A Syllable Lattice Approach to Speaker Verification

Minho Jin, *Student Member, IEEE*, Frank K. Soong, *Senior Member, IEEE*, and Chang D. Yoo, *Member, IEEE*

***Abstract*—This paper proposes a syllable-lattice-based speaker verification algorithm for Mandarin Chinese input. For each speech utterance, a syllable lattice is generated with a speaker-independent large-vocabulary continuous speech recognition system in free syllable decoding. The verification decision is made based upon the likelihood ratio between a target-speaker model and a speaker-independent background model, computed on the decoded syllable lattice. The likelihood function is calculated efficiently in a forward algorithm by considering all paths in the lattice. The proposed algorithm was evaluated using a Mandarin Chinese database, where 1832 true and 26 250 impostor trials were recorded by 19 target speakers and 180 impostors. The average duration of each trial is 2 s long without silence. The target-speaker model was adapted from the speaker-independent background model using enrollment data of two minutes with silence. The proposed algorithm achieved an equal-error rate of 0.857% which is beter than 1.21% of the hidden Markov model-based speaker verification algorithm without using syllable lattices. The equal-error rate was further reduced to 0.617% by incorporating the Goussian mixture model–universal background model algorithm with 2048 Gaussian kernels whose equal error rate is 0.990%.**

***Index Terms*—Lattice-based speaker adaptation, lattice rescoring, Mandarin Chinese, speaker recognition.**

I. Introduction

A SPEAKER verification system accepts or rejects a speaker's claimed identity based on his or her speech. It can be operated in a text-dependent (TD) mode or a text-independent (TI) mode. In a TD mode, the system takes the input speech along with its prespecified text transcription, whereas in a TI mode, only the input speech is used. When the true underlying transcription is unavailable, as in the case of free conversation over the telephone, the system must operate in a TI mode. This paper will focus on TI mode applications.

Various algorithms have been proposed for TD and TI speaker verification applications. Rosenberg *et al.* [1] proposed a hidden Markov model (HMM)-based algorithm for both TD and TI mode operations. An accept/reject decision is made based on the likelihood-ratio test using HMMs which represent subword units. Matsui and Furui [2] proposed a phoneme HMM-based algorithm, where a test speaker is requested to utter a randomly prompted phrase.

Manuscript received October 26, 2006; revised June 6, 2007. This work was supported in part by MIC and IITA through IT Leading R&D Support Project in Korea. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mary P. Harper.

M. Jin and C. D. Yoo are with the Division of Electrical Engineering, Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: jinmho@kaist.ac.kr; cdyoo@ee.kaist.ac.kr).

F. K. Soong is with Microsoft Research Asia (MSRA), Beijing, China (e-mail: frankkps@microsoft.com).

Digital Object Identifier 10.1109/TASL.2007.906181

Reynolds *et al.* [3] proposed the Gaussian mixture model–universal background model (GMM–UBM) algorithm in the TI mode. The UBM is a Gaussian mixture model (GMM) which is estimated from background speakers. A target-speaker model is adapted from the UBM using speaker adaptation algorithms. An accept/reject decision is made based on the target/background likelihood-ratio score. The GMM–UBM algorithm needs a relatively small number of parameters compared to HMM-based algorithms. Thus, when only a small amount of enrollment data is available, the GMM–UBM is usually preferable to HMM-based algorithms.

Weber *et al.* [4] proposed a large-vocabulary continuous speech recognition (LVCSR)-based algorithm for TI applications, where the best recognition result generated by an LVCSR is assumed to be the true underlying transcription of the input speech. This algorithm achieves good performance for long test speech utterances (30 s or longer). However, for an input with a short duration, say less than 10 s, its performance starts to degrade, and the result is inferior to that of the GMM–UBM algorithm [4].

Doddington [5] demonstrated that the idiolectal characteristics can be used for speaker verification. The idiolect of a speaker was represented by the word n-grams embedded in the underlying transcriptions. This algorithm achieves good performance in the Switchboard corpus [6], which implies that not only acoustic but also idiolectal information is useful for speaker verification

Andrews *et al.* [7] proposed a phonetic information-based speaker verification algorithm. In their work, the verification is performed from the phonetic sequences created by six-phone recognizers which were trained on six different languages. Recently, Hatch *et al.* [8] demonstrated that using phone lattice decodings instead of 1-best phone decodings can improve the performance of the phonetic speaker verification further.

This paper proposes an HMM-based speaker verification algorithm that rescores a syllable lattice using a target-speaker model. The performance of an HMM-based speaker verification algorithm is known generally to be worse than that of the GMM–UBM algorithm in the TI mode. The two main underlying reasons for this disparity in performance are errorful recognition results and insufficient enrollment data: errorful recognition results not only lead to incorrect decision making but also lead to incorrect target-speaker modeling. Our previous work [9] showed that the HMM-based speaker verification performance can be improved by using syllable lattices generated by the target-speaker model and the background model. In this paper, an algorithm is proposed to 1) make verification decision by rescoring syllable lattices using the target-speaker model and 2) enroll the target speaker model using a lattice-based speaker adaptation.

1558-7916/$25.00 © 2007 IEEE

There was one other speaker verification algorithm that is based on the lattice decoding [8]. The proposed algorithm is different from the lattice-based algorithm proposed by Hatch *et al.* [8] in that ours uses the lattice decoding to capture the difference between speakers in acoustical information (such as Mel-frequency cepstral coefficients) represented by HMMs while the other uses the lattice decoding to capture the difference in the dynamic realization of phonetic features represented by the relative frequencies of phone n-grams.

The proposed algorithm uses syllable lattices instead of phone lattices to exploit the specific syllabic characteristics of Mandarin Chinese. Mandarin Chinese is a syllabically paced language which consists of about 1300 syllables [10]. Since the combination of several specific syllables in general stands for very few words, accurate syllable decodings often lead to accurate word-recognition results [11]. For this reason, this paper uses syllable-lattice decodings for both speech recognition and speaker verification.

Using the decoded lattice instead of the 1-best recognition result increases the probability that the correct hypothesis and its alignment is incorporated in the decision-making process. In the proposed algorithm, the likelihood given a model is approximated as the joint probability of an input observation sequence and a lattice given the model using the forward algorithm. The accept/reject decision is based on the target-speaker's posterior probability which is approximated by the likelihoods of the target-speaker model and the background model. The target-speaker model was adapted from the background model using a lattice-based speaker adaptation algorithm which is known to be robust against the 1-best word error rate (WER) [12], [13].

The rest of this paper is organized as follows. Section II describes the proposed algorithm. Section III demonstrates the performance of the proposed algorithm, and Section IV summarizes our work.

II. Lattice-Based Speaker Verification Algorithm

A. Speaker Verification

Fig. 1 illustrates a generic speaker verification system that uses a background and a target-speaker model. In this paper, we use a single speaker-independent (SI) model as a background model for all target speakers. The target-speaker model is obtained by adapting the background model with a limited amount of enrollment data of the target speaker. The Neyman–Pearson criterion leads to the following hypothesis-testing procedure:

$$\frac{p\left(\mathbf{O}_1^T|\text{TAR}\right)}{p\left(\mathbf{O}_1^T|\text{BGM}\right)} \underset{\text{false}}{\overset{\text{true}}{\gtrless}} \tau_\alpha \tag{1}$$

where $\mathbf{O}_1^T$, τ_α, $p\left(\mathbf{O}_1^T|\text{TAR}\right)$, and $p\left(\mathbf{O}_1^T|\text{BGM}\right)$ are the input observation sequence of T frames, a preset decision threshold, the likelihood given the target-speaker model, and the likelihood given the background model, respectively.

B. Lattice-Based Speaker Verification

Multiple hypotheses encoded in a lattice carry more information than the single best hypothesis, and they can potentially enhance the speaker verification performance. HMM-based

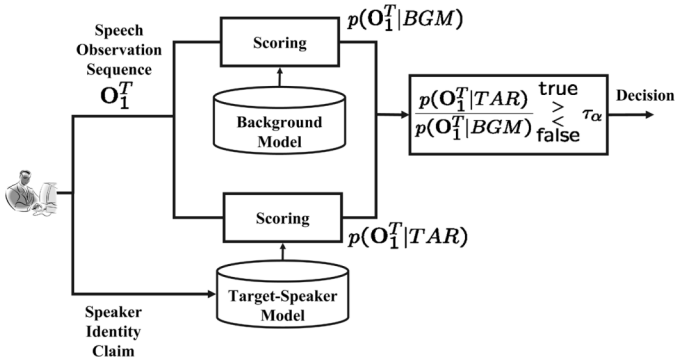

Fig. 1. Generic speaker verification system.

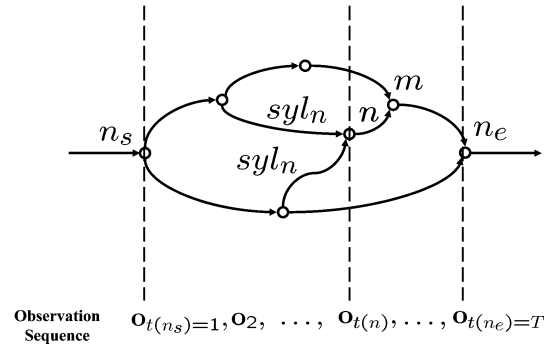

Fig. 2. Lattice forward algorithm.

speaker verification algorithms using an LVCSR system approximate the likelihoods of the background model and the target-speaker model by [4], [14]

$$p\left(\mathbf{O}_1^T|\text{BGM}\right)_{1-\text{best}} \approx p\left(\mathbf{O}_1^T, Q^*|\lambda_{\text{SI}}\right) \tag{2}$$

and

$$p\left(\mathbf{O}_1^T|\text{TAR}\right)_{1-\text{best}} \approx p\left(\mathbf{O}_1^T, Q^*|\lambda_{\text{TAR}}\right) \tag{3}$$

where Q^*, λ_{SI}, and λ_{TAR} denote the optimal state alignment of the best recognition result generated by a speech recognizer, the SI-HMMs and the target-speaker HMMs, respectively. Since (2) and (3) depend only on the best speech recognition result, the speaker verification performance may deteriorate when the recognition result is highly errorful. Rather than using only the best recognition result, we utilize the decoded lattice to incorporate more hypotheses in verification decision. In the phonetic speaker verification and the language identification, using decoded phone-lattices improves the performances [8], [15]. This paper uses decoded syllable lattices for HMM-based speaker verification.

1) Forward Algorithm for Lattice: To calculate lattice likelihoods, we generate syllable lattices using a free syllable decoding network. Fig. 2 illustrates the structure of a syllable lattice: n_s and n_e are the starting and the ending nodes of the lattice, respectively. Each arc in the lattice is associated with a syllable hypothesis whose label is determined by its fan-in node. For example, all arcs entering the same fan-in node n represent the same syllable syl_n.

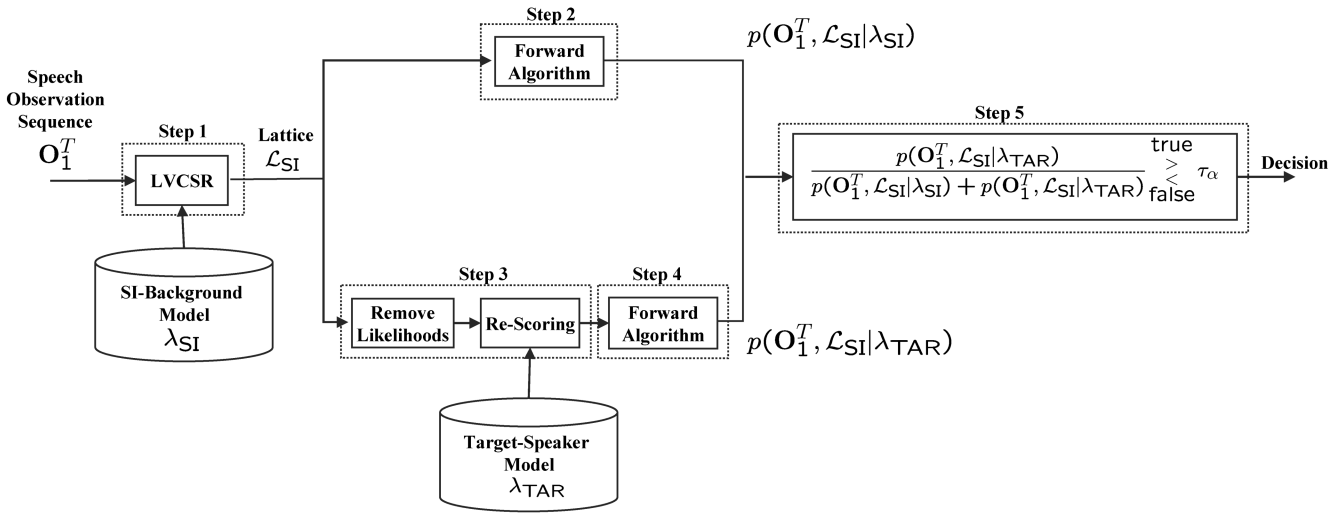

Fig. 3. Plug-in lattice rescoring for lattice-based speaker verification.

Let α_n be the forward probability of a node n defined by

$$\alpha_n = p(\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_{t(n)}, n, \mathcal{L}|\lambda) \tag{4}$$

where $\mathbf{o}_i$, $t(n)$, $\mathcal{L}$, and λ denote the observation at frame index i, the frame index corresponding to node n, the lattice generated by a speech recognizer, and the set of HMMs, respectively. Then, at node m, α_m is induced as

$$\alpha_m = \sum_{\{l|l\to m\}} \alpha_l p(m|l) \tag{5}$$

where $l \to m$ is the transition from node l to node m, and $p(m|l)$ is the corresponding transition probability. Using (5), the likelihood of $\mathbf{O}_1^T$ given λ can be approximated as the joint probability of $\mathbf{O}_1^T$ and $\mathcal{L}$ as given by

$$\begin{aligned} p\left(\mathbf{O}_1^T|\lambda\right) &\approx p(\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_{t(n_e)=T}, \mathcal{L}|\lambda) \\ &= \alpha_{n_e} \end{aligned} \tag{6}$$

where n_e denotes the ending node of lattice $\mathcal{L}$.

2) Plug-in Lattice Rescoring: The proposed algorithm approximates the likelihoods given the background model and the target-speaker model by

$$p\left(\mathbf{O}_1^T|\text{BGM}\right) \approx p\left(\mathbf{O}_1^T, \mathcal{L}_{\text{SI}}|\lambda_{\text{SI}}\right) \tag{7}$$

and

$$p\left(\mathbf{O}_1^T|\text{TAR}\right) \approx p\left(\mathbf{O}_1^T, \mathcal{L}_{\text{SI}}|\lambda_{\text{TAR}}\right) \tag{8}$$

where $\mathcal{L}_{\text{SI}}$ denotes the lattice generated by the speech recognizer using the SI background model λ_{SI}.

Since the lattice $\mathcal{L}_{\text{SI}}$ is generated with λ_{SI}, the likelihood $p\left(\mathbf{O}_1^T, \mathcal{L}_{\text{SI}}|\lambda_{\text{SI}}\right)$ can be directly obtained by the forward algorithm. In order to calculate $p\left(\mathbf{O}_1^T, \mathcal{L}_{\text{SI}}|\lambda_{\text{TAR}}\right)$, plug-in lattice rescoring can be adopted. For an arc κ in the lattice $\mathcal{L}_{\text{SI}}$, we denote κ_s and κ_e as its starting and ending nodes, respectively. In the plug-in lattice rescoring, the joint probability of both an arc κ and its observation sequence $\mathbf{O}_{t(\kappa_s)}^{t(\kappa_e)} = [\mathbf{o}_{t(\kappa_s)} \ldots \mathbf{o}_{t(\kappa_e)}]$ given a model λ_{SI}, $p\left(\mathbf{O}_{t(\kappa_s)}^{t(\kappa_e)}, \kappa|\lambda_{\text{SI}}\right)$, is replaced by $p\left(\mathbf{O}_{t(\kappa_s)}^{t(\kappa_e)}, \kappa|\lambda_{\text{TAR}}\right)$ for speaker verification. Fig. 3 illustrates the steps in calculating $p\left(\mathbf{O}_1^T, \mathcal{L}_{\text{SI}}|\lambda_{\text{SI}}\right)$ and $p\left(\mathbf{O}_1^T, \mathcal{L}_{\text{SI}}|\lambda_{\text{TAR}}\right)$ with the plug-in lattice rescoring which is also shown as follows.

Plug-In Lattice Rescoring for Speaker Verification

Step 1) Generate a lattice $\mathcal{L}_{\text{SI}}$ using the SI background model λ_{SI}.
Step 2) Calculate $p\left(\mathbf{O}_1^T, \mathcal{L}_{\text{SI}}|\lambda_{\text{SI}}\right)$ using the forward algorithm.
Step 3) Replace $p\left(\mathbf{O}_{t(\kappa_s)}^{t(\kappa_e)}, \kappa|\lambda_{\text{SI}}\right)$ with $p\left(\mathbf{O}_{t(\kappa_s)}^{t(\kappa_e)}, \kappa|\lambda_{\text{TAR}}\right)$ for all κ's in $\mathcal{L}_{\text{SI}}$ (plug-in rescoring).
Step 4) Calculate $p(\mathbf{O}_1^T, \mathcal{L}_{\text{SI}}|\lambda_{\text{TAR}})$ using the likelihoods $p\left(\mathbf{O}_{t(\kappa_s)}^{t(\kappa_e)}, \kappa|\lambda_{\text{TAR}}\right)$ obtained in Step 3 and the forward algorithm.
Step 5) Make verification decision.

The posterior probability of the target speaker will be used as a test statistic in this paper. When a noninformative prior is used, i.e., both prior probabilities of the background and target speaker are assumed to be 0.5, the posterior probability of the target speaker is

$$\begin{aligned} p\left(\text{TAR}|\mathbf{O}_1^T\right) &= \frac{p\left(\mathbf{O}_1^T|\text{TAR}\right)}{p(\mathbf{O}_1^T|\text{BGM}) + p\left(\mathbf{O}_1^T|\text{TAR}\right)} \\ &\approx \frac{p\left(\mathbf{O}_1^T, \mathcal{L}_{\text{SI}}|\lambda_{\text{TAR}}\right)}{p\left(\mathbf{O}_1^T, \mathcal{L}_{\text{SI}}|\lambda_{\text{SI}}\right) + p\left(\mathbf{O}_1^T, \mathcal{L}_{\text{SI}}|\lambda_{\text{TAR}}\right)}. \end{aligned} \tag{9}$$

C. Lattice-Based Speaker Adaptation

The lattice-based speaker adaptation is used here to estimate the target-speaker model. Mariéthoz and Bengio [16] demonstrated that the GMM–UBM using the maximum *a posterior* (MAP) adaptation [17], [18] outperforms the GMM–UBM using the maximum-likelihood linear regression (MLLR) [19] and the eigenvoices [20]. In HMM-based speaker verification algorithms, the amount of enrollment data is relatively small to update all parameters of HMMs. And the underlying transcriptions of enrollment data may not be available. In such a situation, the MLLR outperforms the MAP in speech recognition performance. We evaluate both the MLLR and the MAP for speaker verification applications in addition to lattice-based speaker adaptation algorithms.

1) MAP and MLLR: Let s^m be the state containing the mth Gaussian kernel in SI-HMM λ_{SI}. For the observation sequence $\mathbf{O}_1^T = [\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T]$, the Gaussian occupation count at frame i is given by

$$N_m(i) = p\left(s_i = s^m, m_i = m | \lambda_{\mathrm{SI}}, \mathbf{O}_1^T\right) \tag{10}$$

where s_i and m_i denote the state and the mixture at frame i, respectively. Using the Gaussian occupation count $N_m = \sum_{i=1}^{T} N_m(i)$, the MAP adaptation estimates the speaker-adapted mean $\hat{\boldsymbol{\mu}}_m$ of the mth Gaussian kernel as follows [15], [21]:

$$\hat{\boldsymbol{\mu}}_m = \frac{N_m}{N_m + \tau}\bar{\boldsymbol{\mu}}_m + \frac{\tau}{N_m + \tau}\boldsymbol{\mu}_m \tag{11}$$

where

$$\bar{\boldsymbol{\mu}}_m = \sum_{i=1}^{T} \frac{N_m(i)}{N_m}\mathbf{o}_i \tag{12}$$

and where $\boldsymbol{\mu}_m$ and τ denote the mean of the mth Gaussian kernel and a prior knowledge chosen beforehand, respectively.

In MLLR, the Gaussian kernels are clustered into regression classes by the regression tree clustering algorithm based on a centroid splitting algorithm [19], [21]. Let $\boldsymbol{\zeta}_m$ be the extended mean of the mth Gaussian kernel defined by

$$\boldsymbol{\zeta}_m = \left[1\, \boldsymbol{\mu}_m^T\right]^T. \tag{13}$$

For all Gaussian kernels in a regression cluster r, the mean-parameter transformation $\mathbf{W}_r$ is obtained by solving

$$\sum_{i=1}^{T}\sum_{k=1}^{K_r} N_{r(k)}(i)\boldsymbol{\Sigma}_{r(k)}^{-1}\mathbf{o}_i\boldsymbol{\zeta}_{r(k)}^T = \sum_{i=1}^{T}\sum_{k=1}^{K_r} N_{r(k)}(i)\boldsymbol{\Sigma}_{r(k)}^{-1}\mathbf{W}_r\boldsymbol{\zeta}_{r(k)}\boldsymbol{\zeta}_{r(k)}^T \tag{14}$$

where K_r, $r(k)$, and $\boldsymbol{\Sigma}_{r(k)}$ are the number of Gaussian kernels in the regression class r, the index of the kth Gaussian kernel in the regression class r and the covariance matrix of the $r(k)$th Gaussian kernel, respectively. Then, the speaker-adapted mean $\hat{\boldsymbol{\mu}}_{r(k)}$ in a class r is adapted from the SI mean $\boldsymbol{\mu}_{r(k)}$ as follows:

$$\hat{\boldsymbol{\mu}}_{r(k)} = \mathbf{W}_r\boldsymbol{\zeta}_{r(k)}. \tag{15}$$

2) Lattice-Based Gaussian Occupation Count: The unsupervised adaptation based on the best recognition result is usually performed by the following steps.

Unsupervised Adaptation Based upon 1-Best Recognition Result

Step 1) Generate the best recognition result of adaptation data using a speech recognizer.
Step 2) Accumulate Gaussian occupation counts using (10).
Step 3) Update parameters with (11) for the MAP and (15) for the MLLR.

Since the above unsupervised speaker adaptation adopts the recognition result as the true transcription, the adaptation performance deteriorates when the best recognition result is highly errorful. In order to alleviate this negative impact on the speaker verification performance, the lattice-based adaptation was proposed before for the MLLR adaptation algorithm [12], [13]. In our implementation, we first generate a lattice $\mathcal{L}_{\mathrm{SI}}$ using the SI-HMM λ_{SI}. Then, the Gaussian occupation count given $\mathcal{L}_{\mathrm{SI}}$ is accumulated as follows:

$$\begin{aligned}\tilde{N}_m &= E\left[N_m | \mathcal{L}_{\mathrm{SI}}\right] \\ &= \sum_{\text{all arcs } \kappa}\ \sum_{i=t(\kappa_s)}^{t(\kappa_e)} p\left(s_i = s^m, m_i = m | \mathbf{O}_1^T, \lambda_{\mathrm{SI}}\right) \\ &\quad \times p\left(\kappa | \mathbf{O}_1^T, \mathcal{L}_{\mathrm{SI}}, \lambda_{\mathrm{SI}}\right).\end{aligned} \tag{16}$$

The procedure for lattice-based speaker adaptation using (16) is performed by the following steps.

Unsupervised Adaptation Based upon the Lattice

Step 1) Generate the lattice $\mathcal{L}_{\mathrm{SI}}$ rather than the 1-best recognition result of adaptation data obtained by a speech recognizer.
Step 2) Calculate the posterior probability for each arc κ, $p(\kappa | \mathbf{O}_1^T, \mathcal{L}_{\mathrm{SI}}, \lambda_{\mathrm{SI}})$, using the forward-backward algorithm.
Step 3) Accumulate Gaussian occupation counts by (16).
Step 4) Update parameters with (11) for the MAP and (15) for the MLLR using $\tilde{N}_m$ instead of N_m.

D. Maximum Likelihood Linear Transformation for HMM-Based Speaker Verification

In this paper, we perform the HMM-based speaker verification with the maximum-likelihood linear transformation (MLLT) which improves the speech recognition performance [22], [23]. For the HMM-based speaker verification algorithm, we showed that using the MLLT can also improve the speaker verification performance [9]. Here, the MLLT was estimated when training the SI-HMMs representing the tonal segment models. For the GMM–UBM algorithm, another MLLT was estimated in training the UBM, which was shown to improve the performance of the GMM–UBM algorithm [24]. For both

the GMM–UBM and the HMM-based speaker verification algorithms, the MLLTs were estimated with diagonal-covariance constraints [22].

The MLLT is a global linear transformation that is designed to maximize the likelihoods of the training observation sequence. Let $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ be the mean and the variance of the jth Gaussian kernel, respectively. When a linear transformation $\boldsymbol{\theta}$ is applied to the training observation sequence $\mathbf{O}_1^T$, the log-likelihood given Gaussian kernels, and the transformation $\boldsymbol{\theta}$, $\log R\left(\mathbf{O}_1^T|\{\boldsymbol{\mu}_j,\bar{\boldsymbol{\Sigma}}_j\},\boldsymbol{\theta}\right)$ is given by

$$\begin{aligned}\log R&\left(\mathbf{O}_1^T|\{\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j\},\boldsymbol{\theta}\right)\\ &=-\frac{Nd}{2}\log 2\pi+N\log|\boldsymbol{\theta}|-\sum_{j=1}^{J}\frac{N_j}{2}\log|\boldsymbol{\Sigma}_j|\\ &\quad-\frac{1}{2}\sum_{j=1}^{J}\sum_{\mathbf{o}_i\in C_j}\left(\boldsymbol{\theta}^T\mathbf{o}_i-\boldsymbol{\mu}_j\right)^T\boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{\theta}^T\mathbf{o}_i-\boldsymbol{\mu}_j\right)\end{aligned}\tag{17}$$

where N, d, and J are the total number of data, the dimension of the data, and the number of Gaussian kernels, respectively. C_j and N_j denote the set of training observations corresponding to the jth Gaussian kernel and the number of observations $\{\mathbf{o}_i\in C_j\}$, respectively.

The maximum-likelihood (ML) estimates of $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ are given by

$$\hat{\boldsymbol{\mu}}_j=\frac{1}{N}\sum_{\mathbf{o}_i\in C_j}\boldsymbol{\theta}^T\mathbf{o}_i\tag{18}$$

and

$$\hat{\boldsymbol{\Sigma}}_j=\mathrm{Diag}\left(\boldsymbol{\theta}^T\bar{\mathbf{W}}_j\boldsymbol{\theta}\right)\tag{19}$$

where $\bar{\mathbf{W}}_j$ denotes the sample full-covariance matrix of observations $\{\mathbf{o}_i\in C_j\}$. $\mathrm{Diag}\left(\boldsymbol{\theta}^T\bar{\mathbf{W}}_j\boldsymbol{\theta}\right)$ is a diagonal matrix whose diagonal elements are those of $\boldsymbol{\theta}^T\bar{\mathbf{W}}_j\boldsymbol{\theta}$. Replacing $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ in (17) with $\hat{\boldsymbol{\mu}}_j$ and $\hat{\boldsymbol{\Sigma}}_j$ gives

$$\begin{aligned}\log R\left(\mathbf{O}_1^T|\{\hat{\boldsymbol{\mu}}_j,\hat{\boldsymbol{\Sigma}}_j\},\boldsymbol{\theta}\right)&=-\frac{Nd}{2}(1+\log 2\pi)+N\log|\boldsymbol{\theta}|\\ &\quad-\sum_{j=1}^{J}\frac{N_j}{2}\log\left|\mathrm{Diag}\left(\boldsymbol{\theta}^T\bar{\mathbf{W}}_j\boldsymbol{\theta}\right)\right|.\end{aligned}\tag{20}$$

The $\hat{\boldsymbol{\theta}}$ that maximizes the likelihood in (20) is given by

$$\hat{\boldsymbol{\theta}}=\underset{\boldsymbol{\theta}}{\mathrm{argmax}}\,f(\boldsymbol{\theta})\tag{21}$$

where

$$f(\boldsymbol{\theta})=N\log|\boldsymbol{\theta}|-\sum_{j=1}^{J}\frac{N_j}{2}\log\left|\mathrm{Diag}\left|\left(\boldsymbol{\theta}^T\bar{\mathbf{W}}_j\boldsymbol{\theta}\right)\right|\right|.\tag{22}$$

In this paper, $\hat{\boldsymbol{\theta}}$ is optimized using the David–Fletcher–Powell algorithm [25].

TABLE I
DESCRIPTION OF DATABASE FOR OUR EXPERIMENTS

Language	Mandarin Chinese
Number of speakers	500 (250/250 female/male speakers)
Number of sessions/speaker	1
Intersession interval	None
Type of speech	Read sentences About one million different sentences 200 sentences/speaker (20 min long/speaker)
Microphone	Wide-band electret microphone
Channels	Wide-band (16kHz sampling rate)
Acoustic environment	Quiet sound booth

TABLE II
EXPERIMENTAL SETUP

Background model	Trained on 90 h of data from 156/144 female/male speakers
Target speakers	9/10 female/male speakers Enrollment data of 2 min long/speaker
Trials	2 s long on average without silence 1832 true speaker trials 26 250 impostor trials from 180 impostors
Feature	39-dimensional MFCCs with the MLLT
Acoustic model	HMMs of 5434 tied states 16 Gaussian kernels per state
Language model	Free syllable decoding

III. EXPERIMENTS

A. Experimental Setup

Table I summarizes the information of the database used for our experiments. The database consists of about one million different read sentences from 500 Mandarin Chinese speakers. Each speaker's data amounts to 20 min long on average. This database was collected in a clean and wide-band (16-kHz sampling rate) environment and recorded in one session for each speaker.

Table II describes the experimental setup in this paper. The number of target, impostor, and background speakers are 19, 180, and 300, respectively. The three speaker sets are mutually exclusive. Each target speaker provides enrollment data of 2 min long (silence included). Both target and impostor-speaker test trials are 2 s long with 9.4 syllables on average. These trials

were created by selecting only speech intervals using a voice activity detector [26]. That is, silence in test trials was edited out. We applied the MLLT to 39-dimensional Mel-frequency cepstral coefficients (MFCCs) which consist of 13 coefficients (12 cepstral coefficients and energy), their delta and delta-delta time differences. Different MLLTs were applied to the GMM–UBM and the HMM-based speaker verification algorithms.

For HMM-based algorithms, a set of SI-HMMs, λ_{SI}, was used as a background model. λ_{SI} was trained on 90 h of data from 300 speakers (156/144 female/male speakers). The basic units of HMMs in λ_{SI} are tonal segments [27]. Each syllable is represented by 2 to 4 tonal segment models. By using a state tying algorithm in HTK [21], we modeled λ_{SI} as context-dependent HMMs of 5434 tied-states. The observation probability of each state was estimated by a mixture of 16 Gaussian kernels.

In Mandarin Chinese, a tonal syllable is a base-syllable with a corresponding lexical tone. The syllable-recognition performance in Mandarin Chinese can be either measured by tonal-syllable accuracy or base-syllable accuracy [10]. In tonal-syllable accuracy, both the base syllable and the corresponding tone must be correctly recognized. When evaluating the speech recognition performance in this paper, we used the tonal-syllable error rate rather than the tonal-syllable accuracy.

The syllable lattices were generated in a free syllable decoding network with λ_{SI}, and the tonal-syllable error rate of the 1-best recognition result was 45.3%. The GMM–UBM algorithm was implemented by modifying the HTK [21] where 1 emitting state HMM was used, and the state transitions in HMM were ignored. For the GMM–UBM algorithm, the UBM λ_{UBM} was trained using the same training data that was used to train λ_{SI}. The model λ_{UBM} was created by pooling two 1024 GMMs together, one for male speech and the other one for female speech as described in [3]. The speaker verification performance was measured by the detection error tradeoff (DET) curve and the equal error rate (EER) [28].

B. Experimental Results

1) Speaker Verification With the 1-Best and Syllable-Lattice Scoring: We evaluated the HMM-based speaker verification performance with various speaker adaptation algorithms. Fig. 4 illustrates the speaker verification performances of the 1-best scoring and the lattice rescoring with the MAP, the MLLR, and the MLLR followed by the MAP algorithms. The MLLR followed by the MAP is abbreviated as the MLLR + MAP. For all speaker adaptation algorithms, the lattice rescoring outperformed the 1-best scoring. Fig. 5 illustrates the speaker verification performances with the lattice-based MLLR and the lattice-based MLLR followed by the MAP which are abbreviated as the LAT-MLLR and the LAT-MLLR + MAP.

Comparing the EERs in Figs. 4 and 5, the speaker verification algorithms with the LAT-MLLR and the LAT-MLLR + MAP outperform those using the MAP, the MLLR, and the MLLR + MAP. When the lattice-based speaker adaptation algorithms were used, the performance gap between the 1-best scoring and the lattice rescoring decreased.

In both 1-best scoring and lattice rescoring, the HMM-based speaker verification algorithm with the LAT-MLLR + MAP achieved the lowest EER followed by that with the LAT-MLLR,

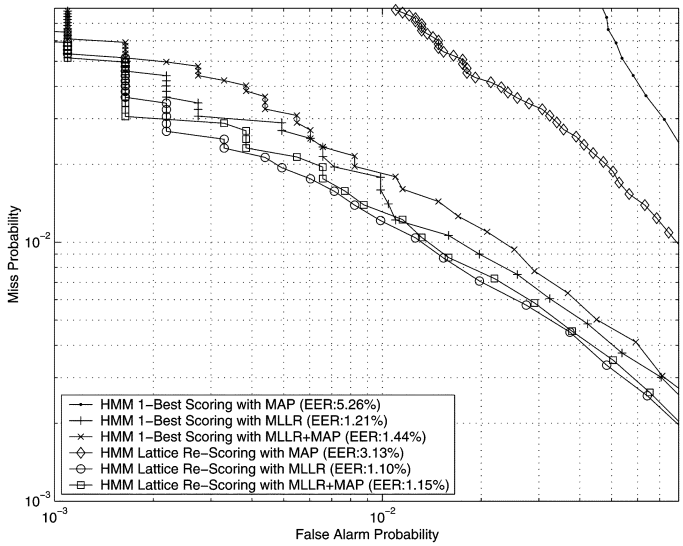

Fig. 4. DET curves of HMM 1-best scoring with MAP, HMM 1-Best scoring with MLLR, HMM 1-best scoring with MLLR + MAP, HMM lattice rescoring with MAP, HMM lattice rescoring with MLLR, and HMM lattice rescoring with MLLR + MAP.

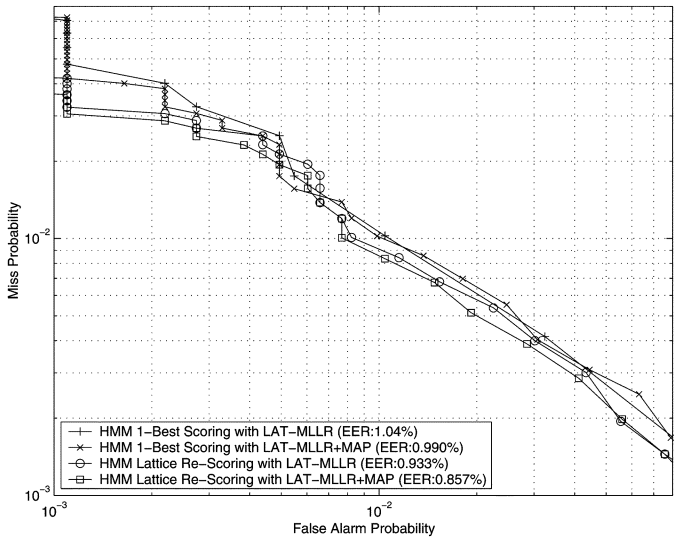

Fig. 5. DET curves of HMM 1-best scoring with LAT-MLLR, HMM 1-best scoring with LAT-MLLR + MAP, HMM lattice rescoring with LAT-MLLR, and HMM lattice rescoring with LAT-MLLR + MAP.

the MLLR, the MLLR + MAP, and the MAP. The same tendency is found at the tonal-syllable error rates of the speaker-adapted models in the unsupervised mode illustrated in Fig. 6. The tonal-syllable error rates were evaluated using 950 utterances from 19 target speakers. In the unsupervised mode, the speaker-adapted models used in the tonal-syllable recognition experiments are exactly the same as the target-speaker models used in the speaker verification experiments. The LAT-MLLR + MAP achieved the lowest tonal-syllable error rate followed by the LAT-MLLR, the MLLR, the MLLR + MAP, and the MAP. The MAP and the MLLR + MAP achieved higher error rate than the SI model and the MLLR since highly errorful recognition results can lead to incorrectly adapted models in the MAP. In the supervised mode, the MAP slightly reduced the

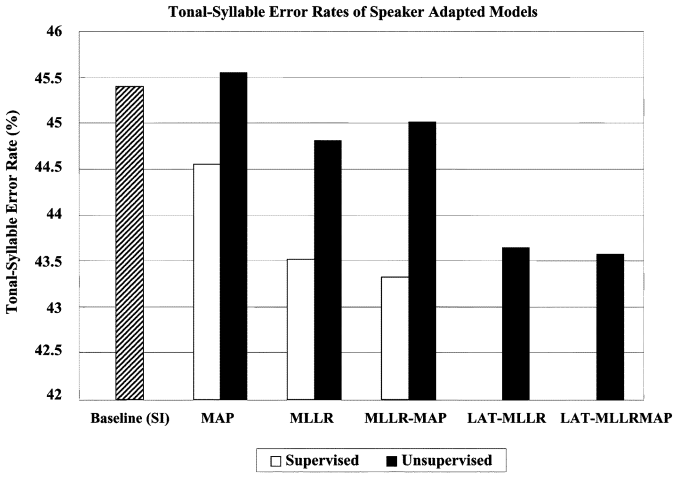

Fig. 6. Tonal-syllable error rate of speaker-adapted models using the MAP, the MLLR, the MLLR + MAP, the LAT-MLLR, and the LAT-MLLR + MAP.

TABLE III
LATTICE COMPLEXITY AND SPEAKER VERIFICATION PERFORMANCE

Number of tokens	2	3	4
EER	0.933%	0.857%	0.930%
Number of syllables/lattice	46.4	117	234
Number of arcs/node	1.81	2.68	3.55
Lattice error rate	23.7%	18.2%	15.4%
Lattice density	5.32	13.4	27.2

error rate. In contrast, when using the MLLR, the error rate decreased since similar models were clustered before adaptation. The LAT-MLLR + MAP outperformed the LAT-MLLR, which implies that the lattice-based speaker adaptation algorithms are more robust to recognition errors than the conventional unsupervised adaptation in both speech recognition and speaker verification applications.

2) Lattice Complexity: The lattices for the speaker adaptation and the speaker verification were generated by the token passing algorithm implemented in the HTK [21]. We used three tokens for experiments in Section III-BI. Table III enumerates the EER, the number of syllables/lattice, the number of arcs/node, the lattice error rate (LER), and the lattice density when using 2, 3, and 4 tokens. The LER is the lower bound on the tonal-syllable error rate from the decoded lattice, and the lattice density is the number of arcs in the lattice divided by the number of syllables in the true transcription. Here, the LER and the lattice density were computed on the 950 utterances from the target speakers, which were used to evaluate the tonal-syllable error rate of speaker-adapted models. The others were computed on the syllable lattices used for the speaker verification. The EER was measured using the proposed algorithm with the LAT-MLLR + MAP.

As shown in Table III, using more tokens generates the lattice with larger lattice density and smaller LER. However in

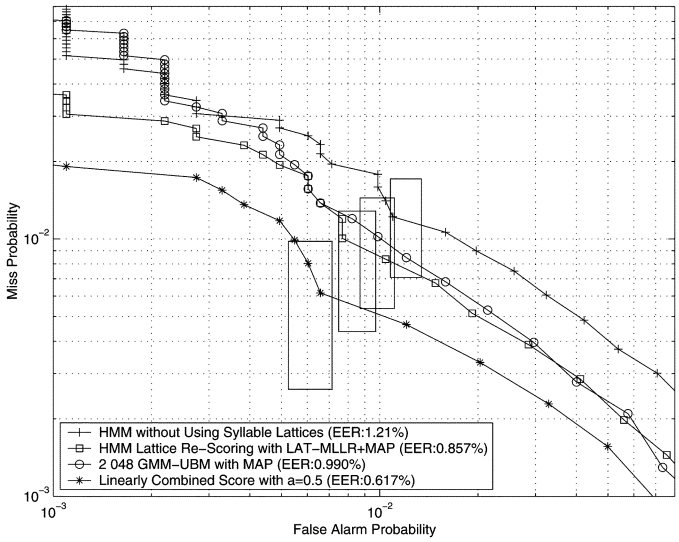

Fig. 7. DET curves of HMM without using syllable lattices, HMM lattice rescoring with LAT-MLLR + MAP, 2048 GMM–UBM with MAP, and linearly combined score with $a = 0.5$: boxes denote the 95% confidence intervals at EER operating points.

speaker verification, the number of tokens should be properly set to achieve the lowest EER. Our conjecture for this is that when the lattice with a large density is used, many confusing hypotheses are also incorporated in the speaker adaptation and the lattice rescoring, which can deteriorate the speaker verification performance.

3) Combination of the Proposed and the GMM–UBM Algorithm: Fig. 7 illustrates the DET curves of the GMM–UBM algorithm with 2048 Gaussian kernels, the HMM-based speaker verification algorithm without using syllable lattices, and the proposed algorithm with the LAT-MLLR + MAP. Here, the HMM-based speaker verification without using syllable lattices is the HMM-based speaker verification algorithm with the 1-best scoring and the MLLR in Fig. 4 which yielded the best performance among the HMM-based speaker verification algorithms that do not use syllable lattices. In addition, the test statistics of the proposed algorithm and the GMM–UBM algorithm were linearly combined as follows:

$$p_{\text{comb}}\left(\text{TAR}|\mathbf{O}_1^T\right) = (1-a) * p_{\text{LAT-SCR}}\left(\text{TAR}|\mathbf{O}_1^T\right) + a * p_{\text{GMM-UBM}}\left(\text{TAR}|\mathbf{O}_1^T\right) \quad (23)$$

where $p_{\text{LAT-SCR}}(\text{TAR}|\mathbf{O}_1^T)$ is the posterior probability of the proposed algorithm calculated using (9), and $p_{\text{GMM-UBM}}\left(\text{TAR}|\mathbf{O}_1^T\right)$ is the posterior probability of the GMM–UBM calculated in the same manner. Boxes of DET curves in Fig. 7 denote the 95% confidence intervals at EER operating points. By assuming that all target and impostor-speaker trials are independent and the number of target and impostor-speaker trials are large, the false alarm and the miss probabilities follow approximately normal distributions whose standard deviations are given by [29]

$$\sigma_{fa} = \sqrt{\frac{P_{fa}(1-P_{fa})}{N_{\text{imp}}}} \quad (24)$$

and

$$\sigma_{\text{miss}} = \sqrt{\frac{P_{\text{miss}}(1 - P_{\text{miss}})}{N_{\text{tar}}}} \tag{25}$$

where P_{fa}, P_{miss}, N_{imp}, and N_{tar} denote the false alarm probability, the miss probability, the number of impostor-speaker trials, and the number of target-speaker trials, respectively. The 95% confidence interval at operating point $(P_{fa}, P_{\text{miss}})$ is given by the set of $\left(P^*_{fa}, P^*_{\text{miss}}\right)$ satisfying

$$P_{fa} - 1.96\sigma_{fa} \le P^*_{fa} \le P_{fa} + 1.96\sigma_{fa}$$
$$P_{\text{miss}} - 1.96\sigma_{\text{miss}} \le P^*_{\text{miss}} \le P_{\text{miss}} + 1.96\sigma_{\text{miss}}.$$

The 95% confidence interval at the EER operating point of the proposed algorithm does not overlap with that of the HMM-based speaker verification algorithm without using syllable lattices. This implies that the proposed algorithm achieves better performance in terms of the EER than the HMM-based speaker verification algorithm without using syllable lattices. The 95% confidence interval of the proposed algorithm at the EER operating point overlaps with that of the GMM–UBM algorithm: the confidence intervals do not ovelap when the confidence are smaller than 75%. The linearly combined score in (23) achieved the best performance in terms of the EER: as shown in Fig. 7, the linearly combined score achieved lower miss probability than both the GMM–UBM and the proposed algorithm at the same false alarm probability.

The GMM–UBM algorithm captures the speaker characteristics without utilizing any temporal information, while the HMM-based speaker verification algorithm captures the pronunciation difference following the decoded state sequence (in 1-best recognition results or lattices) with the Markov assumption. The linear combination of scores from the proposed and the GMM–UBM algorithms can capture speaker characteristics in different aspects and yields better performance than either algorithm.

IV. Conclusion

This paper proposed a syllable lattice-based rescoring for speaker verification in Mandarin Chinese. The proposed algorithm outperformed the HMM-based algorithm without using syllable lattices. Unlike the GMM–UBM algorithm in which the MAP adaptation outperforms the MLLR or the eigenvoices adaptation [16], the LAT-MLLR + MAP performed the best, followed by the LAT-MLLR, the MLLR, the MLLR + MAP, and the MAP. One reason for this is that the MAP deteriorates the speaker-adapted HMMs because of the highly errorful 1-best recognition results. The lattice-based speaker adaptation further improves the proposed HMM-based speaker verification algorithm.

The proposed algorithm adopted context-dependent HMMs, which are used for SI speech recognition, for speaker verification application. Thus, the complexity of the proposed algorithm is intrinsically higher than that of the GMM–UBM algorithm. In our experiment, context-dependent tonal segment models of 5434 tied states were used. However, the increased complexity is not superfluous since our proposed system facilitates not only speaker verification but also syllable recognition. Additionally, by linearly combining the score of the proposed algorithm and that of the GMM–UBM algorithm, we were able to further improve the speaker verification performance.

The proposed algorithm was evaluated using syllable lattices for Mandarin Chinese which is a syllabically paced language. Thus, this algorithm is expected to be equally effective for syllabically paced languages such as Cantonese Chinese or Vietnamese. The performance with respect to other languages such as English needs further investigations. Our future work is focused on 1) reducing the number of Gaussian kernels and 2) modifying this algorithm to English speakers (e.g., NIST speaker recognition evaluation) task.

References

[1] A. E. Rosenberg, C.-H. Lee, and F. K. Soong, "Sub-word unit talker verification using hidden Markov models," in *Proc. ICASSP*, 1990, vol. 1, pp. 269–272.

[2] T. Matsui and S. Furui, "Concatenated phoneme models for text-variable speaker recognition," in *Proc. ICASSP*, 1993, vol. 2, pp. 391–394.

[3] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Process.*, vol. 10, pp. 19–41, 2000.

[4] F. Weber, B. Peskin, M. Newman, A. C. Emmanuel, and L. Gillick, "Speaker recognition on single- and multispeaker data," *Digital Signal Process.*, vol. 10, pp. 75–92, 2000.

[5] G. Doddington, "Speaker recognition based on idiolectal differences between speakers," in *Proc. Eurospeech*, 2001, vol. 4, pp. 2517–2520.

[6] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proc. ICASSP*, 1992, vol. 1, pp. 517–520.

[7] W. D. Andrews, M. A. Kohler, and J. P. Campbell, "Phonetic speaker recognition," in *Proc. Eurospeech*, 2001, pp. 2517–2520.

[8] A. Hatch, B. Peskin, and A. Stolcke, "Improved phonetic speaker recognition using lattice decoding," in *Proc. ICASSP*, 2005, vol. 1, pp. 169–172.

[9] M. Jin, F. K. Soong, and C. D. Yoo, "Syllable lattice based rescoring for speaker verification," in *Proc. ICASSP*, 2006, vol. 1, pp. 921–924.

[10] L.-S. Lee, "Voice dictation of Mandarin Chinese," *IEEE Signal Process. Mag.*, vol. 14, no. 4, pp. 63–101, Jul. 1997.

[11] H.-M. Wang, "Experiments in syllable-based retrieval of broadcast news speech in Mandarin Chinese," in *Speech Commun.*, 2000, vol. 32, pp. 49–60.

[12] M. Padmanabhan, G. Saon, and G. Zweig, "Lattice-based unsupervised MLLR for speaker adaptation," in *Proc. ISCA ITRW Automatic Speech Recognition*, 2000, pp. 128–131.

[13] L. F. Uebel and P. C. Woodland, "Speaker adaptation using lattice-based MLLR," in *Proc. ISCA ITRW Adaptation Methods in Speech Recognition*, 2001, pp. 57–60.

[14] T. J. Hazen, D. A. Jones, A. Park, L. C. Kukolich, and D. A. Reynolds, "Integration of speaker recognition into conversational spoken dialogue systems," in *Proc. Eurospeech*, 2003, pp. 1961–1964.

[15] J. L. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone lattices," in *Proc. ICSLP*, 2004, pp. 25–28.

[16] J. Mariéthoz and S. Bengio, "A comparative study of adaptation methods for speaker verification," in *Proc. ICSLP*, 2002, pp. 581–584.

[17] C.-H. Lee, C.-H. Lin, and B.-H. Juang, "A study on speaker adaptation of the parameters of continuous density hidden Markov models," *IEEE Trans. Signal Process.*, vol. 39, no. 4, pp. 806–814, Apr. 1991.

[18] J.-L. Gauvain and C.-H. Lee, "Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 291–298, Apr. 1994.

[19] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," in *Comput. Speech Lang.*, 1995, vol. 9, pp. 171–185.

[20] R. Kuhn, P. Nguyen, J. C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini, "Eigenvoices for speaker adaptation," in *Proc. ICSLP*, 1998, pp. 1771–1774.

[21] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[22] R. A. Gopinath, "Maximum likelihood modeling with Gaussian distributions for classification," in *Proc. ICASSP*, 1998, pp. 661–664.
[23] P. A. Olsen and R. A. Gopinath, "Modeling inverse covaraince matrices by basis expansion," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 1, pp. 37–46, Jan. 2004.
[24] G. N. Ramaswamy, J. Navratil, U. V. Chaudhari, and R. D. Zilca, "The IBM system for the NIST-2002 cellular speaker verification evaluation," in *Proc. ICASSP*, 2003, vol. 2, pp. 61–64.
[25] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization," *Comput. J.*, vol. 6, pp. 163–168, 1963.
[26] Y. Tian, Z. Wang, and D. Lu, "Nonspeech segment rejection based on prosodic information for robust speech recognition," *IEEE Signal Process. Lett.*, vol. 9, no. 11, pp. 364–367, Nov. 2002.
[27] C. Huang, Y. Shi, J. Zhou, M. Chu, T. Wang, and E. Chang, "Segmental tonal modeling for phone set design in Mandarin LVCSR," in *Proc. ICASSP*, 2004, pp. 901–904.
[28] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proc. Eurospeech*, 1997, pp. 1895–1898.
[29] D. A. van Leeuwen, A. F. Martin, M. A. Przybocki, and J. S. Bouten, "NIST and NFI-TNO evaluations of automatic speaker recognition," in *Comput. Speech Lang.*, 2006, vol. 20, pp. 128–158.

Minho Jin (S'06) received the B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, in 2002 and 2004, respectively. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering, KAIST.

His research interests are speaker verification, speech recognition, multimedia retrieval, and machine learning.

Frank K. Soong (S'76–M'82–SM'91) received the Ph.D. degree from Stanford University, Stanford, CA.

He joined Bell Labs Research, Murray Hill, NJ, in 1982 as a Member of Technical Staff and retired as Distinguished Member of Technical Staff in 2001. From 2002 to 2004, he was as an Invited Researcher at the Spoken Language Translation Labs, ATR, Kyoto, Japan. Currently, he is with Microsoft Research Asia (MSRA), Beijing, China, and leads the Speech Research Group there. Over the years, he had worked on various aspects of speech research, including speech and speaker recognition; speech segmentation, analysis, and coding; stochastic modeling of speech signals; efficient search of multiple hypotheses via dynamic programming; discriminative training of HMMs; dereverberation of audio signals; microphone array signal processing; acoustic echo cancellation; and hands-free speech recognition in a noisy environment. He was responsible for transferring advanced speech recognition technology to AT&T voice-activated cell phones which were rated by the Mobile Office Magazine as the best among many competing products evaluated. His tree-trellis algorithm for finding the n-best sentence hypotheses, forms the core of the popular free software JULIUS developed in Japan for speaker-independent, large-vocabulary, continuous-speech recognition application. He has visited Japan twice as an Invited Researcher, first, from 1987 to 1988, at the NTT Electro-Communication Labs, Musashino, Tokyo, then from 2002 to 2004 at ATR. He is a Visiting Professor of the Chinese University of Hong Kong (CUHK) and the codirector of the CUHK-MSRA Joint Research Lab. He has published more than 100 technical papers in the field of speech.

Dr. Soong was the corecipient of the Bell Labs President Gold Award for developing the Bell Labs Automatic Speech Recognition (BLASR) software package. He cochaired the 1991 IEEE International Arden House Speech Recognition Workshop. He has served on the IEEE Speech Technical Committee of the Signal Processing Society and as a committee member and Associate Editor of the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING.

Chang D. Yoo (S'92–M'96) received the B.S. degree in engineering and applied science from the California Institute of Technology, Pasadena, in 1986, the M.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1988, and the Ph.D. degree in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, in 1996.

From January 1997 to March 1999, he worked at Korea Telecom as a Senior Researcher. He joined the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, in April 1999. From March 2005 to March 2006, he was with the Research Laboratory of Electronics, MIT. His current research interests are in the application of machine learning and digital signal processing in multimedia. He is a member of Tau Beta Pi and Sigma Xi.