Available online at www.sciencedirect.com
ScienceDirect

Pattern Recognition 40 (2007) 3110–3119

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Incremental MLLR speaker adaptation by fuzzy logic control

Ing-Jr Ding*

Department of Computer Science, National Chiao Tung University, Hsin-Chu 30050, Taiwan, ROC

Received 5 April 2006; received in revised form 22 January 2007; accepted 30 January 2007

Abstract

This paper presents a fuzzy control mechanism for conventional maximum likelihood linear regression (MLLR) speaker adaptation, called FLC-MLLR, by which the effect of MLLR adaptation is regulated according to the availability of adaptation data in such a way that the advantage of MLLR adaptation could be fully exploited when the training data are sufficient, or the consequence of poor MLLR adaptation would be restrained otherwise. The robustness of MLLR adaptation against data scarcity is thus ensured. The proposed mechanism is conceptually simple and computationally inexpensive and effective; the experiments in recognition rate show that FLC-MLLR outperforms standard MLLR especially when encountering data insufficiency and performs better than MAPLR at much less computing cost.

Keywords: Speech recognition; Speaker adaptation; Hidden Markov model; Maximum likelihood linear regression; T–S fuzzy logic controller

1. Introduction

Speech recognition systems can be classified either as speaker-independent type (SI) or speaker-dependent type (SD), depending on how speech samples are collected during system construction. An SI system typically collects speech samples from a as large population of speakers as possible, whereas a SD system collects a large amount of sample data from possibly just one designated speaker. In general, a well-trained SD model achieves better performance than an SI model on recognizing the speech of a specific speaker. However, when the amount of training data available to acquire the SD model is not sufficient, such superiority would no longer exist. This is where speaker-adaptive techniques (SA), sometimes referred to as model-based adaptation techniques, get in to play, which would adapt a full SI model into an SD one and achieves SD-like performance, requiring only a small fraction of the speaker-specific training data.

To the author's knowledge, there are mainly three categories of SA techniques: (1) Bayesian-based, (2) transformation-based, and (3) speaker-clustering-based (also known as Eigenvoice-based). In the Bayesian-based model adaptation, the acoustic model parameters are directly re-estimated, using maximum *a posteriori* (MAP) adaptation [1,2], for example, under the Bayesian reasoning framework. In the transformation-based model adaptation, certain appropriate transformations have to be derived from a set of adaptation utterances of a new speaker and then applied to clusters of hidden Markov model (HMM) parameters. The form of transformation can be as simple as adding a cepstral bias for model adaptation, as was done in Ref. [3], or an affine transformation over HMM parameters when adding a bias alone could not take care of the variations in test environments or among different speakers. In the work by Leggetter et al. [4], maximum likelihood linear regression (MLLR) was firstly proposed as the framework for affine transformation. The MLLR scheme has been quite successful for its rapid adaptation and a series of variants arise, many of which aiming at the problem concerning the quality of the estimated transformation resulting from insufficient adaptation data. For instance, the transformation parameters were estimated by maximizing the posterior density using MAP theory [5,6], or by using a prior distribution for calculating the mean transformation matrix parameters as suggested in Refs. [7,8] and thus dubbed as maximum *a posteriori* linear regression (MAPLR), or by using the so-called discounted likelihood estimation as was proposed

* Tel./fax: +886 2 27990171.
E-mail address: ingjr.csie90g@nctu.edu.tw.

0031-3203/$30.00 © 2007 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2007.01.027

in Ref. [9]. All were designed for the robust estimation of transformation against the insufficiency of adaptation data and were essentially complicated and time consuming in computation, an adverse factor against on-line applications.

Kuhn et al. [10] proposed the eigenvoice adaptation where *a priori* knowledge concerning the variations among all training speakers was represented as the set of SD model parameters in the form of eigenvectors named eigenvoices; a new speaker model was then expressed as the linear combination of the set of eigenvoices. Following that, the eigenvoice versions of conventional MLLR and MAPLR adaptation have been reported in Refs. [11,12], respectively. Eigenvoice-based approach has received intensive attention and various extensions of eigenvoice adaptation have been developed recently [13–18].

In order to tackle the issue of unreliable MLLR model transformation due to the scantiness of training data without the daunting cost of MAPLR-like adaptation, a fuzzy control mechanism is proposed in this paper so that, based on the amount of adaptation utterances available, MLLR transformation could be regulated in the way that the rapidness of MLLR adaptation could be fully exploited when the amount of training data allows, while the undesired effect of poor MLLR adaptation would be alleviated. In fact, the fuzzy theory was firstly used in MLLR adaptation (fuzzy-MLLR) as described in Ref. [19] where the motivation and purpose of using fuzzy mechanism were completely different from the work here, as will be detailed in Section 2.5. Fuzzy logic controllers (FLC) have been deployed in a wide range of applications with great success [22], including speech recognition too. Takagi–Sugeno (T–S) FLC is a type of fuzzy process renowned for its idea that behaviors of a seemingly complex system can be represented by a set of fuzzy implications (or rules if one preferred) from which the system output is derived systematically [20,21].

The rest of the paper is organized as follows. In Section 2, theoretical formulation of MLLR and MAPLR is briefly described first. Then the idea of incremental model transformation under fuzzy regulation is introduced, followed by the formulation of the T–S fuzzy mechanism for model adaptation in this work. Complexity analysis of the proposed scheme is also given, following which a reference to SA techniques employing fuzzy schemes is mentioned. Section 3 presents the experiment results where the effectiveness and performance of FLC-MLLR are fully demonstrated as compared to standard MLLR and MAPLR. Some concluding remarks are given in Section 4.

2. FLC-MLLR adaptation

Under the framework of transformation-based speaker adaptation, it starts with a set of SI HMM, Λ, to which certain transformations F_η with parameters η derived from adaptation data, Y, of a new speaker are to be applied such that the transformed model $F_\eta(\Lambda)$ would recognize the incoming speech better than Λ did. The transformation parameters η are usually assumed to be fixed and then be estimated via statistical measures with specific criteria such as maximum likelihood (ML) or MAP, as were done in Refs. [4,7], respectively.

2.1. MLLR and MAPLR

MLLR is quite popular because of the simplicity of ML criterion, which states that the transformed model $\hat{\eta}_{ML}$ should maximize the likelihood of the adaptation data $p(Y|\Lambda,\eta)$, i.e.

$$\hat{\eta}_{ML} = \arg\max_{\eta} p(Y|\Lambda,\eta). \tag{1}$$

Consider the Gaussian mean vector of the model at state s, μ_s, and the associated affine transformation action as follows:

$$\hat{\mu}_s = A_s\mu_s + b_s, \tag{2}$$

which sometimes is written as

$$\hat{\mu}_s = W_s\xi_s, \tag{3}$$

and ξ_s is the extended mean vector in the form

$$\xi_s = [\omega, \mu_{s_1}, \ldots, \mu_{s_n}]', \tag{4}$$

where ω is the offset term of the regression, usually being set as 1.

The transformation matrix W_s is estimated such that the likelihood of the adaptation data is maximized, for which a closed form solution is available in Ref. [4] as follows:

$$\sum_{t=1}^{T}\sum_{r=1}^{R}\gamma_{s_r}(t)\Sigma_{s_r}^{-1}o_t\xi'_{s_r} = \sum_{t=1}^{T}\sum_{r=1}^{R}\gamma_{s_r}(t)\Sigma_{s_r}^{-1}W_s\xi_{s_r}\xi'_{s_r}, \tag{5}$$

where $\gamma_{s_r}(t)$ is the total occupation probability for the state s_r at time t given the generation of the observation vectors of adaptation data o_t at time t, $\Sigma_{s_r}^{-1}$ is the covariance matrix of the output probability distribution, and R is the number of states. W_s in Eq. (5) can be solved by the expectation–maximization (E–M) algorithm as suggested in Ref. [23].

Besides the adaptation of Gaussian mean vectors, other HMM parameters like Gaussian variance can also be adapted as was done in Ref. [24] where the transform action for variance vectors is estimated following the transformation of mean vectors. However, the improvement of the recognition performance is extremely limited and, as a result, only the adaptation of the mean vectors is considered in this work.

When the amount of adaptation data is very small, it is unlikely to calculate a regression matrix for each state. In such a case, there would be no alternative but tying all states with one regression matrix to adapt all Gaussian distributions.

Apart from MLLR, Chesta et al. [7] suggest that the prior density can be taken into account in the estimation process of transformation parameters by using an MAP criterion:

$$\hat{\eta}_{MAP} = \arg\max_{\eta} p(\eta|Y, \Lambda) \tag{6}$$

which is proportional to $\arg\max_\eta p(Y|\eta, \Lambda)p(\eta)$. According to this criterion, the MAPLR technique for adaptation is thus derived, where the transformation matrix W_s appears in the form of $p \times (p+1)$ linear equations as follows [7]:

$$\begin{aligned}&\sum_{k=1}^{p}\sum_{l=1}^{p+1} w_{kl}\left[\sum_{n=1}^{N}\sum_{m=1}^{M}\left(\sum_{t=1}^{T}\gamma_t(n,m)\right) r_{ik}\overline{\mu}_l\overline{\mu}_j + \frac{1}{2}\sigma_{ki}\phi_{jl} + \frac{1}{2}\sigma_{ik}\phi_{lj}\right]\\ &\quad = \sum_{k=1}^{p}\sum_{l=1}^{p+1}\left[\sum_{n=1}^{N}\sum_{m=1}^{M}\left(\sum_{t=1}^{T}\gamma_t(n,m)o_k(t)\right) r_{ik}\overline{\mu}_j + \frac{1}{2}\sigma_{ki}m_{kl}\phi_{jl} + \frac{1}{2}\sigma_{ik}m_{kl}\phi_{lj}\right], \qquad 1 \leqslant i \leqslant p, \quad 1 \leqslant j \leqslant p+1,\end{aligned} \tag{7}$$

where $w_{ij} \in W_s, \gamma_{ij} \in R_{nm}, m_{ij} \in M, \sigma_{ij} \in \Sigma, \phi_{ij} \in \Phi, \gamma_t(n,m)$ is the probability of the mixture m in state n at time t, given the observation $o(t)$, and $\overline{\mu}_i$ is the ith component of the mean vector μ_{nm}.

Note that R_{nm} is the precision matrix and M, Σ, and Φ are hyperparameter matrices associated with the prior density. Solving the system of equations in Eq. (7) for W_s is obviously much more time consuming than standard MLLR due to the use of additional hyperparameters $\{M, \Sigma, \Phi\}$ of the prior distribution. Details for the estimation of $\{M, \Sigma, \Phi\}$ can be found in Ref. [25].

2.2. *Incremental model transformation under fuzzy regulation*

As already noted, when the amount of adaptation data is sufficient (though may be small), the model-transformed adaptation scheme would be quite effective and the performance improvement saturates quickly as the amount of adaptation data increases. However, when only a limited and insufficient amount of adaptation data are available, the robustness of the transformation matrix W_s, especially derived by the MLLR approach, would be in doubt; poor estimation of W_s could lead to the corruption of underlying structure of the acoustic space. The problem due to the scarcity of adaptation data can be alleviated by utilizing the MAPLR scheme instead, if one disregards the heavy computation involved.

With insufficient training data, one would naturally tend to be more "conservative" while using the transformation matrix thus derived, i.e. the effect of the adaptation should be restricted in this case so that the adapted mean vector would not vary too much from the state prior to adaptation. Accordingly, an incremental approach to MLLR model transformation is proposed as follows:

$$\widetilde{\mu}_s = \alpha\mu_s + (1-\alpha)W_s\xi_s, \quad 0 \leqslant \alpha \leqslant 1, \tag{8}$$

where μ_s is the initial mean vector, ξ_s is the extended mean vector as defined in Eq. (4), and W_s is the transformation matrix derived from Eq. (5). Note that the weight α is to vary in a way depending on how much confidence one has in W_s. A possibly not so well-estimated W_s due to insufficient adaptation data would preferably go with α approaching 1 so that $\widetilde{\mu}_s$ stays closer to μ_s, instead of drifting away drastically. On the opposite, 0-approaching α should be taken for full advantage of fast adaptation. Owing to the uncertainty in the amount of adaptation data, the control of α weighting is formulated as a fuzzy control problem governed by the simple rule: the more the training data are, the smaller α will be.

2.3. *T–S fuzzy model and speaker adaptation*

Under the framework of T–S fuzzy model, a generic system can be formulated as a set of fuzzy implications (or rules) together with a system output determined by consequences in the set of implications.

The system representation would be of the form:

Rule 1: If $x(1)$ is A_1^1 and ... and $x(n)$ is A_n^1 then

$$y^1 = a_0^1 + a_1^1 x(1) + \cdots + a_n^1 x(n),$$

$$\vdots$$

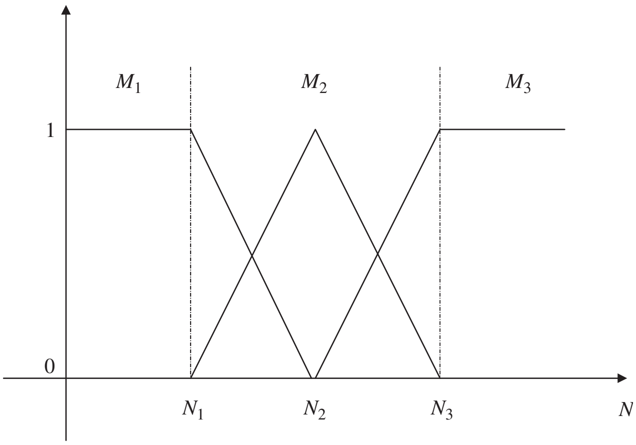

Fig. 1. Membership functions.

Rule i: If $x(1)$ is A_1^i and ... and $x(n)$ is A_n^i then

$$y^i = a_0^i + a_1^i x(1) + \cdots + a_n^i x(n),$$

$$\vdots \tag{9}$$

Rule l: IF $x(1)$ is A_1^l and ... and $x(n)$ is A_n^l then

$$y^l = a_0^l + a_1^l x(1) + \cdots + a_n^l x(n).$$

System output: $y = \sum_{i=1}^{l} w^i y^i / \sum_{i=1}^{l} w^i$, given that $w^i = \prod_{p=1}^{n} A_p^i(x(p))$, for a system of n inputs and l implications. Note that A_p^i $(p = 0, 1, \ldots, n)$ are fuzzy sets and $A_p^i(x(n))$ denote the fuzzy values of the membership function associated with A_p^i for the input $x(n)$; a_p^i $(p = 0, 1, \ldots, n)$ are consequent parameters through which the ith consequence y^i is expressed as a linear combination of n inputs.

For the specific problem in this paper during speaker adaptation, the aforementioned simple rule governing α regulation, given N training samples observed for all Gaussian mixture components, can be formulated as the following implications:

Rule 1: If N is small then α is large.

Rule 2: If N is medium then α is medium.

Rule 3: If N is large then α is small.

Let $M_1(N)$, $M_2(N)$, and $M_3(N)$ be membership functions associated, respectively, with small, medium, and large amounts of training data available for adaptation, as shown in Fig. 1, and α_L, α_M, and α_S be the α values determined, respectively, by functions $f_1(N)$, $f_2(N)$, and $f_3(N)$ in each of the three cases.

Then the previous set of rules can be further clarified as

Rule 1: If N is $M_1(N)$ then $\alpha_L = f_1(N)$.

Rule 2: If N is $M_2(N)$ then $\alpha_M = f_2(N)$.

Rule 3: If N is $M_3(N)$ then $\alpha_S = f_3(N)$,

where

$$M_1(N) = \begin{cases} 1, & N \leqslant N_1, \\ \dfrac{N_2 - N}{N_2 - N_1}, & N_1 \leqslant N \leqslant N_2, \\ 0, & N \geqslant N_2, \end{cases} \quad M_2(N) = \begin{cases} 0, & N \leqslant N_1 \text{ or } N \geqslant N_3, \\ \dfrac{N - N_1}{N_2 - N_1}, & N_1 < N \leqslant N_2, \\ \dfrac{N_3 - N}{N_3 - N_2}, & N_2 \leqslant N < N_3, \end{cases} \quad M_3(N) = \begin{cases} 0, & N \leqslant N_2, \\ \dfrac{N - N_2}{N_3 - N_2}, & N_2 < N < N_3, \\ 1, & N \geqslant N_3, \end{cases}$$

together with the implication functions

$$f_1(N) = a_1 \cdot N + b_1,\ f_2(N) = a_2 \cdot N + b_2,\ f_3(N) = a_3 \cdot N + b_3,$$

and the final system output

$$\alpha = \frac{\sum_{i=1}^{3} M_i(N) f_i(N)}{\sum_{i=1}^{3} M_i(N)}. \tag{10}$$

By Eq. (10), it is observed that for $N < N_1$, α is solely determined by $f_1(N)$, i.e. $\alpha = \alpha_L$, whereas for $N > N_3$, α is determined by $f_3(N)$ alone. In the case that N is around N_2, α is determined by $f_2(N)$ since $M_2(N)$ is much greater than $M_1(N)$ and $M_3(N)$.

The system now has nine hyperparameters (a_1, a_2, a_3, b_1, b_2, b_3, N_1, N_2, and N_3) to be fixed, for which an iterative process is developed as follows:

Step 1: Let $N_1 : N_2 : N_3 = 1 : 2 : 3$ and take 500 as the initial value of N_1.

Step 2: Estimate the parameters a_1 and b_1 under the condition $N < N_1$, wherein $M_1(N) = 1$, $M_2(N) = M_3(N) = 0$, and

$$\alpha = \frac{M_1(N) f_1(N)}{M_1(N)} = f_1(N) = a_1 \cdot N + b_1.$$

The procedure for fixing a_1 and b_1 is explained in the following pseudo-code sequence:

```
$a_1 =$ initial value; $b_1 = 0$; $k = 0$;
$F^0 =$ baseline_recognition_rate;
$a_1 += \Delta a_1$; $k++$;
$F^k =$ speech_recognition($\alpha = a_1 \cdot N + b_1$, testing_utterances);
if($F^k > F^{k-1}$)
    Repeat
        {$a_1 += \Delta a_1$; $k++$;
         $F^k =$ speech_recognition($\alpha = a_1 \cdot N + b_1$, testing_utterances);
        } while ($F^k > F^{k-1}$);
else
    Repeat
        {$a_1 -= \Delta a_1$; $k++$;
         $F^k =$ speech_recognition($\alpha = a_1 \cdot N + b_1$, testing_utterances);
        } while ($F^k > F^{k-1}$);
$b_1 += \Delta b_1$; $k++$;
$F^k =$ speech_recognition($\alpha = a_1 \cdot N + b_1$, testing_utterances);
if($F^k > F^{k-1}$)
    Repeat
        {$b_1 += \Delta b_1$; $k++$;
         $F^k =$ speech_recognition($\alpha = a_1 \cdot N + b_1$, testing_utterances);
        } while ($F^k > F^{k-1}$);
else
    Repeat
        {$b_1 -= \Delta b_1$; $k++$;
         $F^k =$ speech_recognition($\alpha = a_1 \cdot N + b_1$, testing_utterances);
        } while($F^k > F^{k-1}$);
return $F^k$;
```

Step 3: Estimate the parameters a_3 and b_3 under the condition $N > N_3$, wherein $M_1(N) = M_2(N) = 0$, $M_3(N) = 1$, and

$$\alpha = \frac{M_3(N) f_3(N)}{M_3(N)} = f_3(N) = a_3 \cdot N + b_3.$$

The determination of a_3 and b_3 is done by the same process for a_1 and b_1.

Step 4: Estimate the parameters a_2 and b_2 under the condition $N_1 < N < N_2$, wherein $M_1(N) = (N_2 - N)/(N_2 - N_1)$, $M_2(N) = (N - N_1)/(N_2 - N_1)$, $M_3(N) = 0$, and

$$\begin{aligned}\alpha &= \frac{M_1(N) f_1(N) + M_2(N) f_2(N)}{M_1(N) + M_2(N)} \\ &= \frac{((N_2 - N)/(N_2 - N_1))(a_1 \cdot N + b_1) + ((N - N_1)/(N_2 - N_1))(a_2 \cdot N + b_2)}{(N_2 - N)/(N_2 - N_1) + (N - N_1)(N_2 - N_1)} \\ &= \frac{(N_2 - N)(a_1 \cdot N + b_1) + (N - N_1)(a_2 \cdot N + b_2)}{N_2 - N_1}.\end{aligned}$$

With a_1 and b_1 already obtained at step 2, the parameters a_2 and b_2 are determined through the tuning for best recognition rate process too.

Step 5: Re-estimate the parameter N_3 under the condition $N_2 < N < N_3$, wherein $M_1(N) = 0$, $M_2(N) = (N_3 - N)/(N_3 - N_2)$, $M_3(N) = (N - N_2)/(N_3 - N_2)$, and

$$\begin{aligned}\alpha &= \frac{M_2(N)f_2(N) + M_3(N)f_3(N)}{M_2(N) + M_3(N)}\\ &= \frac{((N_3 - N)/(N_3 - N_2))(a_2 \cdot N + b_2) + ((N - N_2)/(N_3 - N_2))(a_3 \cdot N + b_3)}{(N_3 - N)/(N_3 - N_2) + (N - N_2)/(N_3 - N_2)}\\ &= \frac{(N_3 - N)(a_2 \cdot N + b_2) + (N - N_2)(a_3 \cdot N + b_3)}{N_3 - N_2}.\end{aligned}$$

With a_2 and b_2 together with a_3 and b_3 already obtained at steps 4 and 3, respectively, a new value for N_3 can be found through the same process too.

Step 6: Update N_1 and N_2 such that $N_1 : N_2 : N_3 = 1 : 2 : 3$. Repeat from step 2 until the settings of a_1, a_2, a_3, b_1, b_2, b_3, N_1, N_2, and N_3 maximize the system performance over the training data set (10 adaptation utterances from each of the 15 subjects and 60 more from each of them for testing purposes).

2.4. Time complexity analysis

The computation time required by Eq. (8) involves two parts, the one for α computation and the other for W_s. Computing W_s by using MLLR is much less expensive than by using MAPLR as already mentioned.

The overhead of finding α in terms of the number of multiplications, as compared to standard MLLR, can be analyzed through its computation defined by Eq. (10). For $N < N_1$, $\alpha = a_1 \cdot N + b_1$ which requires 1 multiplication, as is for the case when $N > N_3$, $\alpha = a_3 \cdot N + b_3$.

For $N_1 < N < N_2$,

$$\begin{aligned}\alpha &= \frac{M_1(N)f_1(N) + M_2(N)f_2(N)}{M_1(N) + M_2(N)}\\ &= \frac{N^2(a_2 - a_1) + N(a_1N_2 - a_2N_1 + b_2 - b_1) + b_1N_2 - b_2N_1}{N_2 - N_1} = p(c_1N^2 + c_2N + c_3),\end{aligned}$$

the computation of which involves 4 multiplications, as is for the case when $N_2 < N < N_3$,

$$\begin{aligned}\alpha &= \frac{M_2(N)f_2(N) + M_3(N)f_3(N)}{M_2(N) + M_3(N)}\\ &= \frac{N^2(a_3 - a_2) + N(a_2N_3 - a_3N_2 + b_3 - b_2) + b_2N_3 - b_3N_2}{N_3 - N_2} = q(d_1N^2 + d_2N + d_3).\end{aligned}$$

Thus the computation of Eq. (8) is of the same order as computing Eq. (3), given that W_s is estimated by MLLR.

2.5. FLC-MLLR and fuzzy-MLLR

While T–S fuzzy control mechanism was applied in determining the weight α for incremental MLLR adaptation in this work, other fuzzy techniques were also used for MLLR adaptation. For instance, in order to solve the problem of component clustering such that the components clustered into a certain base class can be transformed by an associated regression matrix in the context of multiple regression classes, Gales proposed a fuzzy clustering scheme [19] for determining the weight γ_p in the adaptation below:

$$\hat{\mu}_s = \left[\sum_{p=1}^{P} \gamma_p W_s^{(p)}\right] \xi_s, \tag{11}$$

where γ_p represents the degree of how much ξ_s belongs to regression class p. As a result, in the case that transformation matrices are few, the adaptation for those base classes without a corresponding regression matrix can still be carried out with more reliability guaranteed. The role of γ_p in Gales' work is thus quite different from that of α herein. Interestingly, Eq. (11) could be extended as

$$\widetilde{\mu}_s = \alpha\mu_s + (1 - \alpha)\left[\sum_{p=1}^{P} \gamma_p W_s^{(p)}\right] \xi_s, \tag{12}$$

which would reduce to the form of Eq. (8) in the context of one regression class adaptation, i.e. $p = 1$, as is the case considered in this paper.

3. Experiments and results

In Mandarin, each syllable consists of a beginning initial part and an ending final part. The modeling of Mandarin syllables assumes that the initial part is right dependent on the beginning phone of the following final part and the final part is context independent [27]. A Mandarin utterance may contain one to several syllables. The experiments involve: (1) the establishment of initial SI models, (2) the training phase for fixing hyperparameters of the FLC, and (3) the recognition phase for performance evaluation on the tuning of α weight by FLC.

3.1. Database and experiment design

The initial SI models are trained for a set of HMM parameters using the database MAT400 sub-database DB3 [26], which consists of 4800 utterances from native Mandarin speakers. The HMM of a syllable consists of HMM of 3 states for the initial part and HMM of 6 states for the final part, and HMM of an utterance includes HMMs of the constituent syllables. Together there are 440 states, each of which has 4 Gaussian distributions, in the SI models.

In the training phase, the training data used for tuning the hyperparameters of the FLC as described in Section 2.3 were collected from 15 speakers. From each of the 15 speakers, 10 utterances of city names (picked among 30 cities) were requested as adaptation data, and then 60 utterances for all cities (two utterances for each) as testing data, which were recorded by a close-talking microphone. Note that speech signals were sampled at 8 kHz. The analysis frames were 30-ms wide with a 20-ms overlap. For each frame, a 24-dimensional feature vector was extracted. The feature vector for each frame consisted of a 12-dimensional (12-D) mel-cepstral vector and a 12-D delta-mel-cepstral vector. The training phase experiment procedure is explained in the pseudo-code sequence below for readability and clearness:

```
$\overline{F}^0$ = baseline recognition rate; $t = 0$;
*Repeat*
{ $t++$;
 $\overline{F}^t_2$ = 2_utterances_training (SI_models, hyperparameters);
 $\overline{F}^t_4$ = 4_utterances_training (SI_models, hyperparameters);
 $\overline{F}^t_6$ = 6_utterances_training (SI_models, hyperparameters);
 $\overline{F}^t_8$ = 8_utterances_training (SI_models, hyperparameters);
 $\overline{F}^t_{10}$ = 10_utterances_training (SI_models, hyperparameters);

     $\overline{F}^t = \frac{\sum_{i=1}^{5}\overline{F}^t_{2\cdot i}}{5}$;

     $\Delta\overline{F}^t = |\overline{F}^t - \overline{F}^{t-1}|$;

} *until* $\Delta\overline{F}^t <$ threshold;
```

where $2 \cdot i$_utterances_training$(\cdot)$, $i = 1, 2, 3, 4, 5$, is the procedure using $2 \cdot i$ adaptation utterances from 15 speakers for fixing the nine hyperparameters of FLC defined in Section 2.3 and thus returning a somewhat satisfactory overall recognition rate $\overline{F}_{2\cdot i}{}^t$ for the 15 training speakers as explained in the code-like sequence below:

```
$2 \cdot i$_utterances_training (SI_models, hyperparameters) // $i = 1, 2, 3, 4, 5$.
{$k = 0$;
$\overline{F}^0_{2\cdot i}$ = baseline recognition rate;
*Repeat*
    {$k++$;
    $\overline{F}^k_{(2\cdot i)1}$ = speaker_training (SI_models, test_data$_1$, hyperparameters,
        $2 \cdot i$_utterances$_1$);
        ⋮
    $\overline{F}^k_{(2\cdot i)j}$ = speaker_training (SI_models, test_data$_j$,hyperparameters,
        $2 \cdot i$_utterances$_j$);
        ⋮
```

```
  $\overline{F}^k_{(2\cdot i)15}$ = speaker_training (SI_models, test_data$_{15}$,hyperparameters,
  $2 \cdot i$_utterances$_{15}$);

          $\overline{F}^k_{2\cdot i} = \frac{\sum_{j=1}^{15} \overline{F}^k_{(2\cdot i)j}}{15}$;

          $\Delta\overline{F}_{2\cdot i} = |\overline{F}^k_{2\cdot i} - \overline{F}^{k-1}_{2\cdot i}|$;

  } until $\Delta\overline{F}_{2\cdot i}$ < threshold 1;
 return $\overline{F}^k_{2\cdot i}$;
 };
```

where $2\cdot i$_utterances$_j$ and test_data$_j$ denote the adaptation utterances in the number of 2, 4, 6, 8, and 10 and the 60 test utterances from the jth speaker, $1 \leqslant j \leqslant 15$. And speaker_training$(\cdot)$ is the procedure that would incrementally adapt the SI models by appropriate settings of the hyperparameters of the T–S FLC, as already described in Section 2.3, such that the adaptation would not jeopardize the recognition rate, given $2 \cdot i$ utterances.

```
 speaker_training (SI_models, test_data$_j$, hyperparameters, $2 \cdot i$_utterances$_j$)
 // $j = 1, \ldots, 15$.
 {
          Estimation_of_$W_s$($2 \cdot i$_utterances$_j$);
          $\overline{F}_{(2\cdot i)j}$ = Iterative_process (SI_models, test_data$_j$, $W_s$, hyperparameters);
          // as described in Section 2.3 for maximizing the recognition rate $\overline{F}_{(2\cdot i)j}$.
          return $\overline{F}_{(2\cdot i)j}$;
 };
```

In the recognition phase, a group of 15 speakers that are entirely different from the previous group were recruited and again each were requested 10 and 60 utterances for adaptation and recognition, respectively. The weight α is calculated by using the hyperparameters acquired in the training stage for adaptation. For comparison, full transformation matrices were used for standard MLLR, MAPLR, and the proposed FLC-MLLR. Since the amount of adaptation data was very small, only one common regression matrix tying all states was used for MLLR to make the most efficient use of the data available for adaptation, and MAPLR and FLC-MLLR used one single regression matrix of their own too. The prior densities required by MAPLR were derived directly from the SI models alone. Five adapted models were constructed using 2, 4, 6, 8, and 10 adaptation utterances from each of the 15 speakers, each model being associated with an α weight. Then 60 utterances from each of the 15 speakers were fed into the five adapted models for recognition rate evaluation.

3.2. Experiment results

Several observations were made during experiment works and these are described here:

(1) It is observed that the weight α decreases as the number of adaptation utterances increases. As depicted in Fig. 2, α drops by a noticeable step when the number of utterances increases from two to four, and then declines gradually, somewhat stabilized, as the number of utterances increases.

(2) Recognition performance comparisons with various numbers of adaptation utterances were made among the proposed FLC-MLLR utilizing a T–S FLC, the conventional MLLR without exploiting prior knowledge of the initial SI model, and the MAPLR with the prior density derived directly from the initial SI model alone. As shown in Fig. 3, it is observed that FLC-MLLR is better than MAPLR and MLLR for all cases, especially when training data are quite limited. Note that the performance of MLLR falls below the baseline when two utterances were available for adaptation. All three methods demonstrate improved recognition rate and MAPLR tends to catch up FLC-MLLR when the amount of training data increases.

(3) Effects of full-range variation of α on the recognition performance of MLLR under extreme cases of training data availability are also observed, as shown in Figs. 4 and 5, respectively. The former shows that while the training data are scarce, two utterances say the performance would go below the baseline if, for α being less than 0.5, the model adaptation is to be largely determined by the transformation matrix W_s which is very much likely poorly estimated. With increasing α, the influence of W_s on the adaptation will be reduced and the recognition rate is improved as expected. However, when α goes beyond 0.5, further the performance degrades as the system in a sense ceases to adapt. On the other hand, when the training data are sufficient, 10 utterances, for instance, full advantage of adaptation by W_s should be exploited, by using a small α value, for good performance, as depicted in Fig. 5.

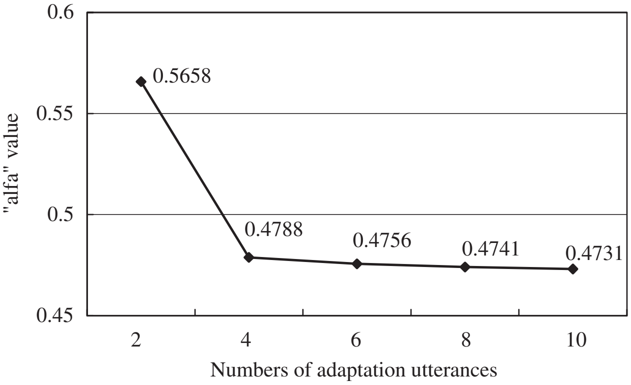

Fig. 2. The curve of the training values of α.

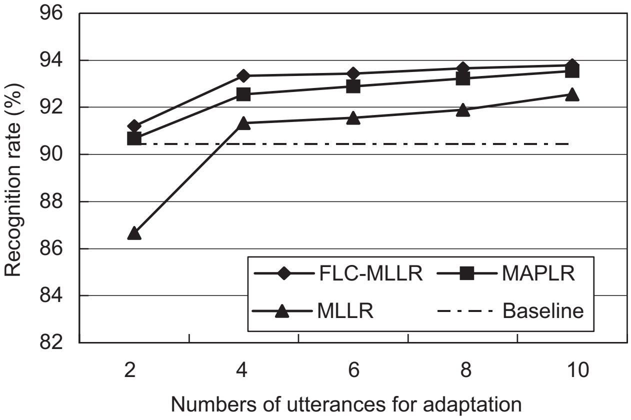

Fig. 3. The performance curves of FLC-MLLR, MAPLR, and conventional MLLR in the recognition testing experiments of the different amount of adaption data.

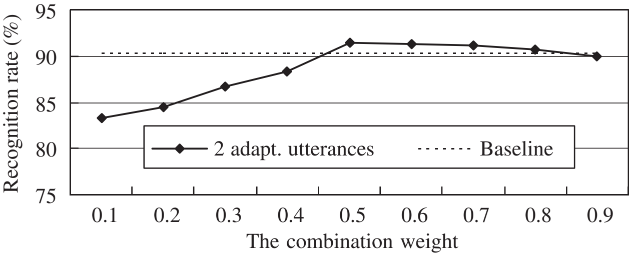

Fig. 4. Numbers of adaptation utterances =2.

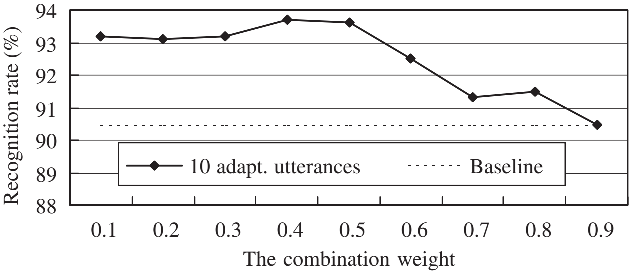

Fig. 5. Numbers of adaption utterances =10.

4. Conclusions

The proposed FLC-MLLR adaptation uses a T–S FLC for adjusting the weight α during model adaptation governed by

$$\widetilde{\mu}_s = \alpha\mu_s + (1-\alpha)W_s\xi_s.$$

The experiments showed that, via the T–S fuzzy mechanism, α is regulated in the way as expected: getting greater so as to compromise the incorrectness lurking in W_s due to insufficient adaptation data and getting smaller when W_s is trust-worthy. Thus, FLC-MLLR not only demonstrates the robustness against data scantiness when compared to conventional MLLR, but also executes at much lower computing cost than MAPLR, while attaining better recognition rate in all experiment conditions.

References

[1] J.L. Gauvain, C.H. Lee, Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, IEEE Trans. Speech Audio Process. 2 (2) (1994) 291–298.

[2] G. Zavagliogkos, R. Schwartz, J. Makhoul, Batch, incremental and instantaneous adaptation techniques for speech recognition, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1995, pp. 676–679.

[3] S.J. Cox, J.S. Bridle, Unsupervised speaker adaptation by probabilistic spectrum fitting, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1989, pp. 294–297.

[4] C.J. Leggetter, P.C. Woodland, Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models, Comput. Speech Lang. 9 (1995) 171–185.

[5] J.T. Chien, L.M. Lee, H.C. Wang, Estimation of channel bias for telephone speech recognition, in: Proceedings of the International Conference on Spoken Language Processing (ICSLP), vol. 3, 1996, pp. 1840–1843.

[6] J.T. Chien, H.C. Wang, Telephone speech recognition based on Bayesian adaptation of hidden Markov models, Speech Commun. 22 (1997) 369–384.

[7] C. Chesta, O. Siohan, C.H. Lee, Maximum a posteriori linear regression for hidden Markov model adaptation, in: Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH), 1999, pp. 211–214.

[8] W. Chou, Maximum a posteriori linear regression with elliptically symmetric matrix priors, in: Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH), 1999, pp. 1–4.

[9] A. Gunawardana, W. Byrne, Robust estimation for rapid speaker adaptation using discounted likelihood techniques, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2000, pp. 985–988.

[10] R. Kuhn, J.-C. Junqua, P. Nguyen, N. Niedzielski, Rapid speaker adaptation in eigenvoice space, IEEE Trans. Speech Audio Process. 8 (6) (2000) 695–707.

[11] K.T. Chen, W.W. Liau, H.M. Wang, L.S. Lee, Fast speaker adaptation using eigenspace-based maximum likelihood linear regression, in: Proceedings of the International Conference on Spoken Language Processing (ICSLP), vol. 3, 2000, pp. 742–745.

[12] K.T. Chen, H.M. Wang, Eigenspace-based maximum a posteriori linear regression for rapid speaker adaptation, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, 2001, pp. 917–920.

[13] B. Mak, S. Ho, J.T. Kwok, Speedup of kernel eigenvoice speaker adaptation by embedded kernel PCA, in: Proceedings of the International Conference on Spoken Language Processing (ICSLP), 2004, pp. 2913–2916.

[14] B. Mak, R. Hsiao, Improving eigenspace-based MLLR adaptation by kernel PCA, in: Proceedings of the International Conference on Spoken Language Processing (ICSLP), 2004, pp. 13–16.

[15] R. Hsiao, B. Mak, Kernel eigenspace-based MLLR adaptation using multiple regression classes, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, 2005, pp. 985–988.

[16] B. Mak, J.T. Kwok, S. Ho, Kernel eigenvoice speaker adaptation, IEEE Trans. Speech Audio Process. 13 (5) (2005) 984–992.

[17] B. Zhou, J. Hansen, Rapid discriminative acoustic model based on eigenspace mapping for fast speaker adaptation, IEEE Trans. Speech Audio Process. 13 (4) (2005) 554–564.

[18] B. Mak, S. Ho, Various reference speakers determination methods for embedded kernel eigenvoice speaker adaptation, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, 2005, pp. 981–984.

[19] M.J.F. Gales, The generation and use of regression class trees for MLLR adaptation, Technical Report CUED/F-INFENG/TR263, Cambridge University, 1996.

[20] R. Yager, D. Filev, Essentials of Fuzzy Modeling and Control, Wiley, New York, 1994.

[21] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst. Man Cybern. 15 (1985) 116–132.

[22] J. Yen, R. Langari, L.A. Zadeh (Eds.), Industrial Applications of Fuzzy Logic and Intelligent Systems, IEEE Press, New York, 1995.

[23] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. 39 (1977) 1–38.

[24] M.J.F. Gales, P.C. Woodland, Mean and variance adaptation within the MLLR framework, Comput. Speech Lang. 10 (1996) 249–264.

[25] O. Siohan, C. Chesta, C.-H. Lee, Hidden Markov model adaptation using maximum a posteriori linear regression, in: Proceedings of the Workshop on Robust Methods for Speech Recognition in Adverse Conditions, 1999, pp. 147–150.

[26] H.C. Wang, MAT—a project to collect Mandarin speech data through telephone networks in Taiwan, Comput. Linguist. Chin. Lang. Process. 2 (1997) 73–89.

[27] C.H. Lin, C.H. Wu, P.Y. Ting, H.M. Wang, Frameworks for recognition of Mandarin syllables with tones using sub-syllabic units, Speech Commun. 18 (2) (1996) 175–190.

About the Author—ING-JR DING was born in 1975. His research interests include speech recognition, speaker adaptation, pattern recognition, and artificial intelligence.