Fuzzy Logic Based Control for VFS Speaker adaptation

Ing-Jr Ding

Abstract

This work presents a fuzzy logic control (FLC) mechanism for the widely adopted vector field smoothing (VFS) speaker adaptation scheme. The proposed mechanism, called FLC-VFS, regulates the influence of VFS adaptation when the training data from a new speaker is inadequate, specifically by taking the amount of adaptation data into account at the stage of transfer vectors interpolation, and thus ensures the robustness of VFS adaptation against data scarcity. The proposed mechanism is conceptually simple and effective. Experimental results indicate that FLC-VFS outperforms conventional VFS, particularly when the adaptation data are scarce.

Keywords: speech recognition, speaker adaptation, Takagi-Sugeno fuzzy logic controller, vector field smoothing.

1. Introduction

Speaker adaptation is a process for transforming a speaker independent (SI) speech recognition system into a speaker dependent (SD) one, which is indispensable in the pursuit of the satisfactory recognition performance. What it does is basically adjusting acoustic parameters of the SI speech model, typically in the form of HMM, with speech samples acquired from a new/target speaker and results in a speaker-adapted (SA) speech model.

Techniques in the mainstream for speaker adaptation include MAP- and MLLR-adaptation [1][2]. Due to the inherent computational nature, MAP-adaptation adapts only those model parameters associated with the adaptation samples, whereas MLLR-adaptation adapts the entire or subclass of the model given the very scarce data. Vector field smoothing (VFS) adaptation is a post-processing after MAP-adaptation based on the idea that, in the model space, for those speech parameter vectors not altered during MAP-adaptation and yet lying in the vicinity of MAP adapted ones, an expectation of collateral adaptation in terms of near by vector adaptations would seem plausible [3][4]. As a consequence, it came out with the MAP-VFS adaptation which proved to be in general better than MAP alone given the same limited amount of adaptation data.

The quality of MAP-adaptation, MLLR alike, depends largely upon the number of utterances acquired from adapted speakers, i.e., insufficient or inadequate amounts of adaptation data would most likely lead to an unreliable speech model adaptation, which inevitably jeopardizes the recognition performance. VFS adaptation, as a complementary measure to the local-adaptation nature of MAP, shares the same weakness. Variants of MAP [5][6] and VFS [7], working under the key notion that the adaptation should not deviate too much from prior means in the speech model when adaptation data is limited, have been reported for tackling the issues caused by the scarcity of adaptation data, as will be explained in the next section.

The scheme for VFS adaptation proposed in this article works in the same line of thought by regulating the adaptation according to the amount of adaptation data under a fuzzy logic control (FLC) mechanism [8][9], as will be detailed in the third section.

By incorporating FLC-VFS with the conventional MAP, a composite of MAP-FLCVFS adaptation is put to the work in a series of experiments for the recognition of 100 worldwide renowned cities. The observations from the experiments show that the recognition rate of MAP-FLCVFS is better than that of MAP-VFS in a series of tests using various amounts of adaptation data ranging from the case of extremely limited (5 utterances) to abundant (100 utterances), as will be reported in Section 4. Some remarks are given in the final section to conclude this article.

2. MAP-VFS Adaptation

As aforementioned, MAP-adaptation adapts the portion of an SI speech model associated with the adaptation samples, whereas the rest remains intact. And VFS is more or less a patching-up measure based on the idea of “collateral adaptation” for propagating the effect of MAP-adaptation around the adapted spots.

A. MAP Adaptation

MAP-adaptation, by the nature of taking into account of the prior information during the maximum likelihood estimation, can be formulated as Bayesian learn process for adjusting HMM parameters. Assume that in the process only Gaussian mean vector adaptation is consid-
etered, the adaptation form of the $k$th mean vector in a GMM unit can be expressed in terms of the mean of adaptation samples and the prior mean as follows [1]

$$\hat{\mu}_k = \frac{N_k}{r + N_k} \bar{y}_k + \frac{r}{r + N_k} \mu_k,$$  

(1)

where $\mu_k$ is the mean vector of the model prior to adaptation, $\bar{y}_k$ indicates the mean vector of the adaptation samples and $N_k$ represents the number of the adaptation samples (frames). Note that the balance between $\bar{y}_k$ and $\mu_k$ for estimating $\hat{\mu}_k$ is biased by the setting of $r$, which could be exploited for enhancing the quality of adaptation against the scantiness of adaptation samples, as already been done in some variants of MAP [5][6]. As a rule of thumb, $\bar{y}_k$ is prone to being unreliable when $N_k$ is small, in which case $\hat{\mu}_k$ should stay as closer to $\mu_k$ as possible and hence a large $r$ is desirable; on the opposite, a small $r$ should be taken with a large $N_k$ so that $\bar{y}_k$ dominates the adaptation.

B. VFS Adaptation

The idea of “collateral adaptation” behind VFS adaptation is best illustrated by Figure 1, where the mean vector that isn’t adapted, $\mu_j$, has three MAP-adapted neighbors $\mu_1$, $\mu_2$ and $\mu_3$ in its vicinity of radius $R$ denoted by $N_k(j)$. Setting the collateral adaptation $v_j$ as a weighted sum of $k$-nearest adaptations, $v_k$’s, occurring around was adopted in the original VFS [3][4].

$$v_j = \frac{\sum_{k \in N_k(j)} \lambda_j, k \cdot v_k}{\sum_{k \in N_k(j)} \lambda_j, k},$$  

(2)

$$\lambda_j, k = \exp \left( -\frac{d_{j,k}}{f} \right),$$  

(3)

where $v_k$’s are referred to as the transfer vectors for $\mu_j$ and the weighting $\lambda_j, k$ was determined solely by the distance $d_{j,k} = |\mu_j - \mu_k|$ together with a tuning parameter $f$.

The $K$-nearest adapted neighbors $\mu_k$’s participating the estimation of $v_j$ can be selected purely geometrically by Euclidean distance $d_{j,k}$ in the mean vector space [3][4] or more elaborated by choosing from a cluster-structure in which acoustic relations among parameter vectors are established [5].

![Figure 1. Rationale behind VFS adaptation.](image-url)

Nevertheless, one key issue has been neglected in precedent VFS scheme [3][4] from the point of view regarding the quality of the transfer vectors $v_k$’s. Considering a specific mean vector $\mu_k$ adapted by very little adaptation samples and yet very close to $\mu_j$, in which case the associated $v_k$ very likely would be unreliable and accompanied by a significant weighting $\lambda_j, k$, and thus degrades the estimate of $v_k$, as can be readily seen in (2) and (3).

3. FLC-VFS Adaptation

The effect of VFS scheme in MAP-VFS adaptation can be further enhanced by adjusting the weighting $\lambda_j, k$ according to the quality of $v_k$ in the following way:

(1) When the transfer vector $v_k$ is reliable as a result of abundant adaptation samples (i.e., $N_k$ is large in MAP-adaptation), $\lambda_j, k$ should be large.

(2) When the quality of $v_k$ is in doubt as a result of a little adaptation samples (i.e., $N_k$ is small in MAP-adaptation), $\lambda_j, k$ should be lowered.

Referring to (2) and (3), the two requirements above can be fulfilled by the tuning of $f$ under the following rules.

Rule 1: If $N_k$ is small, then $f$ is to be small,
Rule 2: If $N_k$ is large, then $f$ is to be large.
For translating lingual statements into quantitative expressions for computation, fuzzy techniques [8] quite naturally come into use. Here a specific type of fuzzy logic control mechanism by Takagi-Sugeno (T-S hereafter) [9] is employed for reasons explained in the following.

A. T-S Fuzzy Control Mechanism
The framework of T-S fuzzy control includes a set of rules and the system output. For a system of \( n \) inputs governed by \( l \) rules, it would appear as follows.

Rule 1: IF \( x(1) \) is \( A_1^1 \) and ... and \( x(n) \) is \( A_n^1 \)
THEN \( y^1 = a_0^1 + a_1^1 x(1) + ... + a_n^1 x(n) \),

Rule 2: IF \( x(1) \) is \( A_1^2 \) and ... and \( x(n) \) is \( A_n^2 \)
THEN \( y^2 = a_0^2 + a_1^2 x(1) + ... + a_n^2 x(n) \),

Rule \( l \): IF \( x(1) \) is \( A_1^l \) and ... and \( x(n) \) is \( A_n^l \)
THEN \( y^l = a_0^l + a_1^l x(1) + ... + a_n^l x(n) \).

System output:
\[
y = \frac{\sum w^j y^j}{\sum w^j}, \quad \text{and} \quad w^j = \prod_{p=1}^{n} A_p^j(x(p)).
\]

Note that \( A_p^j(p=0,1,...,n) \) are fuzzy sets and \( A_p^j(x(n)) \) denotes the fuzzy values of the membership function associated with \( A_p^j \) for the input \( x(n) \);
\( a_p^j(p=0,1,...,n) \) are consequent parameters through which the \( i \)th consequence \( y^i \) is expressed as a linear combination of \( n \) inputs.

T-S fuzzy control model is conceptually simple in that overall system dynamics/behaviors, which are generally nonlinear and complicated, can be decomposed and formulated as a set of local and linear expressions.

B. FLC-VFS Formulation
For the specific problem in this work during VFS speaker adaptation, the aforementioned simple rule governing \( f \) regulation, given \( N_k \) adaptation samples observed for the \( k \)th Gaussian mean vector, can be formulated as the following implications

Rule 1: IF \( N_k \) is small, then \( f \) is small,
Rule 2: IF \( N_k \) is large, then \( f \) is large.

Let \( M_1(N_k) \) and \( M_2(N_k) \) be membership functions associated respectively with small and large amounts of adaptation data available, as shown in Figure 2. Also let functions \( g_1(N_k) \) and \( g_2(N_k) \) set small and large values of \( f \) respectively in each of the two cases. The previous set of rules can then be further clarified as:

Rule 1: If \( N_k \) is \( M_1(N_k) \), then \( f_1 = g_1(N_k) \),
Rule 2: If \( N_k \) is \( M_2(N_k) \), then \( f_2 = g_2(N_k) \),

where
\[
M_1(N_k) = \begin{cases} 1 & N_k < (N_k)_1, \\ (N_k)_2 - (N_k)_1 & (N_k)_1 \leq N_k \leq (N_k)_2, \\ 0 & N_k > (N_k)_2, \end{cases}
\]
\[
M_2(N_k) = \begin{cases} 0 & N_k < (N_k)_1, \\ (N_k)_2 - (N_k)_1 & (N_k)_1 \leq N_k \leq (N_k)_2, \\ 1 & N_k > (N_k)_2, \end{cases}
\]

along with the implication functions
\[
g_1(N_k) = a_1 \cdot N_k + b_1, \quad g_2(N_k) = a_2 \cdot N_k + b_2,
\]
and the final system output as follows [9]
\[
f = \frac{\sum_{j=1}^{2} M_j(N_k) \cdot g_j(N_k)}{\sum_{j=1}^{2} M_j(N_k)}.
\]

Figure 2. Membership functions.

Equation (4) shows that for \( N_k < (N_k)_1 \), \( f \) is solely determined by \( g_1(N_k) \), while for \( N_k > (N_k)_2 \), \( f \) is determined by \( g_2(N_k) \) alone. If \( N_k \) is between \((N_k)_1\) and \((N_k)_2\), then \( f \) denotes the weighted average of \( g_1(N_k) \) and \( g_2(N_k) \) with the weights \( M_1(N_k) \) and \( M_2(N_k) \).

The system now has six hyperparameters \((a_1, a_2, b_1, b_2, (N_k)_1, \text{and} (N_k)_2)\) to be fixed. The following it-
The iterative process is developed to set these hyperparameters.

**Step 1:** Let \((N_k)_1 : (N_k)_2 = 1:3\), and initialize \((N_k)_1\).

In this work, a dataset with fewer than 10 utterances, and a dataset with more than 30 adaptation utterances, are empirically regarded as SMALL and LARGE, respectively. As ten adaptation utterances take approximately 500 frames, the initiation starts with \((N_k)_1 = 500\) and \((N_k)_2 : (N_k)_2 = 1:3\).

**Step 2:** Estimate the parameters \(a_1\) and \(b_1\) under the condition \(N_k < (N_k)_2\), where

\[
M_1(N_k) = 1, \quad M_2(N_k) = 0, \quad \text{and} \quad f = \frac{M_1(N_k) \cdot g_1(N_k)}{M_1(N_k)} = g_1(N_k) = a_1 \cdot N_k + b_1.
\]

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

\[
a_1 = \text{initial value}; \quad b_1 = 0; \quad q = 0;
\]

/* The symbol \(q\) denotes the iterative index while fixing \(a_1\) and \(b_1\). */

\[
R^0 = \text{baseline_recognition_rate};
\]

/* The symbol \(\Delta a_1\) denotes an increment of \(a_1\). */

\[
R^q = \text{speech_recognition}(f = a_1 \cdot N_k + b_1, \text{testing_utterances}); \quad \text{while} \ (R^q > R^{q-1});
\]

if \((R^q > R^{q-1})\) /* Increasing \(a_1\) */

Repeat

\[
\{ a_1 = \Delta a_1; \quad q = q + 1; \}
\]

/* The function \(\text{speech_recognition()}\) is used to return the recognition performance of the proposed FLC-VFS adaptation with the parameter \(f\) controlled by selecting \(a_1\) and \(b_1\) for the testing data set \(\text{testing_utterances}\). */

else /* Decreasing \(a_1\) */

Repeat

\[
\{ a_1 = a_1; \quad q = q + 1; \}
\]

/* The symbol \(\Delta b_1\) denotes an increment of \(b_1\). */

\[
R^q = \text{speech_recognition}(f = a_1 \cdot N_k + b_1, \text{testing_utterances}); \quad \text{while} \ (R^q > R^{q-1});
\]

**Step 3:** Estimate the parameters \(a_2\) and \(b_2\) under the condition \(N_k > (N_k)_2\), where

\[
M_1(N_k) = 0, \quad M_2(N_k) = 1, \quad \text{and} \quad f = \frac{M_2(N_k) \cdot g_2(N_k)}{M_2(N_k)} = g_2(N_k) = a_2 \cdot N_k + b_2.
\]

The values of \(a_2\) and \(b_2\) are fixed using the same process as for \(a_1\) and \(b_1\) with the initial condition \(R^0 = R^q\) from step 2.

**Step 4:** Re-estimate the parameter \((N_k)_2\) under the condition \((N_k)_1 \leq N_k \leq (N_k)_2\), where

\[
M_1(N_k) = \frac{(N_k)_2 - N_k}{(N_k)_2 - (N_k)_1}, \quad M_2(N_k) = \frac{N_k - (N_k)_1}{(N_k)_2 - (N_k)_1},
\]

and

\[
f = \frac{M_2(N_k) \cdot g_2(N_k) + M_2(N_k) \cdot g_2(N_k)}{M_1(N_k) + M_2(N_k)}
\]

Since \(a_1\) and \(b_1\), together with \(a_2\) and \(b_2\), have already been determined at steps 2 and 3 respectively, a new value for \((N_k)_2\) can now be obtained by tuning for a higher \(R^q\) value than in step 3.

**Step 5:** Update \((N_k)_1\) such that \((N_k)_1 : (N_k)_2 = 1:3\),

\[
\delta = \frac{|R^q - R^0|}{R^q}, \quad \text{/*} \quad R^0 = R^q.
\]

Repeat from step 2 until \(\delta\) is less than a predefined threshold.

Note that while fixing \(a_1\) and \(b_1\) in step 2, the process is designed in such way that if a better recognition rate can be attained by increasing \(a_1\), then \(a_1\) will keep increasing until the recognition rate reaches a local peak, otherwise \(a_1\) will keep decreasing until a local peak of the recognition rate is reached. Thus \(a_1\) can only be increasing or decreasing monotonically in step 2, allowing no chance of oscillation; \(b_1\) is treated in the
same way afterward. Likewise, \( a_2 \) and \( b_2 \) in step 3 are taken care of.

Compared to conventional VFS, the computation overhead of FLC-VFS adaptation for calculating \( f \) is practically minor, considering that at most 4 extra multiplications are required. The analysis is straightforward: For \( N_k < (N_k)_1 \), \( f = a_1 \cdot N_k + b_1 \) which takes only 1 multiplication, as if for the case when \( N_k > (N_k)_1 \), \( f = a_2 \cdot N_k + b_2 \), and for the case \( (N_k)_1 \leq N_k \leq (N_k)_2 \),

\[
f = \frac{M_1(N_k) \cdot g_1(N_k) + M_2(N_k) \cdot g_2(N_k)}{M_1(N_k) + M_2(N_k)} = p \cdot (c_1 \cdot N_k^2 + c_2 \cdot N_k + c_3),
\]

which involves 4 multiplications.

### 3. Experiments and Results

The experiments concern the recognition of 100 worldwide renowned city names in Mandarin and were run in three parts: (1) establishing the initial SI models, (2) the training phase for fixing FLC hyperparameters, and (3) the recognition phase, to evaluate the performance of the proposed FLC-VFS.

#### A. Databases and Speech Models Design

The MAT-2000 database [10] collected Mandarin utterances from 2000 native Mandarin speakers in Taiwan, and was used to setup the initial SI models as a set of HMM parameters. A Mandarin utterance consists of one to several syllables, and a Mandarin syllable in turn has two sub-syllabic units, the initial and final unit [11]. Some syllable goes without the initial unit. The HMM of a syllable comprises an HMM of 3 states for the initial unit and an HMM of 6 states for the final unit. The HMM of an utterance is the set of all HMMs of the constituent syllables. The SI models established for the work have a total of 440 states, each being characterized by a Gaussian mixture of 4 Gaussian distributions.

#### B. Models Adaptation

The training data were collected from 30 speakers in the training phase, where each of the 30 speakers was asked to offer one utterance for each of the 100 cities as the adaptation data, and another two utterances for each city to be used in following-up observations. Specifically, from the 3000 adaptation utterances were taken 5, 10, 20, 30 and 100 utterances for adapting the SI models through conventional MAP-VFS adaptation to acquire 5 SA models respectively. In all 5 MAP-VFS adaptations, the settings \( \tau = 30 \) for MAP, together with \( K = 10 \) and \( f = 20 \) for VFS were taken. Each of the 5 SA models were then fed with 200 utterances under various \( f \) settings ranging from 5 to 50 at a step of 5 and the recognition performances were recorded as shown in Table 1, from which it can be seen that the best choice of \( f \) for VFS-adaptation would be 20 in all cases.

The same 5 lots of adaptation data were used for adapting the SI models through the proposed MAP-FLCVFS adaptation to acquire 5 SAFLC models respectively, and again \( \tau = 30 \) together with \( K = 10 \) is taken, leaving \( f \) to be determined by the FLC mechanism during the process.

As a supplement, all utterances were recorded using a close-talking microphone, and the speech signals were sampled at 8 KHz. The analysis frames were 30 ms wide with a 15 ms overlap. A 24-dimensional feature vector consisting of 12 mel-cepstral and 12 delta-mel-cepstral components was extracted for each frame.

<table>
<thead>
<tr>
<th>Numbers of utterances for adaptation</th>
<th>Average recognition rates (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>93.2 92.3 93.6 95.3 96.3 97</td>
</tr>
<tr>
<td>10</td>
<td>93.2 92.2 93.7 95.6 96.8 97.6</td>
</tr>
<tr>
<td>15</td>
<td>93.2 92.2 94.1 96.1 97.3 98</td>
</tr>
<tr>
<td>20</td>
<td>93.2 92.7 94.8 96.6 97.9 98.5</td>
</tr>
<tr>
<td>25</td>
<td>93.2 92.3 94.2 96.3 97.4 98.3</td>
</tr>
<tr>
<td>30</td>
<td>93.2 92 93.8 95.8 96.9 98.3</td>
</tr>
<tr>
<td>35</td>
<td>93.2 91.7 93.6 95.5 96.5 98.4</td>
</tr>
<tr>
<td>40</td>
<td>93.2 91.9 93.8 95.5 96.6 98.2</td>
</tr>
<tr>
<td>45</td>
<td>93.2 91.8 93.6 95.2 96.2 98.4</td>
</tr>
<tr>
<td>50</td>
<td>93.2 91.6 93.6 95   96.8 98.5</td>
</tr>
</tbody>
</table>

#### C. Recognition Performance

In the recognition phase, a new group of 30 speakers was recruited, each again being asked for one utterance for each city to be used for MAP adaptation alone (\( \tau = 30 \) again) and 5 SAMAP models were built with 5, 10, 20, 30 and 100 adaptation utterances respectively. Two more utterances for each city were requested from each of the 30 subjects as the testing data for comparing the recognition performance by the three adaptation schemes:

- MAP with 5 SAMAP models
- MAP-VFS with 5 SA models built in the training phase
- MAP-FLCVFS with 5 SAFLC models built in the training phase

The records were shown in Figure 3, from which several observations are readily made:

1. The recognition rate gets improved as the number of adaptation utterances increases, which is true for all three adaptations,
2. In the case of limited adaptation utterances, the performance of MAP and MAP-VFS adaptation fall below the baseline recognition rate of 93.2 by using
the SI models, which is an indication exposing the potential incorrectness or unreliability lurking in the inadequately adapted SA_{MAP} and SA models due to insufficient adaptation samples.

In all testing cases, it is noted that the proposed MAP-FLCVFS adaptation leads to the best recognition, followed by MAP-VFS adaptation and then by MAP-adaptation, which indicating the effect of VFS by propagating the adaptation in SA_{MAP} through the estimate of collateral adaptation, as well as the effect of the FLC-VFS that takes into account of \( N_k \), by which the quality of collateral adaptation is ensured to push the performance one more step forward.

![Figure 3. Average recognition rates of MAP, MAP-VFS and MAP-FLCVFS with \( f = 20 \) for VFS and \( \tau = 30 \) for MAP.](image)

4. Conclusions

This work presents an FLC-VFS speaker adaptation with a weight control parameter \( f \) determined by the fuzzy logic controller. The fuzzy mechanism regulates \( f \) according to the amount of adaptation data. The performance evaluation demonstrated that the proposed FLC-VFS adaptation in MAP-FLCVFS composite performs significantly better than the conventional VFS adaptation with a fixed \( f \). The FLC-VFS adaptation is an adaptive learning method that is more robust against data insufficiency than the conventional VFS adaptation, with only a small additional cost in computation, 4 multiplications at most.

References


