ell)}{\sum_{\forall m} c_{\mathrm{E}}(h_m...h_{m+N-1}|\ell)}, \tag{2}$$

Let $d_i = h_i...h_{i+n-1}$ ($n <= N$), the probabilities of phonetic N-grams in the lattice ℓ can form a phonotactic feature supervector for the given utterance $\varphi(x)$:

$$\varphi(x) = [p(\mathbf{d_1}|\ell_x), p(\mathbf{d_2}|\ell_x), ..., p(\mathbf{d_F}|\ell_x)], \tag{3}$$

here $F = f_{\mathrm{n}}^N$ (f_{n} is the number of the phonemes of the frontend phone recognizer and N is the order of N-gram). ℓ_x denotes the lattice generated from data x by a phone recognizer. $p(\mathbf{d_q}|\ell_x)$ is the probability of the N-gram $\mathbf{d_q}$ in the lattice.

2.3 VSM

In PPRVSM, each spoken utterance is represented by a super-vector and then modeled using an SVM [13], the output score is computed as following:

$$f(\varphi(x)) = \sum_l \alpha_l K_{\mathrm{TFLLR}}(\varphi(x), \varphi(x_l)) + d, \tag{4}$$

here $\varphi(x_l)$ are support vectors that are trained using the Mercer condition. In this paper, SVM using term frequency log-likelihood ratio (TFLLR) kernel [16] are employed as back-end of the language recognition system. K_{TFLLR} is a TFLLR kernel computed as:

$$\begin{aligned} K_{\mathrm{TFLLR}}(\varphi(x_i), \varphi(x_j)) &= \sum_{q=1}^{F} p(\mathbf{d_q}|\ell_{x_i}) * p(\mathbf{d_q}|\ell_{x_j}) \\ &= \sum_{q=1}^{F} \frac{p(\mathbf{d_q}|\ell_{x_i})}{\sqrt{p(\mathbf{d_q}|\ell_{\mathrm{all}})}} * \frac{p(\mathbf{d_q}|\ell_{x_j})}{\sqrt{p(\mathbf{d_q}|\ell_{\mathrm{all}})}}, \end{aligned} \tag{5}$$

the $p(\mathbf{d_q}|\ell_{\mathrm{all}})$ is the observed probability of $\mathbf{d_q}$ across all lattices. In this work the training stage is always carried out with a one-versus-rest strategy.

3 Discriminative Boosting Algorithm SLR System

The discriminative boosting algorithm is proposed in this section. The DBA algorithm takes as input a training set of n utterances $\mathbf{Tr} = \{(x_i, y_i)|i = 1, 2, ..., n\}$ where x_i is the i-th input training data as described in Section 2. $y_i \in \mathbf{Y}$ is the class label associated with x_i. In this paper, it is assumed that the set of possible labels $\mathbf{Y} = \{l_k|k = 1, 2, ..., K\}$ is of finite cardinality K. The test set of m utterances is denoted as $\mathbf{Te} = \{x_{t_j}|j = 1, 2, ..., m\}$. $\mathbf{M} = \{\mathbf{M}_q|q = 1, 2, ..., Q\}$ denotes the set of language models of the system. $\mathbf{M}_q = \{\mathrm{mdl}_{qk}|k = 1, 2, ..., K\}$, in which mdl_{qk} is the language model of the q-th subsystem for k-th language. Because the training stage of language recognition is carried out with a one-versus-rest strategy, then y_i will be mapping to either 1 or -1. The DBA algorithm is described as follows:

a) **Initializing:** When class k is viewed as the target language, $\mathbf{Tr}$ will be updated to $\mathbf{Tr}' = \{(x_i, y_i')|i = 1, 2, ..., n\}$ where

$$y_i' = \begin{cases} +1, i = k \\ -1, i \neq k \end{cases} . \tag{6}$$

b) **Training:** Training the language model for the Q subsystems using the updated training database $\mathbf{Tr}'$. The language model matrix are

$$\mathbf{M} = \begin{bmatrix} \mathbf{M_1} \\ \mathbf{M_2} \\ \vdots \\ \mathbf{M_Q} \end{bmatrix} = \begin{bmatrix} \mathrm{mdl}_{11} & \mathrm{mdl}_{12} & \cdots & \mathrm{mdl}_{1K} \\ \mathrm{mdl}_{21} & \mathrm{mdl}_{22} & & \mathrm{mdl}_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{mdl}_{Q1} & \mathrm{mdl}_{Q2} & & \mathrm{mdl}_{QK} \end{bmatrix} . \tag{7}$$

c) **Testing:** In PR-SVM language recognition system, if an SVM is employed as the classifier, the output score matrix is computed as:

$$\mathbf{F} = \left[\mathbf{F_1 F_2} \cdots \mathbf{F_Q}\right]', \tag{8}$$

where

$$\mathbf{F_q} = \begin{bmatrix} f_q(\varphi(x_{t_1}))|_{\mathrm{mdl}_{q1}} & \cdots & f_q(\varphi(x_{t_1}))|_{\mathrm{mdl}_{qK}} \\ f_q(\varphi(x_{t_2}))|_{\mathrm{mdl}_{q1}} & \cdots & f_q(\varphi(x_{t_2}))|_{\mathrm{mdl}_{qK}} \\ \vdots & \ddots & \vdots \\ f_q(\varphi(x_{t_m}))|_{\mathrm{mdl}_{q1}} & \cdots & f_q(\varphi(x_{t_m}))|_{\mathrm{mdl}_{qK}} \end{bmatrix}, \tag{9}$$

where $f_q(x_{t_j})|_{\mathrm{mdl}_{qk}}$ is the confidence score of j-th test utterance of q-th subsystem based on the k-th language model computed using Eq. 3.

d) **Votes Counting:** Then the subsystems take a vote and decide which language every test utterance belongs to according the belief score. The votes counting matrix is defined as:

$$\mathbf{C_v} = [\mathbf{C_{v1} C_{v2}} \cdots \mathbf{C_{vm}}], \tag{10}$$

where $\mathbf{C_{vj}}$ is a k-dimensional supervector of votes counting of the which is computed as:

$$\mathbf{C_{v}}_j = \left[c_{j1} c_{j2} \cdots c_{jK}\right], \tag{11}$$

where

$$c_{jk} = \sum_{q=1}^{Q} v_{jqk}, \tag{12}$$

where v_{jqk} is the number of votes of q-th subsystem based on the k-th language model for the j-th test utterance, which is computed as:

$$v_{jqk} = \begin{cases} 1, & \text{if } \{f_q(\varphi(x_{t_j}))|_{\mathrm{mdl}_{qk}}\} > 0 \text{ and} \\ & \max\limits_{\forall p}\{f_q(\varphi(x_{t_j}))|_{\mathrm{mdl}_{qp,p\neq k}}\} < 0 \\ 0, & \text{otherwise} \end{cases} . \tag{13}$$

We select this criterion because the confidence score $f(\cdot)$ denotes the distance from the hyperplane of SVM classifier. The constrain $f_q(\varphi(x_{t_j}))|_{\mathrm{mdl}_{qk}} > 0$ and $\{f_j(\varphi(x_{t_j}))|_{\mathrm{mdl}_{qp,p\neq k}}\} < 0$ indicates a high confidence decision between target language and non-target language.

e) **Update Training Database:** Let $y_{t_j} = l_k$ if $c_{jk} > V$. Here V denotes the threshold. Then put (x_{t_j}, l_k) into new database $\mathbf{T_{DBA}}$. We describe two versions of the algorithm which we denote DBA-M1 and DBA-M2. The two versions are equivalent in their pre-processing, feature extracting and decoding steps and differ only in updating methods of new training database $\mathbf{Tr_{DBA}}$. $\mathbf{Tr_{DBA}}$ in DBA-M1 is only composed of high confidence test data, while in DBA-M2 is composed of both test data and original training data.

DBA-M1: $\mathbf{Tr}_{\mathrm{DBA}} = [\mathbf{T_{DBA}}]$
DBA-M2: $\mathbf{Tr_{DBA}} = [\mathbf{T_{DBA}} \;\; \mathbf{Tr}]$

f) **Training:** Repeat Step a)-c) except using the updated training database $\mathbf{Tr_{DBA}}$.

g) **LDA-MMI fusion:** LDA-MMI method is used to maximize the posterior probabilities of all the belief scores [17] with objective function like this [18]:

$$F_{\mathrm{MMI}}(\lambda) = \sum_{\forall i} \log \frac{p(\mathbf{x}_i|\lambda_{g(i)})P(g(i))}{\sum_{\forall j} p(\mathbf{x}_i|\lambda_j)P(j)}, \tag{14}$$

where

$$\mathbf{x} = [w_1 f_1(\varphi(x)), w_2 f_2(\varphi(x)), ..., w_N f_N(\varphi(x))], \tag{15}$$

$g(i)$ denotes its class label. w_n indicate weights of the belief of the n-th ($1 \leq n \leq N$) subsystem. Here $\sum_n w_n = 1$. Usually we define $w_n = M_n/(\sum_m M_m)$. M_n is the number of the test utterances that fit the criterion in the n-th subsystem. $P(j)$ is the prior probability of class j. $p(\mathbf{x}|\lambda)$ is weighted Gaussian mixtures.

The architecture of DBA language recognition system is shown in Fig. 2. Such a language recognition system has three advantages. First, the mentioned criterion can select high confidence test data effectively. Second, the selected criterion is simple and easy to implement. Third, the DBA iteration share the same pre-processing, feature extracting

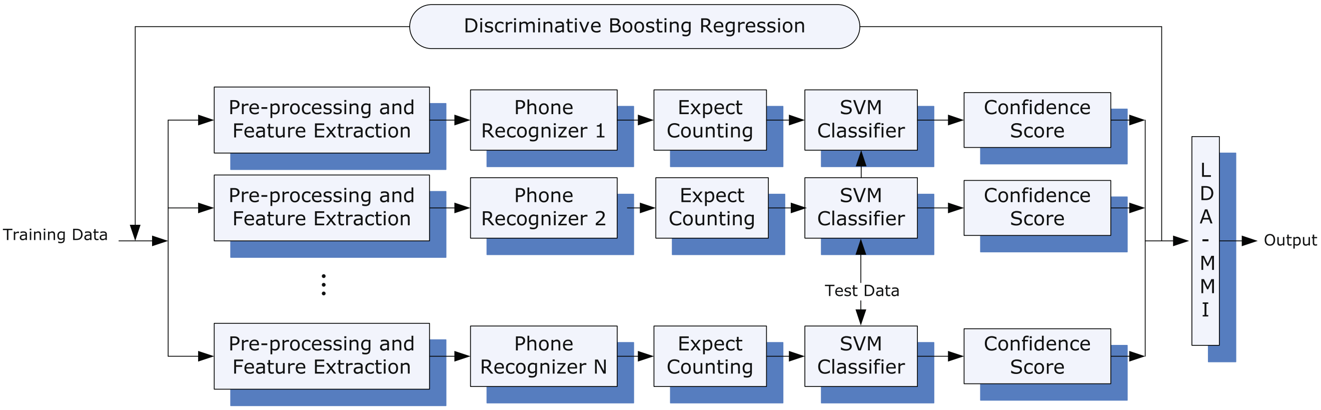


Figure 2 Architecture of DBA language system.

and decoding work with baseline system, so it introduce low additional computation cost.

4 Experimental Setup

4.1 Baseline Language Recognition System

In this paper a PPRVSM language recognition system is used as baseline system. The first step is to tokenize speech by the means of running phone-recognizers and provides the posterior probabilities of the phone occurrences. Several phone recognizers are used in parallel to decode the speech into phone lattices for analysis. In this paper we use:

(a) Hungarian (HU), Czech (CZ) and Russian (RU) Temporal Patterns (TRAPs) phone-recognizers based on ANN-HMM acoustic model that developed by the Brno University of Technology (BUT) [4]. BUT decoders for Czech (CZ), Hungarian (HU) and Russian (RU) are applied to compute phone posteriori probabilities, as used in NIST LRE tasks by many groups [19, 20]. The phone inventory is 43 for Czech, 59 for Hungarian and 50 for Russian.

(b) English (EN) phone-recognizer based on DNN-HMM acoustic model that developed by the Tsinghua University [21]. In this paper, we use the same training algorithm to train DNN-HMMs as in [15]. In the training stage of DNN-HMM acoustic model, 13-dimensional PLP features plus their first order and second order derivatives are input features to DNNs. The input PLP features are normalized to have zero mean and unit variance based on conversation-side information [22]. The GMM-HMM acoustic model contains 150 states with 32 Gaussians each. Firstly the model is trained using maximum likelihood, then the ML-trained model is used to generate state-aligned transcriptions for the succeeding DNN training. A triphone language model is trained using the transcription of the 100h Switchboard English corpus [23]. We get the phone inventories of size 47 for English phoneme recognizer, including non-phonetic units as intermittent noise and non-speech speaker noise (mapped to unknown phoneme), short pause and silence. We set the initial learning rate to 0.2 at the fine-tuning stage. At the end of every epoch, the frame accuracy of the development set is evaluate and the learning rate is reduced by a factor of 2 if the accuracy decreases. Dynamic Bayesian Network (DBN) pre-training is first applied following the process in [24] for the sigmoidal network. The implementations of the DNN are based on an extended version of CUDAMat library [25].

(c) English (EN) and Mandarine (MA) phone-recognizer based on GMM-HMM acoustic model that developed by the Tsinghua University [26]. The Mandarin phone recognizer employed in our experiments are developed using the GMM-HMM architecture and trained on about 30 hours of conversational telephone data. There are 64 phone models for the phone recognizer, each of which is a tied-state left-to-right context-dependent GMM-HMM with 32 Gaussians per state. For acoustic feature extraction, 12 PLP coefficients are extracted every 10 ms over a 25 ms hamming window. These features are augmented by their first and second order deltas, resulting in a 39 dimension feature vector

Table 1 $\mathrm{Tr}_{\mathrm{DBA}}$ of varied threshold V, DBA-M1.

	$V=6$	$V=5$	$V=4$	$V=3$	$V=2$	$V=1$
number	4939	8364	11845	15894	22707	35262
error rate	4.74 %	7.61 %	11.12 %	17.23 %	23.94 %	31.88 %

Table 2 Performance of DBA, NIST LRE 2009, DBA-M1, closed-set (EER and Cavg in %. The optimal values are shown in bold face).

Front-end		Duration		Baseline	$V = 6$	$V = 5$	$V = 4$	$V = 3$	$V = 2$	$V = 1$
	HU	$30s$	EER	2.43	2.86	2.38	2.12	**1.93**	1.96	2.52
			Cavg	2.37	2.76	2.27	2.05	**1.84**	1.90	2.45
		$10s$	EER	7.38	6.87	5.50	4.84	**4.41**	5.37	6.41
			Cavg	7.24	6.76	5.42	4.73	**4.46**	5.23	6.29
		$3s$	EER	23.00	18.86	16.60	15.65	**15.19**	16.86	23.01
			Cavg	22.61	18.54	16.50	15.40	**14.94**	16.78	22.93
	RU	$30s$	EER	2.21	2.53	2.10	1.82	**1.67**	1.82	2.30
ANN			Cavg	2.00	2.40	1.97	1.72	**1.56**	1.65	2.23
-HMM		$10s$	EER	6.23	5.60	4.66	4.11	**3.91**	4.74	5.81
			Cavg	6.07	5.52	4.56	3.99	**3.86**	4.67	5.72
		$3s$	EER	20.53	16.42	14.66	14.16	**13.69**	15.44	21.02
			Cavg	20.38	16.47	14.65	13.89	**13.27**	15.10	20.76
	CZ	$30s$	EER	3.35	3.59	2.85	2.59	**2.30**	2.41	2.69
			Cavg	3.30	3.44	2.78	2.54	**2.34**	2.44	2.61
		$10s$	EER	10.03	8.56	6.75	6.12	**5.31**	6.35	7.51
			Cavg	10.07	8.72	6.93	6.18	**5.42**	6.34	7.60
		$3s$	EER	25.20	22.36	19.94	18.23	**17.95**	19.54	25.05
			Cavg	25.14	22.14	19.76	18.05	**17.90**	19.31	25.18
	EN	$30s$	EER	2.07	2.48	2.03	1.77	**1.58**	1.79	2.42
DNN			Cavg	1.93	2.44	1.86	1.62	**1.49**	1.61	2.34
-HMM		$10s$	EER	6.65	5.44	4.54	3.97	**3.86**	4.24	5.66
			Cavg	6.71	5.53	4.67	4.11	**4.01**	4.38	5.82
		$3s$	EER	19.58	16.35	14.24	13.88	**13.53**	15.07	19.68
			Cavg	19.70	16.44	14.33	13.95	**13.63**	15.19	19.81
	MA	$30s$	EER	2.44	3.35	2.70	2.26	**2.05**	2.09	2.62
			Cavg	2.44	3.39	2.63	2.06	**1.88**	2.16	2.49
		$10s$	EER	7.51	7.37	5.52	4.67	**4.11**	5.16	6.34
			Cavg	8.23	7.41	5.60	4.69	**4.19**	5.60	6.19
		$3s$	EER	20.46	16.75	14.47	12.44	**11.72**	15.63	21.41
GMM			Cavg	20.70	17.04	14.52	12.52	**11.62**	14.50	21.02
-HMM	EN	$30s$	EER	2.29	3.05	2.48	2.15	**1.94**	2.07	2.54
			Cavg	2.30	2.91	2.30	2.01	**1.84**	1.97	2.45
		$10s$	EER	7.39	6.54	5.21	4.57	**4.04**	5.11	6.14
			Cavg	8.05	6.62	5.24	4.54	**4.13**	4.92	5.98
		$3s$	EER	20.75	16.47	14.53	12.82	**12.08**	15.40	20.83
			Cavg	20.52	16.47	14.60	12.86	**12.04**	14.69	20.40

(including c_0). To remove channel variability, cepstral mean subtraction and variance normalization are both applied. The English phone recognizer employed in our experiments are trained on 100h Switchboard English corpus. There are 47 phone models for the phone recognizer. The parameters and configuration for English phone recognizer are similar to the Mandarin phone recognizer.

Then, the decoder named HVite that is produced by HTK [27] is used to produce phone lattices, and a choice of open software (SRILM [28] and RNNLM [29]) is used to produce feature supervector. Then, a popular classifier LIBLINEAR [30] is used to classify. Finally, we use LDA-MMI algorithm [31] for score calibration.

4.2 Training, Test and Development Dataset

The results in the paper are reported for the test trials of the 2009 National Institute of Standards and Technology Language Recognition Evaluation (NIST-LRE2009). The test data is comprised by 41,793 test segments of 23 languages for 30-s, 10-s, and 3-s nominal duration test.

Table 3 Performance of DBA, NIST LRE 2009, DBA-M2, closed-set (EER and Cavg in %. The optimal values are shown in bold face).

Front-end		Duration		Baseline	$V = 6$	$V = 5$	$V = 4$	$V = 3$	$V = 2$	$V = 1$
		30s	EER	2.43	2.34	2.22	2.16	**1.96**	1.85	2.31
			Cavg	2.37	2.28	2.14	2.01	**1.89**	1.81	2.24
	HU	10s	EER	7.38	7.06	6.52	5.72	**5.27**	5.55	6.19
			Cavg	7.24	7.01	6.41	5.71	**5.28**	5.49	6.10
		3s	EER	23.00	2.96	20.22	19.02	**18.01**	18.44	22.91
			Cavg	22.61	20.95	20.20	19.00	**18.20**	18.52	22.90
		30s	EER	2.21	2.05	1.91	1.82	**1.63**	1.68	2.19
			Cavg	2.00	1.92	1.80	1.68	**1.48**	1.54	2.03
ANN	RU	10s	EER	6.23	5.70	5.04	4.52	**4.19**	4.47	5.49
-HMM			Cavg	6.07	5.60	5.05	4.49	**4.19**	4.44	5.38
		3s	EER	20.53	18.17	17.29	16.79	**15.76**	16.41	20.17
			Cavg	20.38	18.12	17.21	16.49	**15.76**	16.26	20.51
		30s	EER	3.35	3.36	3.11	2.85	**2.37**	2.49	2.59
			Cavg	3.30	3.33	3.02	2.80	**2.29**	2.49	2.53
	CZ	10s	EER	10.03	9.80	8.89	8.36	**7.37**	7.42	8.67
			Cavg	10.07	9.84	9.11	8.36	**7.47**	7.51	8.96
		3s	EER	25.20	26.28	24.22	23.08	**21.79**	22.44	25.56
			Cavg	25.14	26.37	24.37	23.11	**21.57**	22.35	25.64
		30s	EER	2.07	1.99	1.87	1.76	**1.57**	1.61	1.96
			Cavg	1.93	1.83	1.79	1.52	**1.39**	1.54	1.81
DNN	EN	10s	EER	6.65	5.43	4.93	4.37	**4.04**	4.31	5.36
-HMM			Cavg	6.71	5.53	5.01	4.43	**4.11**	4.41	5.48
		3s	EER	19.58	17.74	17.03	16.58	**15.33**	16.05	19.58
			Cavg	19.70	17.87	17.14	16.73	**15.56**	16.19	19.74
		30s	EER	2.44	2.31	2.10	2.05	**1.94**	1.97	2.43
			Cavg	2.44	2.32	2.12	1.95	**1.83**	1.88	2.31
	MA	10s	EER	7.51	6.45	5.68	5.11	**4.63**	5.16	5.84
			Cavg	8.23	6.52	5.78	5.09	**4.69**	5.04	5.77
		3s	EER	20.46	18.50	17.53	16.02	**15.12**	16.10	20.04
GMM			Cavg	20.70	18.65	17.54	16.16	**15.11**	16.04	19.87
-HMM		30s	EER	2.29	2.16	2.08	2.01	**1.87**	1.90	2.41
			Cavg	2.30	2.03	1.93	1.85	**1.76**	1.79	2.31
	EN	10s	EER	7.39	6.24	5.45	4.90	**4.45**	4.95	5.65
			Cavg	8.05	6.24	5.45	4.82	**4.42**	4.90	5.52
		3s	EER	20.75	18.95	17.92	16.48	**15.44**	15.75	19.96
			Cavg	20.52	18.97	17.79	16.58	**15.38**	15.96	19.84

Notice that in NIST LRE it is not allowed to exploit the test data information, but NIST LRE corpus is famous and widely used evaluation corpus in language recognition area, which is convenient to compare the performance of LR system. So here we use the NIST data to confirm the effectiveness of discriminative boosting algorithm, which also works in another corpus.

180,000 conversations selected from the Call-Home, Call-Friend, OGI, OHSU and VOA Corpus are used in this paper for training.

22,701 conversations are selected from the database provided by NIST for the 2003, 2005 and 2007 LRE and VOA as development database.

4.3 Evaluation Measures

In this paper, the performance of language recognition systems is reported in terms of equal error rate (EER) and average cost performance Cavg which is defined by NIST LRE 2009 [12].

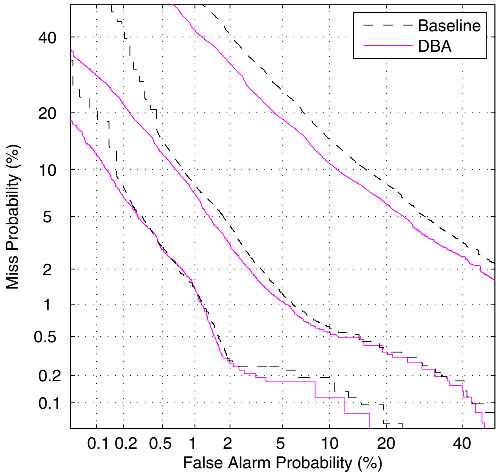


Figure 3 DET curves of baseline and (DBA-M1)+(DBA-M2), (V = 3) system, NIST LRE 2009, ANN-HMM (HU+RU+CZ)+ DNN-HMM (EN) + GMM-HMM (EN+MA) frontend.

5 Experimental Results and Discussion

5.1 Component Analysis of $\mathbf{Tr}_{\mathbf{DBA}}$

Table 1 shows the component of $\mathbf{Tr}_{\mathbf{DBA}}$ of DBA-M1 in different threshold. The error rate of the $\mathbf{Tr}_{\mathbf{DBA}}$ database will decrease with the increasing of the threshold, while the number of the utterances also decreases. The error rate of $\mathbf{Tr}_{\mathbf{DBA}}$ using DBA-M2 will decrease because the low error rate of the original training database **Tr**.

5.2 Performance of DBA System

We investigate the performance of DBA in this subsection. We vary the threshold V and the EER and Cavg results of DBA-M1 and DBA-M2 are listed in Tables 2 and 3, respectively. From the results of Table 2, we can see that for each fixed frontend, the EERs/Cavgs first decrease and then increase with the decreasing of V, and the minimum occurs at $V = 3$. Although when $V = 6$ the error rate of $\mathbf{Tr}_{\mathbf{DBA}}$ is lower than that of $V = 3$, but the smaller number of training samples makes higher EER. In Table 3 the trend with respect to V is similar to Table 2 but the minimum also occurs at $V = 3$ because of the tradeoff of the number of training data and error rate. When V=4,5,6 the error rate of the $\mathbf{Tr}_{\mathbf{DBA}}$ is very low, while the number of utterances in $\mathbf{Tr}_{\mathbf{DBA}}$ is small as the same time. The small number of training data makes the deduction of performance of the system. When V=1,2 the number of utterances in $\mathbf{Tr}_{\mathbf{DBA}}$ is enough for training but the error rate of the $\mathbf{Tr}_{\mathbf{DBA}}$ becomes higher, which also leads to a rise of EER. But when V=3 it occurs a balance of the error rate and the number of training data. So when V=3 the language recognition system achieves the best performance. In the subsection, we have seen that DBA-M2 outperformed DBA-M1 at 30s test because of the plenty training utterances, but DBA-M1 outperformed DBA-M2 at 10s and 3s test because only use test data achieve higher robustness in speakers, background noise, channel conditions.

5.3 Parallel Phone Recognizer Experiments

In the previous subsections, we have seen that DBA-M1 and DBA-M2 outperformed PPRVSM method. In this section, we will further fuse subsystems that uses parallel HU, RU, CZ, MA and EN frontends through LDA + MMI score fusion backend [31]. We only focus on the most challenging case, i.e., PPRVSM versus (DBA-M1)+(DBA-M2) (V = 3). The detection error trade-off (DET) curves are showed

Table 4 Performance of PPRVSM and DBA systems, NIST LRE 2009, closed set, (DBA-M1)+(DBA-M2), $V = 3$ (EER/Cavg in %. The fusion results are shown in bold face).

	System		30s	10s	3s
		HU	2.43/2.37	7.38/7.24	23.00/22.61
	ANN-HMM	RU	2.21/2.00	6.23/6.07	20.53/20.38
		CZ	3.35/3.30	10.03/10.07	25.20/25.14
Baseline	GMM-HMM	EN	2.29/2.30	7.39/8.05	20.75/20.52
		MA	2.44/2.44	7.51/8.23	20.46/20.70
2-6	DNN-HMM	EN	2.07/1.93	6.65/6.71	19.58/19.70
	fusion		**1.11/1.16**	**2.73/3.70**	**12.37/12.76**
		HU	1.89/1.86	4.39/4.41	14.82/14.80
	ANN-HMM	RU	1.60/1.48	3.82/3.83	13.41/13.02
		CZ	2.34/2.27	5.14/5.27	18.16/17.47
DBA	GMM-HMM	EN	1.93/1.81	4.11/4.22	12.38/12.03
		MA	2.06/1.91	4.19/4.32	11.77/11.68
	DNN-HMM	EN	1.53/1.41	3.51/3.56	11.38/11.03
	fusion		**1.09/0.98**	**2.41/2.44**	**10.47/10.68**

in Fig. 3, and the EERs and Cavgs are listed in Table 4. Table 4 shows the consistent performance improvement due to changing from PPRVSM to DBA for both single and parallel frontends. For the parallel frontends and 30s, 10s and 3s test, the EER decreases from 1.11 %, 2.73 % and 12.37 % to 1.09 %, 2.41 %, and 10.47 %.

5.4 Computational Cost

Let F, M_{training} and M_{test} denote the dimension of the phonotactic feature supervector of an utterance, the number of utterances of training dataset and the number of utterances of test dataset, respectively. And let $C'_\varphi(f, N)$ denote the computation cost of the mapping from x to $\varphi(x)$ (here $F = f_{\text{n}}^{N}$), $C'_{\text{modeling}}(F, M)$ denote the computational cost of modeling the languages, which relate to F and M. Then the computational cost of the baseline system is

$$\begin{aligned} C'_{\text{baseline}} = {} & (M_{\text{training}} + M_{\text{test}}) \cdot C'_\varphi \\ & + C'_{\text{modeling}}(F, M_{\text{training}}) + M_{\text{test}} C'_{\text{test}} \end{aligned} \tag{16}$$

where C'_{test} denotes the computational cost of test, and

$$\begin{aligned} C'_\varphi = {} & C'_{\text{Pre-Processing}} + C'_{\text{FeatureExtract}} \\ & + C'_{\text{Decoding}} + C'_{\text{ExpectCounting}} \end{aligned} \tag{17}$$

where $C'_{\text{Pre-Processing}}$, $C'_{\text{FeatureExtract}}$, C'_{Decoding} and $C'_{\text{ExpectCounting}}$ denote the computational cost of preprocessing, feature extracting, decoding and expect counting, respectively.

Let $M'_{|V=n}$ denotes the number of test utterances that have more than n votes, then the computational cost of the DBA system is computed as:

$$\begin{aligned} C'_{\text{DBA}} = {} & (M_{\text{training}} + M_{\text{test}}) \cdot C'_\varphi + C'_{\text{modeling}}(F, M_{\text{training}}) \\ & + C'_{\text{modeling}}(F, (M_{\text{training}} + M'_{|V=n})) + 2 M_{\text{test}} C'_{\text{test}}, \end{aligned}$$

so

$$\frac{C'_{\text{DBA}}}{C'_{\text{baseline}}} = 1 + \frac{C'_{\text{modeling}}(F, (M_{\text{training}} + M'_{|V=n})) + M_{\text{test}} C'_{\text{test}}}{C'_{\text{baseline}}} \tag{18}$$

Usually in PPRVSM, decoding and super vector product is the dominant part, so $C'_\varphi >> C'_{\text{modeling}}(F, (M_{\text{training}} + M'_{|V=n})) > C'_{\text{modeling}}(F, M_{\text{training}})$ and $C'_\varphi >> M_{\text{test}} C'_{\text{test}}$. so

$$\frac{C'_{\text{DBA}}}{C'_{\text{baseline}}} \approx 1 \tag{19}$$

That means the DBA system almost takes no extra computation and achieves a 1.8 %, 11.72 % and 15.35 % relative improvements respectively for 30s, 10s and 3s compared to PPRVSM.

Table 5 Comparison of real time factor for PPRVSM and DBA, HU frontend, NIST LRE 2009, 30-s test. CPU: Xeon E5520@2.27GHz, RAM: 8GB, single thread. SV gen.: super vector generation, SV prod.: super vector product.

System	Decoding	SV gen.	SV prod.
PPRVSM	0.11	1.1×10^{-4}	3.7×10^{-6}
DBA	0.11	3.1×10^{-4}	8.3×10^{-6}

5.5 Real Time Factors

Next, we count the real time (RT) factors of each part and list the results in Table 5. For the training stage, decoding and super vector product is the dominant part, the computational cost for DBA system is same to baseline. For the test stage, decoding and super vector generation are the dominant parts and the computational cost almost does not increase for the DBA compared to PPRVSM.

6 Conclusions

In this paper, an approach of discriminative boosting algorithm has been presented for language recognition. DBA is an discriminative method using simple but effective criterion based on boosting, which uses the complementary information of different frontend and the discriminative information of the test data. The performance improvements demonstrate that DBA can learn the discriminative information of the test data effectively. The experimental results evaluated on NIST 2009 LRE task show that the relative improvements of the proposed DBA are 1.8 %, 11.72 % and 15.35 % for 30s, 10s and 3s over traditional PPRVSM approach respectively.

References

1. Li, H., Ma, B., & Lee, K.A. (2013). Spoken language recognition: from fundamentals to practice. *Proceedings of the IEEE*, *101*(5), 1136–1159.
2. Zissman, M.A. (1996). Comparison of four approaches to automatic language identification of telephone speech. *IEEE Transactions on Speech and Audio Processing*, *4*(1), 31–34.
3. Torres-Carrasquillo, P.A., Singer, E., Kohler, M.A., Greene, R.J., Reynolds, D.A., & Deller Jr, J.R. (2002). Approaches to language identification using Gaussian mixture models and shifted delta cepstral features. In *INTERSPEECH* (pp. 33–36).
4. Schwarz, P. (2009). Phoneme recognition based on long temporal context. PhD thesis. Brno University of Technology.
5. Sim, K.C., & Li, H. (2008). On acoustic diversification front-end for spoken language identification. *IEEE Transactions on Audio, Speech, and Language Processing*, *16*(5), 1029–1037.

6. Wells, W.M., Viola, P., Atsumi, H., Nakajima, S., & Kikinis, R. (1996). Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, *1*(1), 35–51.
7. Bahl, L., Brown, P., de Souza, P.V., & Mercer, R. (1986). Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. 49–52).
8. Povey, D., & Woodland, P.C. (2002). Minimum phone error and I-smoothing for improved discriminative training. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (pp. 101–105).
9. Juang, B.H., & Katagiri, S. (1992). Discriminative learning for minimum error classification [pattern recognition]. *IEEE Transactions on Signal Processing*, *40*(12), 3043–3054.
10. Zhang, W.Q., He, L., Deng, Y., Liu, J., & Johnson, M.T. (2011). Time-Frequency Cepstral Features and Heteroscedastic Linear Discriminant Analysis for Language Recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, *19*(2), 266–276.
11. Singer, E., Torres-Carrasquillo, P.A., Gleason, T.P., Campbell, W.M., & Reynolds, D.A. (2003). Acoustic, phonetic, and discriminative approaches to automatic language identification. In *INTERSPEECH* (pp. 1944–1948).
12. Martin, A.F., & Greenberg, C.S. (2010). The 2009 NIST Language Recognition Evaluation. In *Odyssey* (p. 30).
13. Li, H., Ma, B., & Lee, C.H. (2007). A vector space modeling approach to spoken language identification. *IEEE Transactions on Audio, Speech, and Language Processing*, *15*(1), 271–284.
14. Gauvain, J.L., Messaoudi, A., & Schwenk, H. (2004). Language recognition using phone latices. In *INTERSPEECH* (pp. 1283–1286).
15. Dahl, G.E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(1), 30–42.
16. Campbell, W.M., Campbell, J.P., Reynolds, D.A., Jones, D.A., & Leek, T.R. (2003). Phonetic speaker recognition with support vector machines. In *Advances in Neural Information Processing Systems* (pp. 1377–1384).
17. Matejka, P., Burget, L., Glembek, O., Schwarz, P., Hubeika, V., Fapso, M., & Plchot, O. (2007). BUT system description for NIST LRE 2007. In *2007 NIST Language Recognition Evaluation Workshop* (pp. 1–5).
18. Povey, D. (2005). Discriminative training for large vocabulary speech recognition (Doctoral dissertation, University of Cambridge).
19. Jancik, Z., Plchot, O., Brmmer, N., Burget, L., Glembek, O., Hubeika, V., & Cernocky, J. (2010). Data selection and calibration issues in automatic language recognition-investigation with BUT-AGNITIO NIST LRE 2009 system. In *Odyssey* (pp. 215–221).
20. Torres-Carrasquillo, P.A., Singer, E., Gleason, T., McCree, A., Reynolds, D.A., Richardson, F., & Sturim, D. (2010). The MITLL NIST LRE 2009 language recognition system. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* (pp. 4994–4997).
21. Liu, W.-W., Cai, M., Yuan, H., Xu, J., Liu, J., & Zhang, W.-Q. (2014). DNN-HMM acoustic model for phonotactic language recognition. *International Symposium on Chinese Spoken Language Processing(ISCSLP)*, 148–152.
22. Cai, M., Shi, Y., & Liu, J. (2013). Deep maxout neural networks for speech recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)* (pp. 291–296).
23. Godfrey, J.J., Holliman, E.C., & McDaniel, J. (1992). SWITCHBOARD: Telephone speech corpus for research and development. In *EEE International Conference on Acoustics Speech, and Signal Processing* (pp. 517–520).
24. Seide, F., Li, G., Chen, X., & Yu, D. (2011). Feature engineering in context-dependent deep neural networks for conversational speech transcription. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)* (pp. 24–29).
25. Mnih, V. (2009). Cudamat: a CUDA-based matrix class for python. Department of Computer Science, University of Toronto, Tech. Rep. UTML TR.
26. Deng, Y., Zhang, W.-Q., Qian, Y.M., & Liu, J. (2011). Language recognition based on acoustic diversified phone recognizers and phonotactic feature fusion. *IEICE Transactions on Information and Systems*, *94*(3), 679–689.
27. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X.A., & Woodland, P. (2006). The HTK book (for HTK version 3.4).
28. Stolcke, A. (2002). SRILM-an extensible language modeling toolkit. In *INTERSPEECH*.
29. Mikolov, T., Kombrink, S., Deoras, A., Burget, L., & Cernocky, J. (2011). RNNLM-Recurrent neural network language modeling toolkit. In *Proceeding of the 2011 ASRU Workshop* (pp. 196–201).
30. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., & Lin, C.J. (2008). LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 1871–1874.
31. Zhang, W.Q., Hou, T., & Liu, J. (2010). Discriminative score fusion for language identification. *Chinese Journal of Electronics*, *19*(1), 124–128.

Wei-Wei Liu received the B.S. degree in communication engineering from Xidian University, Xi'an, China, in 2003, the M.S. degree in communication engineering from National University of Defense Technology, Changsha, China, in 2006. She is currently pursuing the Ph.D. degree in the Department of Electronic Engineering, Tsinghua University, Beijing, China. Her research focuses upon speech processing and machine learning.

Meng Cai received the B.S. degree in information engineering from Beijing Institute of Technology in 2010. He is currently Ph.D. candidate in the Department of Electronic Engineering, Tsinghua University. His research interests cover acoustic modeling, machine learning and speech recognition.

Wei-Qiang Zhang received the B.S. degree in applied physics from China University of Petroleum in 2002, the M.S. degree in communication and information systems from Beijing Institute of Technology in 2005, and the Ph.D. degree in information and communication engineering from Tsinghua University in 2009.

He is an Associate Professor in the Department of Electronic Engineering, Tsinghua University. His research interests are in the area of speech and signal processing, machine learning and statistical pattern recognition.

Jia Liu received his B.S., M.S. and Ph.D. degrees in communication and electronic systems from Tsinghua University, Beijing, China, in 1983, 1986 and 1990, respectively. He worked at the Remote Sensing Satellite Ground Station, Chinese Academy of Sciences, after his Ph.D., and worked as a Royal Society visiting scientist at Cambridge University Engineering Department during 1992-1994. He is now a professor in the Department of Electronic Engineering, Tsinghua University. His research fields include speech recognition, speaker recognition, language recognition, expressive speech synthesis, speech coding, and spoken language understanding.

Michael T. Johnson received the B.S. degree in computer science engineering and the B.S. degree in engineering with electrical concentration from LeTourneau University, Longview, TX, in 1989 and 1990, respectively, the M.S.E.E. degree from the University of Texas, San Antonio, in 1994, and the Ph.D. degree from Purdue University, West Lafayette, IN, in 2000. He worked as a Design Engineer and Engineering Manager from 1990 to 1996, and is currently a Professor in the Department of Electrical and Computer Engineering, Marquette University, Milwaukee, WI. His primary research area is speech and signal processing, with interests in machine learning, statistical pattern recognition, and nonlinear signal processing.