國立交通大學

資訊科學與工程研究所

博士論文

模糊邏輯控制於語者調適及
音訊事件偵測之參數調適

On the Use of Fuzzy Logic Control in Adaptive Parameter Tuning for Speaker Adaptation and Audio Event Detection

研究生：丁英智

指導教授：林正中　教授

中華民國九十七年十二月

模糊邏輯控制於語者調適及音訊事件偵測之參數調適

On the Use of Fuzzy Logic Control in Adaptive Parameter Tuning for Speaker Adaptation and Audio Event Detection

研究生：丁英智　　Student：Ing-Jr Ding

指導教授：林正中　　Advisor：Cheng-Chung Lin

國立交通大學

資訊科學與工程研究所

博士論文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science and Engineering

December 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年十二月

模糊邏輯控制於語者調適及音訊事件偵測之參數調適

學生：丁英智　　　　　　　　　　　　指導教授：林正中 博士

國立交通大學 資訊科學與工程研究所

摘　　要

本篇論文在語者調適(speaker adaptation, SA)領域及音訊事件偵測(audio event detection)領域中導入了模糊邏輯控制(fuzzy logic control, FLC)機制以強化調適品質，從而改善自動語音辨識(automatic speech recognition, ASR)系統及音訊事件辨認(audio event recognition)系統的辨識性能。個人提出了數個結合模糊邏輯控制器的方法以有效地掌控辨識系統中的不定參數，進而使系統在處於極為不利的辨識情況時仍能保持令人滿意之辨識結果。對於語者調適領域，個人針對兩個廣為流傳的調適技術範疇：貝氏調適(Bayesian-based)及轉換調適(transformation-based)置入 FLC 調控機制。最大後機率(maximum a *posteriori*, MAP)估測調適是一種貝氏調適的典型方法。根據 MAP 方法，個人提出了結合一適當的模糊控制器的 FCMAP 方法。所發展之 FCMAP 可以藉由所設計的模糊控制器依據調適語料量之多寡有效地糾正隱藏式馬可夫模型(hidden Markov model, HMM)參數。然而，MAP 僅針對調適語料所涉及的 HMM 參數進行調適的改善，對於絕大部份沒有調適語料的 HMM 參數並無法提供有效的助益；FCMAP 亦承繼此一弱點。由於向量場平滑化方法(vector field smoothing, VFS)可以填補 MAP 方法的此項弱點，因此，個人延續 FCMAP 的設計概念而提出了在 VFS 調適程序中整合一個模糊邏輯控制器的 FLC-VFS 調適方法以對較多無調適語料的 HMM 參數在調適上提供有效的改善。目前廣為使用之最大可能性線性迴歸(maximum likelihood linear regression, MLLR)乃經典之轉換調適。以此 MLLR 方法做為基礎，個人提出了一

個 FLC-MLLR 調適方法以確保傳統 MLLR 在遭遇調適語料稀少時的強健性。FLC-MLLR 調適程序乃先建構一種像 MAP 方法的模型結合調適方式，而後再利用所設計的 FLC 依據調適語料量之多寡以決定需參考語者不相關(speaker independent, SI)模型之程度。再者，對於特定音訊事件的偵測，個人也提出了一個在 FLC 的架構之下實現可變動長度之決定視窗的辨識方法。實驗結果顯示在本篇論文中所提出各個整合 FLC 調控機制的方法之辨識精確度皆明顯優於傳統的方法。

On the Use of Fuzzy Logic Control in Adaptive Parameter Tuning for Speaker Adaptation and Audio Event Detection

student：Ing-Jr Ding　　　　Advisors：Dr. Cheng-Chung Lin

Institute of Computer Science and Engineering
National Chiao Tung University

ABSTRACT

In this dissertation, the exploitation of fuzzy logic control (FLC) mechanism in the fields of speaker adaptation (SA) and audio event detection is thoroughly investigated, specifically in the reliable determination of HMM acoustic parameters and in decision window regulation for enhancing the system recognition performance, given ordinary or adverse conditions in both training and operating stages.

For speaker adaptation against data scarcity, the author managed to engineer the FLC mechanism into the MAP and VFS estimate of HMM parameters for Bayesian-based adaptation; also into the MLLR estimate for transformation-based adaptation.

For the detection of singular audio event detection, the author developed an efficacious measure by varying the length of the decision window (DW) under the framework of FLC operation such that, depending on audio-tension in the context, the rate of decision making would adapt accordingly.

To the author's knowledge, the use of FLC mechanism in estimating HMM

acoustic parameters for speaker adaptation and audio event detection has been rarely attempted. Experiment results showed that the adaptation with the support of FLC do have several edges on those without.

(1) better performance in ordinary case,

(2) robustness against the scarcity of training data,

(3) less computation in parameter estimation as compared to other propositions on MLLR-enhancement.

And the detection with FLC support also demonstrates the capacity of self-adjustment in DW size depending on the context while achieving better recognition performance as compared to those running with fixed DW whose sizes are inappropriately selected.

Acknowledgement

I would like to express my sincere thanks to my advisor, Prof. Cheng-Chung Lin. Without his supervision and perspicacious advice, I can not complete this dissertation. Special thanks to my committee members, Prof. Hsiao-Chuan Wang, Prof. Hsin-Min Wang, Prof. Ching-Kuen Lee, and Prof. Berlin Chen for their valuable comments.

I also express my appreciation to all the faculty, staff and colleagues in the Department of Computer Science and Information Engineering, NCTU.

Finally, I am grateful to my family, my father, mother, brother and friends for their encouragement and support during these years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Intelligent human-machine interaction stresses the use of vocal and visual information as the communicating media such that the machines could interact with people just the same as people do with one another. For the visual part, the information process is afferent and the machine needs such device as CCD camera for taking pictures or video sequences as the visual input from which the configuration of the surroundings and even the status/situation reflected by the context have to be figured out such that the machine is claimed to be able to SEE. Computer vision is the discipline taking care of this portion of the job [1, 2].

However, the process of auditory information is both afferent and efferent. The machine would require the microphone and recorder as the input devices for collecting audio streams on which analysis is to be performed so that speech can be recognized, speaker can be identified and types of audio events can be categorized, etc; the machine also needs the synthesizer and speaker as the output devices for vocal reactions to the people. Interestingly enough, as a counterpart of computer vision, there is no such discipline as of “computer audition” in the realm of audio processing.

From the standpoint of signal processing, a comparison between the two media is listed in Table 1.1.

Table 1.1. A comparison between audio and visual media.

	audio	**visual**
nature	1D	2D or above
acquisition limitation	background noises	camera orientation-dependent lighting condition
acquisition devices	microphone audio-recorder	ccd camera image capturer
transmission bandwidth	low	high
regeneration	synthesizer/speaker	LCD display/printer
semantics	lingual or higher metaphor	geometrical or higher metaphor

Note that as far as the social interactions among people are concerned, the exchange of vocal information (i.e., hearing form and speaking to others) plays a far more important role than anything else, since the only way of instantly expressing one's and knowing other's desire or intension in an exact and precise manner without ambiguity is through the exercise of conversation. This is true for the kids in the preschool, as well as for various professionals in their serious careers.

The subjects in this dissertation concerns the computing techniques involved in the process of vocal information rather than visual one, specifically speaker adaptation schemes associated with MAP, VFS and MLLR, and audio event detection with variable decision window, which are to be briefed in the following sections.

1.1 Scope of the Dissertation

The researches pertaining to audio information processing covers a myriad of branches including, but not limited to, those shown in Fig. 1.1.

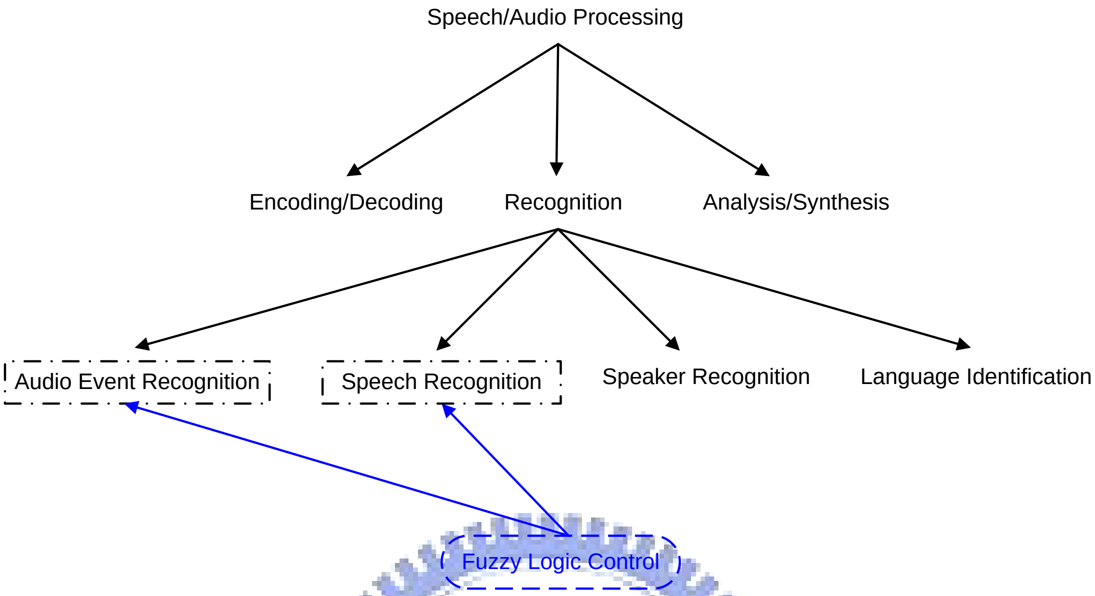


Fig. 1.1. Speech/Audio information processing represented by a myriad of branches.

In the area of speech recognition, the author proposed a framework of fuzzy mechanism that is applicable to some major speaker adaptation schemes for resolving the unreliable adaptation due to insufficient training samples; the implementation of which, a series of fuzzy logic controllers embedded in MAP, VFS and MLLR adaptations, has proven themselves by achieving far superior recognition rate at extreme adverse conditions.

The same framework is also applied to the detection of female screaming against different degree of background interferences, where the width of decision window scanning through the input audio stream was adjusted by a context-driven fuzzy logic controller.

1.2 Speaker Adaptation

Computing techniques for automatic speech recognition have existed for years [3] and, with the ever growing maturity, have found more and more applications in current daily life [4]. Nevertheless, the recognition performance of all speech recognition systems ever built is undeniably inferior to a human listener as already pointed out in [5].

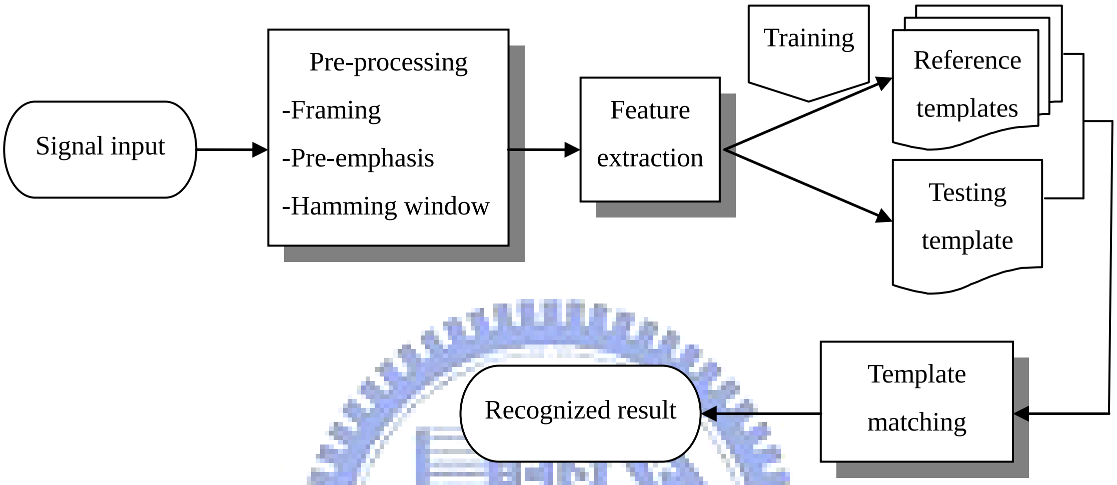


Fig. 1.2. The operating structure of a typical speech recognition system.

Fig. 1.2 depicts the operating structure of a typical speech recognition system for capturing specific short phrases or primitive statements only. Note that during the operation any disturbances causing a mismatch between the pre-established reference templates and the testing template would compromise the recognition performance and the sources of disturbances may include

- speech from speakers strange to the system
- speech from speaker known to the system, only in poor “vocal shape”
- various interferences in the background
- channel distortion induced in the acquisition process

and so forth.

Countermeasures can be taken in two aspects:

(1) Signal filtering and normalization are deployed so that the operating condition is in as much alignment with the referential condition as could be done.

(2) Internal tuning of the referential settings is undertaken so that the system adapts toward the actual operating environment when new speakers appear.

Techniques in the first category work at the level of signal processing, and are referred to as speech enhancement or feature-based adaptation through which noises adhered to the signals are removed to make the speech signals as clean and thus resemble to reference templates as possible. The cepstral mean normalization (CMN, or cepstral mean subtraction CMS) [6] and signal bias removal (SBR) [7] fall into this category too and are popular for their simplicity and effectiveness.

Approaches in the second category use sample utterances collected from the new speaker (the end-user of the system) for adapting the system internal parameter settings of the pre-established speech model. Consequently, they are referred to as model-based adaptation or speaker adaptation.

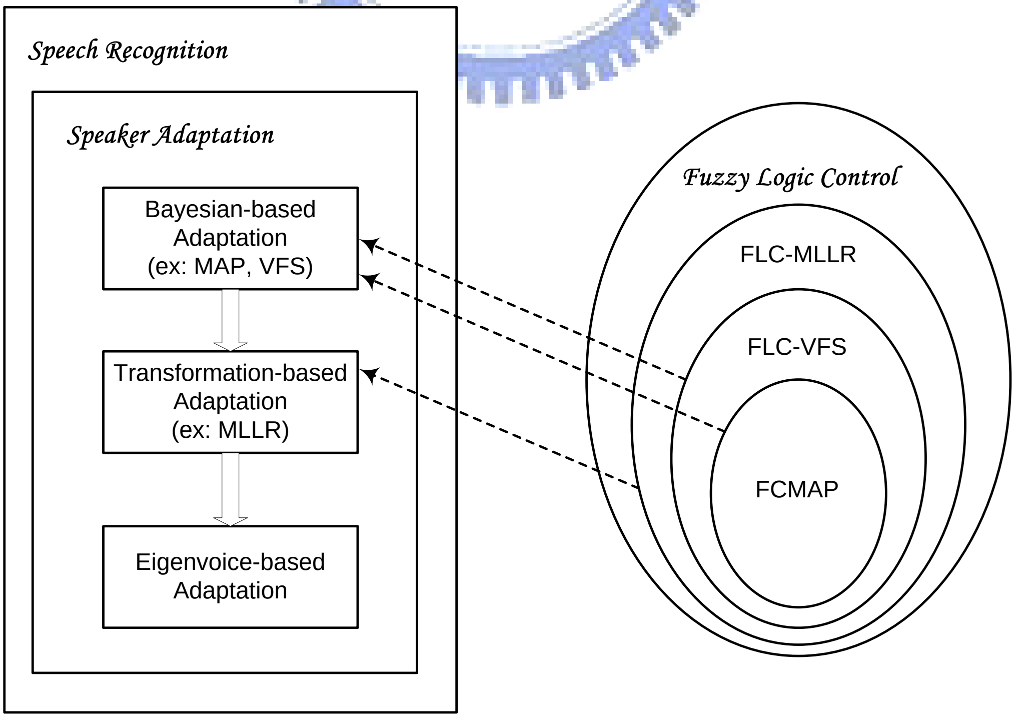

Speech Recognition
Speaker Adaptation
Bayesian-based Adaptation (ex: MAP, VFS)
Transformation-based Adaptation (ex: MLLR)
Eigenvoice-based Adaptation
Fuzzy Logic Control
FLC-MLLR
FLC-VFS
FCMAP

Fig. 1.3. Three categories of speaker adaptation techniques in speech recognition.

Fig. 1.3 reveals the chronological development of the three major speaker adaptation schemes. MAP adaptation, appearing around 1991 and the representative of Bayesian-based adaptation, works better than the ML (maximum likelihood) estimate of the adaptation by taking into account the information of prior means of the model. By the nature of MAP computation, in the speech model only the portions associated with the adaptation samples get updated, for which case VFS scheme came into the play as a supplement to MAP by extending the coverage of adaptation in the model space. The MAP-VFS adaptation in general offers more satisfaction in recognition performance than MAP alone given the same adaptation data. MLLR adaptation first appeared in 1995 and became the representative of transformation-based adaptation, where linear regression was employed to derive the transformation matrix using ML-estimate. Note that through the transformation by matrix multiplication, the entire model space is adapted at one time despite the fact that the sample utterances might convey very limited information for adaptation. In a sense, MLLR adaptation provides with an overall but somewhat coarser speech model adaptation, in contrast to MAP adaptation which brings about a local and yet specific effects of adaptation, given the same adaptation samples.

One thing that is common to both MAP and MLLR is that the quality of adaptation depends on the amount and adequacy of the adaptation samples: the more the samples, the better the adaptation quality which in turn determines the recognition performance. When the adaptation utterances from a new speaker are insufficient, the effects of either MAP or MLLR adaptation would be questionable: the recognition rate of which would fall below the baseline, i.e., worse than no adaptation at all as shown by the author’s experiments [8, 9].

Eigenvoice-based adaptation [10-20] is a relatively young member in the speaker adaptation family, first appearing around 2000, and is also known as speaker-clustering-based adaptation where a speaker dependent (SD) speech model is established for every member in a group of speakers, from which feature vectors called as eigenvoices are extracted through PCA for building the eigenvoice speech model. The adaptation to the speech model (an eigenvoice vector space) then can be undertaken when adaptation data is available, as shown in Fig. 1.4.

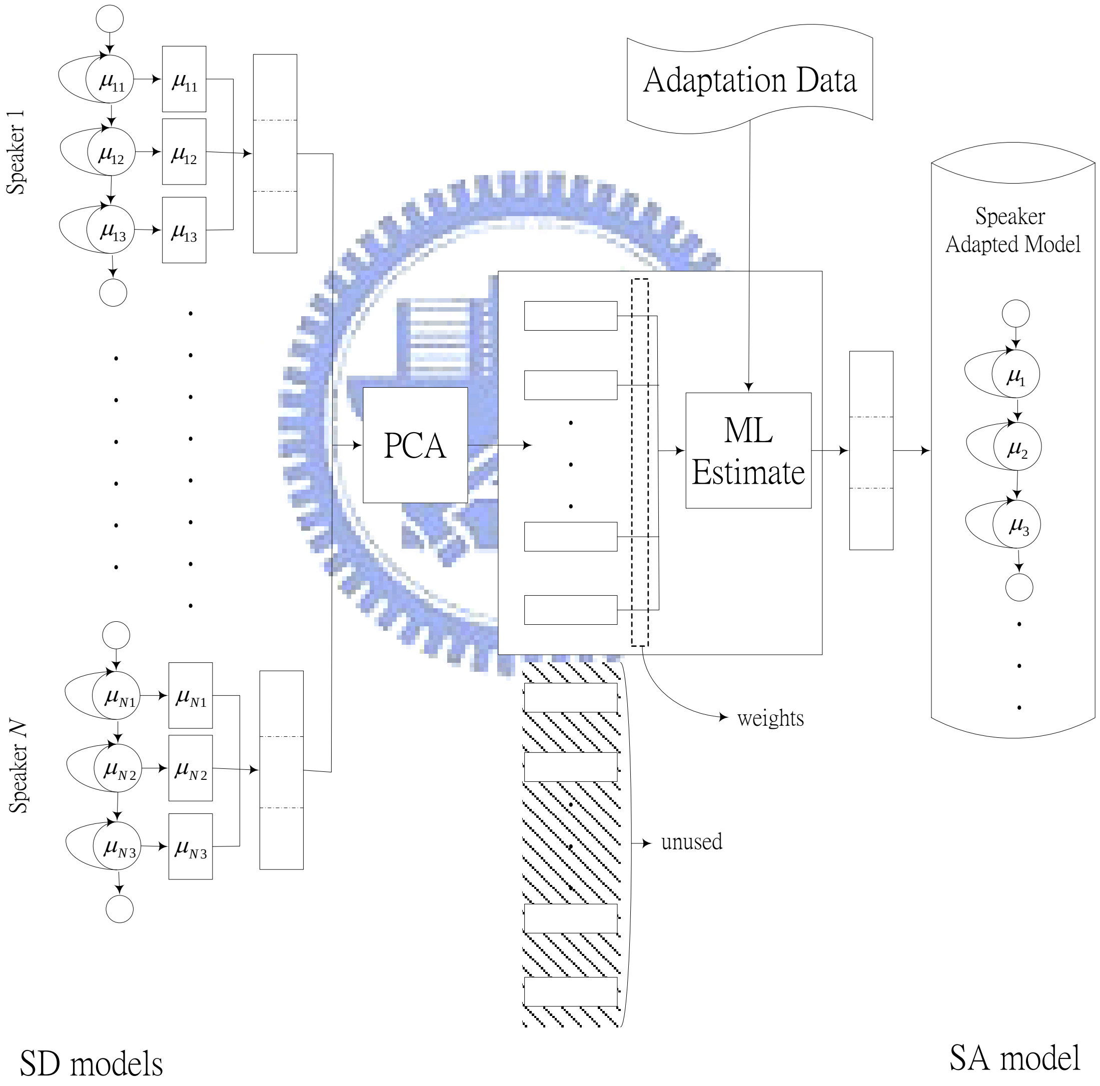


Fig. 1.4. Eigenvoice-based adaptation.

To summarize, speaker adaptation is a process that turn speaker-independent (SI) speech models into speaker-adapted (SA) ones, as is clearly seen in Fig. 1.5.

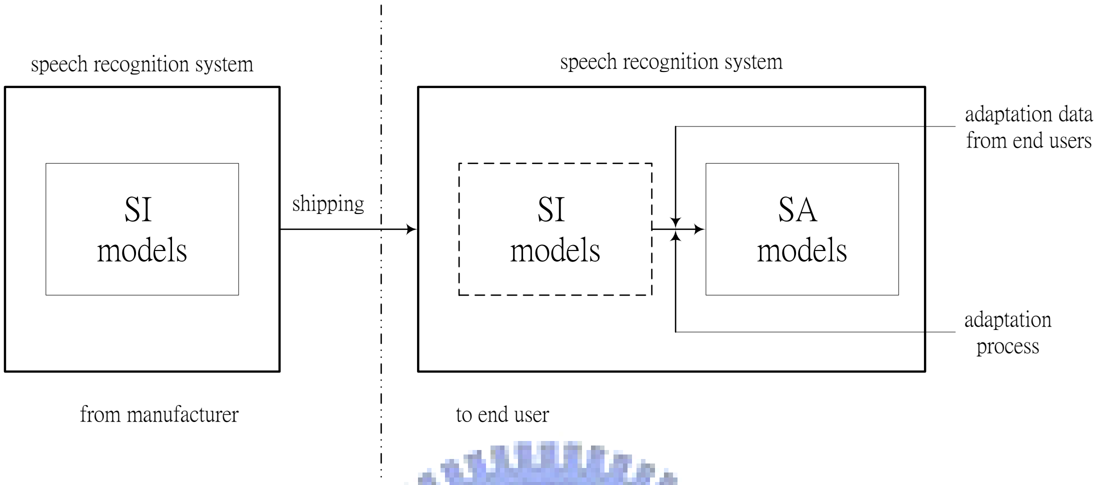


Fig. 1.5. Speaker adaptation scheme.

To ensure the quality of the adaptation at the scarcity of adaptation samples, the author proposes a general framework for enhancing MAP, VFS and MLLR adaptation, and the resultant implementations are named as FCMAP, FLC-VFS and FLC-MLLR respectively where FLC stands for fuzzy logic control, indicating the underlying fuzzy mechanism incorporated in the general system architecture.

1.3 Audio Event Detection

Conventional security, surveillance or remote homecare systems rely heavily, if not exclusively, on the visual information (i.e. data captured by video camera) for detecting specific events in considerations [21-24] through the use of motion tracking-analysis techniques. The similar development is also seen in the field of multimedia retrieval and indexing applications, where video information is the major concern and it is not until recently that audio cues are involved only as an auxiliary role for detecting certain specific shot in a video sequence [25-27]. Depending solely on visual data as the basis for capturing status/situation development in the context inevitably would be confronted by the limitations inherent in the image acquiring process:

- Video camera is an oriented-sighting device and views lying beyond the camera's visual angle are therefore "unseen".
- When the scene is in the darkness or over exposure, activities taking place wherein would become "unseen".
- The scenario like two gangsters threatening of killing each other right in front of the video camera, both with smiling on the faces, is in fact "unaware of" through "clearly seen".

Note that in all these circumstances, acoustic data can act as a complementary source of information for reflecting the auditory aspect of the reality in the context. A further thought in that almost all living creatures that move around in their habitats are equipped with organs for both visual and aural perception would remind us that any security, surveillance or remote homecare system dismissing the use of audio information is effectively a crippled one. And in fact species that can "hear" much better that they can "see" are more than one would have expected; scotopic animals, oceanic mammals and, of course, the moles are only a small group of examples of all. As a result, audio event detection has been getting a lot more attentions in recent years, and fundamental issues include

(1) Categorization of various kinds of sounds that are to be encountered in daily life, of which the sources may be

- artificial: gun shots [28], door opening/closing and glass breaking [29].
- human activities: coughing [30], voices under different emotions [31], crying, talking, walking and running [32], female screaming to be addressed in this

dissertation (Chap. 7).

- nature: wildlife activities [33] and ordinary or catastrophic phenomena [34].

Note that the entities of the categorization are not limited to the above three and in each category good and interesting subjects to be explored are virtually unlimited; "detecting a tiny mouse blowing wind one mile away", for instance, borrowing from the dialog in an old movie in the early 80's "Blue Thunder" is just one if the author is allowed.

(2) Internal representation and modeling of a designated type of sound, in order to be differentiated from other sounds and the background acoustics as were done in [32] and [28], where multi-level or hierarchical tree are utilized for more elaborated audio representation of several human activities and different types of gunshots, respectively.

(3) Representation and modeling of the background acoustics against which the compasison can be done for audio event detection, as was done in [35] for background noise analysis, and in [36] for online adaptation in background modeling where the idea of acoustic background modeling is translated from a precedent counterpart in video background modeling [37].

A typical audio event detection process starts with receiving a stream of audio frames coming in the system at regular time intervals, on which analysis is to be performed every time a fixed number of frames are collected (or equivalently an elapse of a pre-determined time span called decision window, DW) so as to decide if the designated audio event has occurred or not. The author proposes a variable-length decision window of which the window length is governed by a fuzzy mechanism for eliminating the deficiency suffered by the fixed-length DW approaches, as to be detailed in Chap. 7.

The rest of the dissertation is organized as follows. In Chap. 2, an overview of automatic speech recognition based on hidden Markov models for Mandarin is given, together with the mathematic backgrounds for the two speaker adaptation techniques in popular use: MAP-VFS composite and MLLR. Also described in Chap. 2 is audio event detection based on Gaussian mixture models. In Chap. 3, a general framework of fuzzy logic control is described, where the problem formulation by fuzzification, the establishment of fuzzy rule base and inference mechanism, and the defuzzification for final quantitative outputs are provided.

The main theme of this dissertation concerns the enhancement of extant speaker adaptation schemes by additional tuning according to the availability of adaptation data and of audio event detection by scanning the audio stream with a variable-sized decision window, both being govern by a general fuzzy mechanism; the formulation and implementations of which are explained respectively in chapters 4, 5, 6 and 7. And the concluding remarks of the research by the author are given in Chap. 8.

Chapter 2

Overview on Speech Recognition and Audio Event Detection

In the realm of man-machine interactions, audio processing no doubt receives far less attention than it deserves when compared to the resources/efforts invested in its counterpart of video processing. It has been so for over decades despite the fact that for thousands of years in human history instant and precise communications among individuals were mostly realized via the auditory channels: speak and listen (imagine the age before the creation of characters in ancient civilization).

Though the ability to understand what others are talking about is indispensable in social interactions, the auditory perceptual skill of differentiating one kind of sound from others that may be heard in one's living surroundings is far more important and crucial; for instance, being able to tell other's "HELLO" from the noises due to a vehicle's hard break, a gunshot from an explosion, a duck's quack from a goose's honk and the Spanish from the Italian without really understanding both languages etc. could be live-saving or at least useful or even amusing in one's daily life. Paradoxically enough, the development in audio process evolved in the opposite order: speech recognition was addressed far ahead of audio event detection which was not until recent years did it become visible on the stage.

Before the analysis on the audio information could commence, a pre-processing on the input audio signals is generally required for extracting acoustic features in preparation of any particular application under consideration, and in the case of this dissertation, speech recognition and audio event detection. As illustrated in Fig. 2.1, several major steps in the front-end processing is briefly explained as follows [38, 39]:

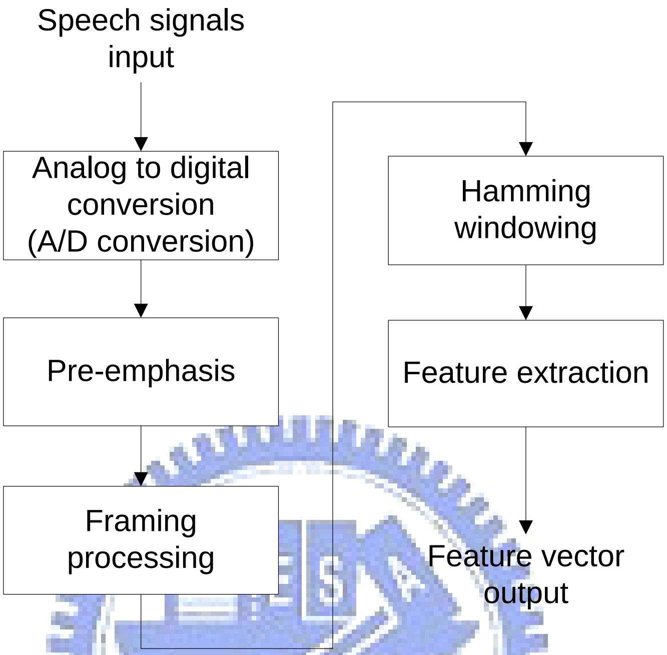


Fig. 2.1. The front-end processing procedure in preparation of subsequent audio analysis.

(1) A/D conversion:

The analog input data is converted into digital forms by sampling and A/D conversion.

(2) Pre-emphasis:

Components in the high-frequency band are enhanced.

(3) Framing:

Samples of audio data are divided into frames, each consisting of a pre-determined and same number of samples.

(4) Hamming windowing:

Discontinuity at the boundary of two consecutive frames is smoothed.

(5) Feature extraction:

From each frame, various parameters are extracted and a feature vector representing acoustic characteristics of the audio input in the associated time period is thus derived.

For the purpose of speech recognition and human voice related application, linear predictive coefficient (LPC) parameters, LPC cepstrum (LPCC) parameters and mel frequency cepstral coefficient (MFCC) parameters are the three most frequently seen in practice, and are employed in the author’s research.

In this chapter, theoretical backgrounds for two fundamental technical issues in speech recognition, namely HMM speech modeling and speaker adaptation, will be given; also given in the final are certain primary issues pertaining to audio event detection.

2.1 HMM Speech Modeling

The modeling of speech patterns can be implemented in the form of neural networks (NN, [40-42]), by using support vector machine (SVM, [43, 44]) or by using hidden Markov models (HMM) which to the author’s knowledge is by far the most popular and widely used one.

2.1.1 HMM and Mandarin Syllable Modeling

HMM is basically a stochastic process operating on an underlying Markov chain of a finite number of states and the same number of random functions: at any given instance of time, the process stays at a certain state and the random function associated with the current state determines what the next state will be. Such issues as how an HMM is to be cast into a model for certain specific applications and how the model parameters are to be estimated are addressed in [45-47] and in practice a state probability transition matrix is used to describe the probability of going from one state to the other states, which in effect defines the Markov chain at work. The applications of HMM to speech recognition can be found in many references. [48-51] are some of the examples. The work by C. H. Lin et al. [50] is particularly note worthy, where a framework for the recognition of syllables was established and later became a widely accepted standard in the modeling of Mandarin syllables with tones. According to which each Mandarin syllable consists of an initial part and a final ending part, each being called as a sub-syllable. The HMM 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. A Mandarin utterance may contain one to several syllables; the HMM of an utterance thus includes HMMs of the constituent syllables. In the actual implementation of the author's work, the HMM of a syllable consists of an HMM of 3 states for the initial part and an HMM of 6 states for the final part, and in total there are 440 states for all Mandarin sub-syllables. The HMM modeling of the initial sub-syllable in 3 states and the final sub-syllable in 6 states are respectively depicted in Fig. 2.2 and Fig. 2.3, where each circle represents a state and P_{ij} represents the probability density function concerning the transition from state *i* to state *j*. The HMM model employed in the author's research is referred to as left-to-right model since only left-to-right transitions are allowed; i.e. the transition from each state is limited to only two alternatives: either moving toward the right-hand side neighbor or staying at the current state.

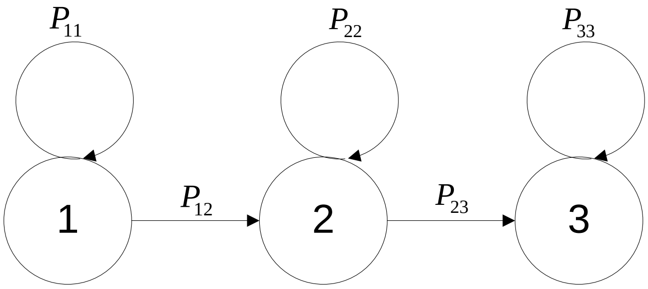


Fig. 2.2. 3-state HMM model for the initial sub-syllable.

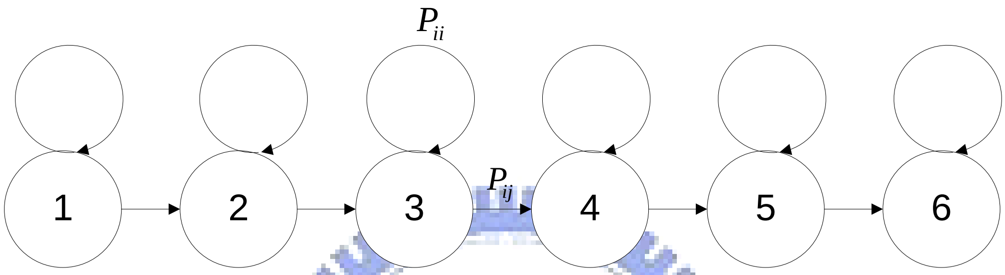


Fig. 2.3. 6-state HMM model for the final sub-syllable (i = 1 ~ 6).

2.1.2 Estimation and Decoding of HMM

Mathematically, a hidden Markov model can be represented by the parameter set $\lambda = (\pi, A, B)$. The underlying Markov chain of N states $S_1, S_2, ..., S_N$ can be specified by an initial state distribution vector $\pi = (\pi_1, \pi_2, ..., \pi_N)$ and a state transition probability matrix $A = \{a_{ij} \mid 1 \le i, j \le N\}$, in which π_i is the probability of S_i at time $t = 0$ and a_{ij} is the state transition probability of going from state S_i to state S_j. Moreover, if the observations composed of M discrete symbols $o_1, o_2, ..., o_M$ are considered, the finite set of probability distributions $B = \{b_j(q) \mid 1 \le j \le N, 1 \le q \le M\}$ with $b_j(q)$ being the probability of observing o_q given the state S_j, represents the random processes associated with the states. Usually, to characterize an HMM the decision of the number of states N and the number of observation symbols M also should be taken into account besides specifying the parameters π, A and B.

In order to acquire an efficient estimation of HMM model during the training phase and an optimal decoding procedure of the estimated HMM model during the recognition phase, three problems need to be taken care of [51]:

(1) If the observation sequence $O = \{o_1, o_2, ..., o_T\}$ is given, how the probability $p(O \mid \lambda)$ is to be evaluated then?

(2) If an HMM model and an observation sequence are known, how the optimal (the most likely) state sequence in the model that produces the observation is to be decide?

(3) If a model and a set of observations are given, how the model parameter set $\lambda = (\pi, A, B)$ to maximize $p(O \mid \lambda)$ is to be estimated?

For the first problem, some methods such as the forward recursive algorithm and backward recursive algorithm have been proven to be efficient [52]. For the third problem, the Baum-Welch method [45-47] is proposed to offer a local maximum solution although the computation for an explicit solution of the model λ is difficult.

For the second problem, the Viterbi algorithm proposed in [53] has been proven to be an effective one for acquiring an optimal state sequence. The score function $\delta_t(i)$ is defined as in Eq. (2-1), given the observation sequence $O = \{o_1, o_2, ..., o_T\}$

$$\delta_t(i) = \max_{s_1, s_2, ..., s_t} P(s_1, s_2, ..., s_t = S_i, o_1, o_2, ..., o_T \mid \lambda), \qquad (2\text{-}1)$$

where $\delta_t(i)$ has the largest probability at time t and at state S_i.

$\delta_{t+1}(i)$ is computed as follows using $\delta_t(i)$ by induction,

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_j(o_{t+1}). \qquad (2\text{-}2)$$

This iterative procedure is essentially a dynamic programming and the state sequence that has the maximum likelihood of generating the given observation sequence will be searched if one keep track of all the states which maximize Eq. (2-1). An array $\psi_t(j)$ is used to store the predecessor state of the state j at t. The steps of the Viterbi algorithm are as follows

(1) Initialization

$$\delta_1(i) = \pi_i b_i(o_1),\ \ 1 \le i \le N, \tag{2-3}$$

$$\psi_1(j) = 0,\ \ 1 \le j \le N, \tag{2-4}$$

(2) Recursion

$$\delta_t(j) = \max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}]b_j(o_t),\ \ 1 \le j \le N, \tag{2-5}$$

$$\psi_t(j) = \arg_{1\le i\le N}[\delta_{t-1}(i)a_{ij}],\ \ 1 \le j \le N, \tag{2-6}$$

(3) End

$$p^* = \max_{1\le i\le N}[\delta_T(i)], \tag{2-7}$$

$$s_T^* = \arg_{1\le i\le N}[\delta_T(i)], \tag{2-8}$$

(4) Back-tracing

$$s_t^* = \psi_t(s_{t+1}^*), t = T-1, T-2, ..., 1. \tag{2-9}$$

During the recursive step of this algorithm, the optimal sequence of states is obtained eventually.

2.2 Speaker Adaptation

Automatic speech recognition systems generally can be classified either as speaker-independent type (SI) or speaker-dependent type (SD), depending on how speech samples are colleted during system construction. An SI system typically collects speech samples from an 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. When a new speaker uses such an adaptive system, the parameters of the HMMs are updated by speech data obtained from this speaker. By speaker adaptation, the recognition performance can be significantly improved for outlier speakers such as non-native speakers or others not well represented in the SI training set.

Generally speaking, the operation for speaker adaptation can be carried out in either supervised mode or in unsupervised mode respectively, depending on if the transcription of the speaker-specific adaptation data has been known or not before performing the adaptation procedure [54]; the speaker adaptation is said to operate in batch mode if all adaptation data acquired from a new speaker is fed into the system before the final adapted system is produced and then put to work, or incremental mode if the adaptation data is continually fed for adaptation while the system is already at work [54].

Currently there mainly three categories of speaker adaptation techniques:

(1) Maximum a *posteriori* (MAP) adaptation, representative of Bayesian-based adaptation.

(2) Maximum likelihood linear regression (MLLR) adaptation, representative of transformation-based adaptation.

(3) Eigenvoice adaptation.

Before the advent of eigenvoice approach in 2000, MAP and MLLR adaptation are the most commonly used techniques for speaker adaptation, and practically are still seen working in almost all speech recognition systems nowadays. The schemes of the three speaker adaptation will be described in the following subsections.

2.2.1 Bayesian-based Adaptation

In early 90s, Lee, Lin and Juang reported speaker adaptation for an HMM with parameters of continuous density (CDHMM) [55], in which the parameter estimation was accomplished by segmental *k*-means algorithm which was developed in their earlier researches for HMM parameter estimation/training [56, 57]. In these works, speaker adaptation of CDHMM parameters is formulated as a Bayesian learning procedure, where prior information were involved in the computation of Bayes theorem $P(\lambda | O)$ where λ is the model parameters and O is the sequence of observations. On this basis, Gauvain and Lee then released in 94 the MAP adaptation by maximum a *posteriori* estimate of the HMM parameters [58]. MAP adaptation is thus Bayesian-based and offers a framework of incorporating newly acquired speaker-specific data into the existing models.

Assume that the CDHMM parameters are characterized by the parameter vector $\lambda = \{w_{ik}, \mu_{ik}, \Sigma_{ik}\}$, where w_{ik}, μ_{ik} and Σ_{ik} are the mixture gain, mean vector and covariance matrix of the *k*-th mixture component from the *i*-th state, respectively. The parameter vector λ is a random vector. A prior knowledge about the random vector is available and characterized by a prior probability density function $p(\lambda)$ where λ is to be determined as the input sequence is observed. Let $Y = (y_1, ..., y_T)$ be a given set of *T* observations. The MAP estimate for λ is defined as

$$\lambda_{MAP} = \arg\max_{\lambda} [p(\lambda | Y)]. \qquad (2\text{-}10)$$

Then the MAP estimate for λ is obtained by solving

$$\frac{\partial}{\partial \lambda} p(\lambda \mid y_1, y_2, ..., y_T) = 0. \tag{2-11}$$

By using Bayes theorem,

$$\begin{aligned} p(\lambda \mid Y) &= p(\lambda \mid y_1, y_2, ..y_T) \\ &= \frac{p(y_1, y_2, ..y_T \mid \lambda) p(\lambda)}{p(y_1, y_2, ..y_T)}. \end{aligned} \tag{2-12}$$

Then Eq. (2-10) can be rewritten as follows:

$$\lambda_{MAP} = \arg\max_{\lambda} [p(Y \mid \lambda) p(\lambda)] \tag{2-13}$$

To accomplish the estimation of the model parameter vector λ, the well-established segmental *k*-means algorithm can be used, and the execution is done in an iterative process as follows:

(1) Obtain the optimal state segmentation of a given observation sequence *Y*, based on a given model λ, i.e.,

$$\hat{s} = \arg\max_{s} P(Y, s \mid \lambda) P(\lambda), \tag{2-14}$$

where $s = (s_0, s_1, ..., s_t, ..., s_T)$ is a state sequence.

(2) Based on the optimal state sequence $\hat{s}$, find the MAP estimate

$$\hat{\lambda} = \arg\max_{\lambda} P(Y, \hat{s} \mid \lambda) P(\lambda). \tag{2-15}$$

(3) Iterates from (1) until some predefined equilibrium is reached.

Assume that the mean μ is random with a prior distribution $P_0(\mu)$ and the variance σ^2 is known and fixed, then the conjugate prior [59, 60] for μ is also a Gaussian distribution with mean γ and variance $\tilde{\tau}^2$, as already shown in [61]. And if the conjugate prior for the mean μ is substituted into Eq. (2-13), the MAP estimate for the adapted parameter μ as derived in [61] would appear as a weighted average of the prior mean γ and the mean of the adaptation observation data $\bar{y}_k$:

$$\hat{\mu}_k = \frac{N_k \cdot \tilde{\tau}^2}{\sigma^2 + N_k \cdot \tilde{\tau}^2} \bar{y}_k + \frac{\sigma^2}{\sigma^2 + N_k \cdot \tilde{\tau}^2} \gamma, \tag{2-16}$$

where N_k is the total number of training samples observed for the corresponding recognition unit with the *k*-th Gaussian and $\bar{y}_k$ is the sample mean with the *k*-th Gaussian.

Let $\tau = \sigma^2 / \tilde{\tau}^2$ and the prior mean γ be replaced by the mean parameter of the initial model with the *k*-th Gaussian, μ_k, Eq. (2-16) could be reformed as

$$\hat{\mu}_k = \frac{N_k}{\tau + N_k}\bar{y}_k + \frac{\tau}{\tau + N_k}\mu_k, \tag{2-17}$$

where τ is a parameter which gives the bias between the maximum likelihood estimate of the mean from the data and the prior mean. That is, τ is a prior density parameter that controls the balance between the prior knowledge and the adaptation data.

Note that, however, the data available for adaptation is often quite limited and most likely could cover a small portion of speech patterns in HMMs, which implies that many HMM parameters will not be adjusted by the nature of Bayesian-based adaptation. As a result, vector field smoothing (VFS) was proposed as a supplement for broadening the extent of adaptation in the HMM parameter vector space [62-65]. The rationale behind VFS adaptation is that, by exploiting the spatial coherence of vector distributions in HMM, the unadapted HMM parameter vector might be "purposely" adjusted in accordance with the MAP adapted vectors nearby.

To be specific, consider an unadjusted parameter vector μ_j and *k* of MAP adapted vectors $\hat{\mu}_k$'s with initial counterparts μ_k's lying in the vicinity of μ_j in the HMM vector space. The amount of MAP adaptation to μ_k is referred to as the transfer vector v_k,

$$v_k = \hat{\mu}_k - \mu_k. \tag{2-18}$$

Given the adapted vectors around, how much adaptation to μ_j should be expected? A weighted average of v_k's as shown in Eq. (2-19) would be a quite natural choice.

$$\tilde{\mu}_j = \frac{\sum_{k \in N(j)} \lambda_{j,k} \cdot v_k}{\sum_{k \in N(j)} \lambda_{j,k}} + \mu_j ,$$

$$\lambda_{j,k} = \exp\left(\frac{-d_{j,k}}{f}\right), \tag{2-19}$$

where $\tilde{\mu}_j$ is the estimate of the untrained mean vector with the *j*-th Gaussian μ_j;

$N(j)$ indicates the set of *K*-nearest neighbor mean vectors, μ_k's, to μ_j;

$\lambda_{j,k}$ represents the weighting coefficient determined by the distance $d_{j,k}$ between μ_j and μ_k, and

f denotes the weight control parameter.

A typical VFS adaptation thus comprises three steps:

(1) transfer vectors calculation for all MAP adapted parameter vectors by Eq. (2-18),

(2) interpolation of transfer vectors for adapting the unadjusted vector by Eq. (2-19),

(3) smoothing.

The composite of MAP-VFS adaptation has been proven to be more robust than MAP adaptation in recognition performance when given the same limited amount of adaptation data. Still there are rooms for MAP-VFS enhancement when the quality of MAP adaptation is in question, which is an issue to be addressed in Chap. 5.

2.2.2 Transformation-based Adaptation

In the transformation-based model adaptation, certain appropriate transformations have to be derived from a set of adaptation utterances acquired from a new speaker and then applied to clusters of HMM parameters. A bias transformation by adding a cepstral bias for model adaptation is the simplest form of transformation, which is easy to estimate and perform, as was done in [66]. Usually, adding a bias alone could not take care of the variations in test environments or among different speakers. An affine transformation (linear transformation) over HMM parameters in general offers a more appropriate model and there have been numerous adaptation schemes using affine transformations. In the work by Leggetter et al. [67], MLLR adaptation was firstly proposed under the framework of affine transformation, which has become quite popular and successful for its rapid adaptation. However, it is necessary to have sufficient adaptation data to ensure the estimate of the MLLR transformation, and various solutions have been suggested for further reinforcement. For instance, instead of using the maximum likelihood (ML) estimate in the MLLR scheme, the maximum a *posteriori* estimate is used to estimate the transformation parameters by maximizing the posterior density [68, 69]. In addition, it is suggested in [70, 71] that a prior distribution for calculating the mean transformation matrix parameters is used, which is generally dubbed as the MAPLR technique. Besides using the estimate of MAP style for acquiring transformation parameters, an alternative using a variant of the Expectation-Maximization (E-M) algorithm [72] to optimize a discounted likelihood criterion, the so-called discounted likelihood estimation, was proposed in [73]. Theoretical formulations of classic transformation-based adaptation schemes, MLLR and MAPLR, are briefly described as follows.

Under the framework of the transformation-based speaker adaptation, it generally starts with a set of SI HMMs, Λ, to which certain transformation F_η with parameters η derived from adaptation data, *Y*, of a new speaker is to be applied such that the transformed model $F_\eta(\Lambda)$ would recognize the incoming speech better than Λ did. The transformation parameters η, called linear regression parameters, are usually assumed to be fixed and then be estimated via statistical measures under specific criteria such as ML or MAP, as were done in [67] and [70] respectively.

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

$$\hat{\eta}_{ML} = \arg\max_{\eta} p(Y \mid \Lambda, \eta) . \tag{2-20}$$

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

$$\hat{\mu}_s = A_s \cdot \mu_s + b_s , \tag{2-21}$$

which sometimes is written as

$$\hat{\mu}_s = W_s \cdot \xi_s , \tag{2-22}$$

and ξ_s is the extended mean vector in the form

$$\xi_s = [\omega, \mu_{s_1}, \ldots, \mu_{s_n}]' , \tag{2-23}$$

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

The transformation matrix W_s is to be estimated such that the likelihood of the adaptation data is maximized, for which a closed form solution is available in [67] by solving the following equation,

$$\sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_{s_r}(t) \Sigma_{s_r}^{-1} o_t \xi_{s_r}' = \sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_{s_r}(t) \Sigma_{s_r}^{-1} W_s \xi_{s_r} \xi_{s_r}' , \tag{2-24}$$

where $\gamma_{s_r}(t)$ is the total occupation probability for the state s_r at time t given 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.

Apart from MLLR, Chesta et al. [70] suggests that the prior density can be taken into account in the estimation process of transformation parameters by using a maximum a *posteriori* criterion:

$$\hat{\eta}_{MAP} = \arg\max_{\eta} p(\eta \mid Y, \Lambda), \tag{2-25}$$

which is proportional to $\arg\max_{\eta} p(Y \mid \eta, \Lambda) p(\eta)$. According to this criterion, the maximum a *posterior* linear regression (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 [70]

$$\sum_{k=1}^{p}\sum_{l=1}^{p+1} w_{kl}\left[\sum_{n=1}^{N}\sum_{m=1}^{M}\left(\sum_{t=1}^{T}\gamma_t(n,m)\right) r_{ik}\bar{\mu}_l\bar{\mu}_j + \frac{1}{2}\sigma_{ki}\phi_{jl} + \frac{1}{2}\sigma_{ik}\phi_{lj}\right] =$$
$$\sum_{k=1}^{p}\sum_{l=1}^{p+1}\left[\sum_{n=1}^{N}\sum_{m=1}^{M}\left(\sum_{t=1}^{T}\gamma_t(n,m)o_k(t)\right) r_{ik}\bar{\mu}_j + \frac{1}{2}\sigma_{ki}m_{kl}\phi_{jl} + \frac{1}{2}\sigma_{ik}m_{kl}\phi_{lj}\right], \quad \begin{matrix} 1 \le i \le p \\ 1 \le j \le p+1, \end{matrix} \tag{2-26}$$

where $w_{kl} \in W_s$, $\gamma_{ik} \in R_{nm}$, $m_{ij} \in M$, $\sigma_{ij} \in \Sigma$, $\phi_{ij} \in \Phi$;

$\gamma_t(n,m)$ is the probability of the mixture m in state n at time *t*, given the observation $o(t)$, and

$\bar{\mu}_i$ is the i^{th} 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. (2-26) 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 [74].

A fuzzy control mechanism reinforced MLLR, called FLC-MLLR, will be presented in this dissertation to perform at a much lower computing cost than MAPLR and still be able to ensure the quality of MLLR adaptation when encountering data insufficiency.

2.2.3 Speaker-clustering-based Adaptation

The basic idea of the speaker-clustering-based adaptation is that a number of speaker clusters can be built up in advance, and the model of the current speaker is then represented as an interpolated form of the weighted sum of the speaker clusters. Such a speaker-clustering-based adaptation is also called as speaker-space-based adaptation. Mathematically, the estimated parameters of the sets of cluster models form the axes of speaker spaces and by estimating an appropriate point for the speaker in the speaker space, the mean vectors for the speaker is then determined. The eigenvoice approach can be regarded as the generalization of speaker-clustering adaptation techniques.

R. Kuhn, et al. [10] firstly 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. By the eigenvoice approach, the number of parameters required to be estimated would be reduced greatly but still capable of retaining the overall system characteristics to capture the variance between speakers.

Typically, the eigenvoice approach needs to take care of two things, namely eigenvoice construction and coefficient estimation. In the eigenvoice construction phase, referring to Fig. 1.4, a set of *N* well-trained SD models must be established first. Then, the model parameters of each SD model are “vectorized”, forming a set of *N* “supervectors”. Space dimension reduction techniques, such as principal components analysis (PCA), are then applied to the set of *N* supervectors to obtain *N* eigenvectors with dimension *D*, also called as “eigenvoices”. In general, only the first *K* eigenvoices are kept which are significant as they possess most information from speech data and thus are capable of representing all the variations in considerations.

Finally, by these *K* eigenvoices, an accurate speaker space “*K*-space” will be spanned and acquired. In the coefficient estimation phase, adaptation is then performed using the maximum likelihood eigen-decomposition (MLED) algorithm proposed in [10], which estimates a set of weights to find a weighted combination of eigenvoices.

Following the eigenvoice representation, the eigenvoice-versioned MLLR and MAPLR adaptation have been reported in [11] and [12] respectively where effective hybrids of MLLR-/MAPLR-eigenvoice adaptation are conceived. For the time being, the eigenvoice-based approach has received intensive attentions and various extensions of eigenvoice adaptation have been developed [13-20].

2.3 Audio Event Detection

The audio event detection system is designed for picking up a designated acoustic phenomenon when it appears in a certain acoustic background, and consequently the operations basically involve the comparison of the input audio signals against two acoustic models (the singular and the normal) and the decision about whether an audio event has occurred or not. Fig. 2.4 shows the architecture of a typical audio event detection system associated with two sound models where the input audio stream is segmented into the frame sequence, from which acoustic features are to be extracted for estimating the likelihood scores of both the normal and the singular situation via the classifier operation. When collecting the likelihood estimates to the degree that a decision can be made, the classifier then makes its call.

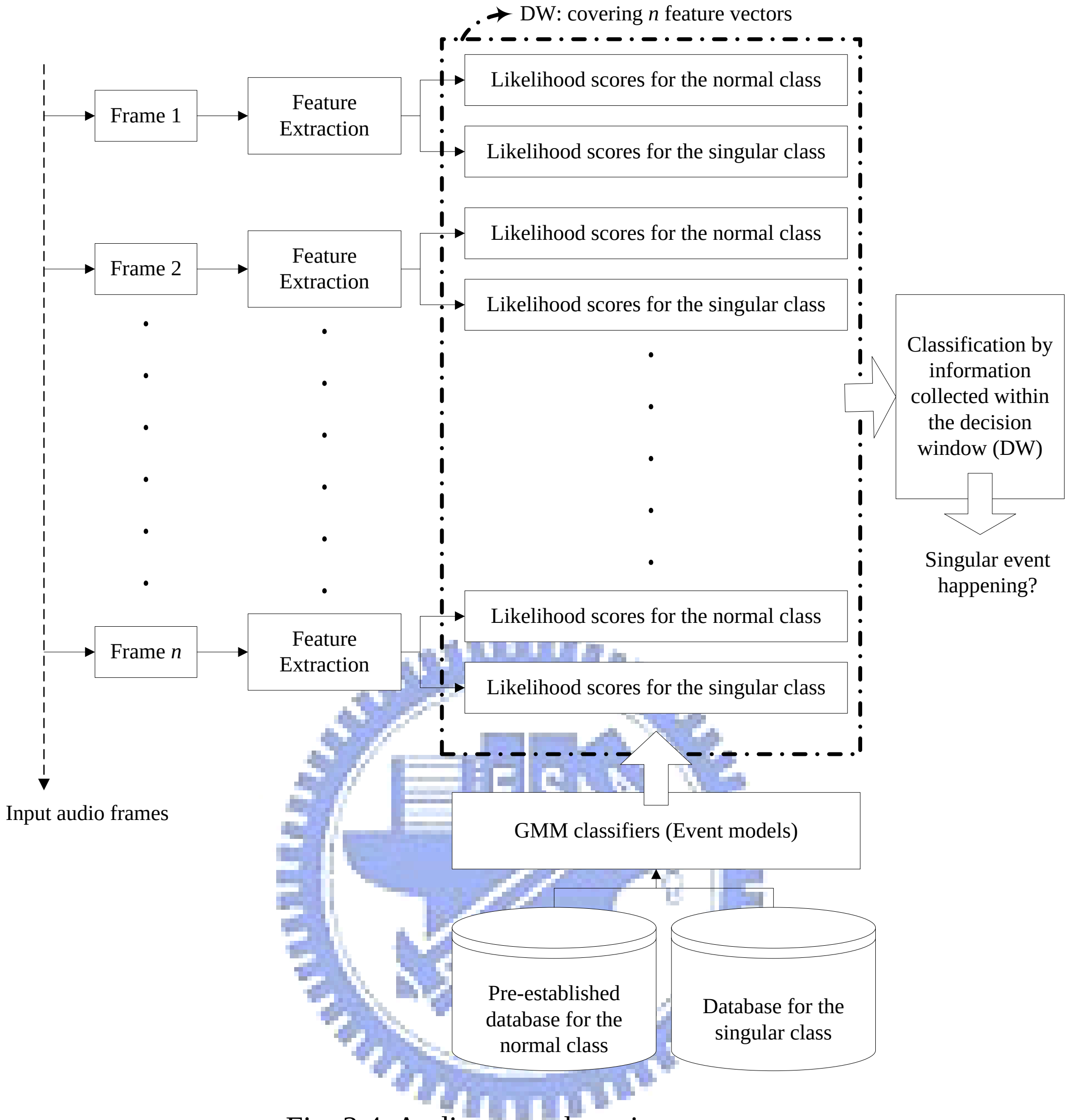

DW: covering *n* feature vectors
Frame 1
Feature Extraction
Likelihood scores for the normal class
Likelihood scores for the singular class
Frame 2
Feature Extraction
Likelihood scores for the normal class
Likelihood scores for the singular class
Classification by information collected within the decision window (DW)
Singular event happening?
Frame *n*
Feature Extraction
Likelihood scores for the normal class
Likelihood scores for the singular class
Input audio frames
GMM classifiers (Event models)
Pre-established database for the normal class
Database for the singular class

Fig. 2.4. Audio event detection system.

For constructing such a system as in Fig. 2.4, several issues have to be resolved.

- acoustic features to be extracted:

LPC, LPCC and MFCC, for example, are good candidates to be considered.

- acoustic/sound models:

In what kind of representations and how the model parameters are to be determined are the primary concerns. For the representation, alternatives like GMM [75], HMM [76] or Bayesian network [77] are available. GMM is in relatively extensive use for its approximation with ease of arbitrary forms of probability density distributions [78]; GMM is also frequently seen employed in the field of speaker identification for its capacity in categorizing voice patterns.

- the criteria for decision making:

  How the likelihood estimates are to be calculated and accumulated, how frequently the decision should be made, and the possibility of making decision not at regular time intervals but being situation driven are all interesting subjects to be explored.

2.3.1 GMM Models and Classifiers

In this dissertation, GMM models are adopted in the development of an audio event detection system for female screaming in the contexts of office space, parking lot and living room. The setting up of models during the training phase and the operation of the GMM classifier during the recognition phase are described in the following.

2.3.1.1 GMMs Establishment

Mathematically, a GMM is a weighted sum of *M* Gaussians, denoted as

$$\lambda = \{w_i, \mu_i, \Sigma_i\},\ i = 1, 2, ..., M\ ,\ \sum_{i=1}^{M} w_i = 1\ , \tag{2-27}$$

where w_i is the weight, μ_i is the mean and Σ_i is the covariance.

To determine the GMM model parameters for a certain sound class, the E-M algorithm as suggested in [72] is readily applicable. It is noted that before running the E-M algorithm, a crucial job is to initialize the model first, *i.e.*, to assign starting values to the parameters, which can be realized by a binary splitting vector quantization algorithm [79]. With the initial model parameter settings, the E-M process starts iteratively maximizing the likelihood estimate of the training data by adjusting the initial model parameters; specifically, the expectation and the maximization steps in the E-M process are repeated so that the parameter set $\lambda=\{w_i,\mu_i,\Sigma_i\},\ i=1,2,...,M$ of the GMM converges to an equilibrium state. The E-M algorithm implemented in the system to establish a GMM model, given a set of acoustic feature vectors $\mathrm{X}=\{x_n \mid n=1,2,...,N\}$, is detailed below:

(1) λ initialization is performed by a binary splitting vector quantization algorithm [79]; Σ_i is in diagonal form for computational consideration; M is determined by the Bayesian Information Criterion as suggested in [80].

(2) The computation for GMM parameters is, as suggested by the name E-M, basically an iterative process through which GMM parameters are progressively updated for maximizing the expectation value of the acoustic data.

REPEAT

{Expectation computation:

$$f(i \mid x_n,\lambda)=\frac{w_i\cdot b_i(x_n)}{\sum_{k=1}^{M} w_k b_k(x_n)}, \tag{2-28}$$

where

$$b_i(x_n)=\frac{1}{(2\pi)^{D/2}\cdot|\Sigma_s|^{1/2}}\cdot\exp\left\{-\frac{1}{2}(x_n-\mu_s)^T(\Sigma_s)^{-1}(x_n-\mu_s)\right\}. \tag{2-29}$$

λ-update for $f(\cdot)$ maximization:

$$w_i=\frac{1}{N}\sum_{n=1}^{N} f(i \mid x_n,\lambda). \tag{2-30}$$

$$\mu_i=\frac{\sum_{n=1}^{N} f(i \mid x_n,\lambda)\cdot x_n}{\sum_{n=1}^{N} f(i \mid x_n,\lambda)}. \tag{2-31}$$

$$\Sigma_i = \frac{\sum_{n=1}^{N} f(i \mid x_n, \lambda) \cdot (x_n - \mu_i) \cdot (x_n - \mu_i)^T}{\sum_{n=1}^{N} f(i \mid x_n, \lambda)} . \tag{2-32}$$

}***UNTIL*** (λ convergence achieved)

The number of iterations typically goes as high as several thousands. In the dissertation, three GMM models for the auditory contexts “office space”, “parking lot” and “living room” are established, respectively; also built are three GMMs for “female screaming” in each of the three auditory contexts with recordings collected from a group of females. And by the end of the training phase, six sets of λ parameters (6 GMM models, that is) are determined.

2.3.1.2 GMM Classifier

After the training, the recognition procedure can then be executed based on these trained GMM models. Note that the classifier deployed here is basically a GMM classifier consisting of two separate GMM models, one for background sound, and the other for singular sound. Consider the classifier operating with a decision window (or equivalently, over a time interval) covering *n* acoustic feature vectors of *D* dimensions, $\mathrm{X} = \{x_i \mid i = 1, 2, ..., n\}$, together with two sound models, λ_1 for normal events and λ_2 for singular events.

During the recognition phase, the class of X is determined by maximizing a *posteriori* probability $P(\lambda_s \mid X)$,

$$\hat{s} = \max_{s=\{1,2\}} P(\lambda_s \mid X) = \max_{s=\{1,2\}} \frac{f(X \mid \lambda_s)}{f(X)} \cdot P(\lambda_s) . \tag{2-33}$$

Note that

$$f(x_i \mid \lambda_s) = \sum_{j=1}^{M} w_j \cdot b_j(x_i) , \tag{2-34}$$

and

$$b_j(x_i) = \frac{1}{(2\pi)^{D/2} \cdot \left|\Sigma_s\right|^{1/2}} \cdot \exp\left\{-\frac{1}{2}(x_i - \mu_s)^T (\Sigma_s)^{-1} (x_i - \mu_s)\right\}. \qquad (2\text{-}35)$$

However, in real implementation, Eq. (2-33) is replaced by

$$\hat{s} = \max_{s=\{1,2\}} \sum_{i=1}^{n} \log f(x_i \mid \lambda_s), \qquad (2\text{-}36)$$

for simplicity.

And at the end of the recognition procedure, the signal X is then classified as one of the two sound classes indicated by $\hat{s}$.

2.3.2 Decision Window of the Classifier

The so-called decision window (DW) used for classification is in fact a time period covering a predetermined number of audio frames, within which successive analysis is conducted and then the decision as to whether an audio event is detected over the associated time span is made. For each audio frame, two likelihood scores are computed, the normal and the singular, using Eq. (2-34) based on the two GMM models. Within the decision window, all normal and singular estimates are respectively taken in log-values and accumulated, and whichever greater determines the class of the DW as the normal or the singular, as indicated by Eq. (2-36). In conventional processing, a fixed-length DW (e.g., 0.5 sec., 1 sec., 2 sec., etc.) is set to accumulate the log-likelihood scores of each audio frame [28, 30, 32], as shown in Fig. 2.5 where the number of frames covered in DW time span is thus held constant, *n*.

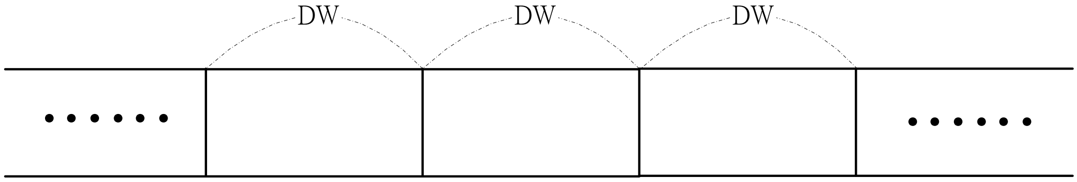


Fig. 2.5. The conventional fixed-length decision window (DW).

Audio event detection systems using fixed-length decision window are ubiquitously seen, for instance, to detect gun shots [28] or coughing in an office [30]. The use of fixed-length DW, however, is plagued by the problem of window sizing. The setting of a relatively narrow DW may potentially increase the rate of false alarms in the case of sudden and abrupt fluctuations in the background acoustic condition, and that of a too wide DW may not suffice the need of the real-time response as decisions are made at a long periodicity. An audio event detection system with a variable-sized DW governed by a fuzzy logic controller is proposed in this dissertation to regulate the length of DW according to the recent situation development in the background acoustics so that the system will be always aware of the occurrence of the specific audio event even in presence of a complicated background environment.

Chapter 3

Fuzzy Set Theory and Logic Control

Fuzzy set theory, since its inception in 1965 by Lofti A. Zadeh [81-86], has evolved with tremendous success in depth and breadth on both its theoretical development and applications to practical and difficult problems of various natures. Fuzzy set theory has embraced (or conversely been embraced by) many well-established mathematic disciplines such as logic/inference, probability/statistics, graph/relation and algebra etc. and resulted in a whole new series of theoretical establishment due to the injection of a new ingredient: fuzziness, by which the gate to a new dimension is opened and associated issues are explored. Because of its capacity of dealing with fuzziness, for which Zadeh had a perhaps the best interpretation of all: "everything is a matter of degree", large-scaled applications with inherent nature of uncertainty/ambiguity/imprecision then could be handled with systematic engineering approaches on a rigid theoretic ground. How successful fuzzy set theory has been and will be? Perhaps that decision making in many domains where strategic or operational decisions were used to be made by human domain experts in their professional careers are now given by fuzzy systems of all kinds with confidence says it all. A fine reference by Zimmermann [87] is highly recommended for gaining an overall picture covering the theoretical/technical/application aspects of the development in details or in a grand view.

Strangely enough, despite its original conception in Europe for over four decades, the western academic circle didn't seem to realize its value until the end of 1980s when the Japanese started, in overnight, advocating "Everything is of Fuzzy and by Fuzzy" in their products of home appliances and industrial controllers. Fuzzy logic control is in fact merely one among the innumerable applications of fuzzy theory, referring to the use of fuzzy logic operations for the automation of an engineering/technical process, usually small-scaled and man-maneuvered; temperature control or audio information processing in the author’s case, for instance.

In the following, the intuition behind the fuzzy theory, including the constituent entities, will be introduced. The major components comprising the operational space are as follows [87]:

(1) Fuzzy Set $\tilde{A}$:

$\tilde{A} = \{x, \mu_{\tilde{A}}(x) \mid x \in \mathrm{X}\}$ where

- X refers to a set of entities of certain attribute like AGE or LOOKING with certain degree of ambiguity in nature; for instance, X may concern the matter of AGE and contains 8 elements VERY YOUNG, QUITE YOUNG, YOUNG, MORE OR LESS YOUNG, MORE OR LESS OLD, OLD, QUITE OLD and VERY OLD, or the matter of LOOKING from UNBEARABLY UGLY to ASTONISHINGLY PRETTY etc.
- $\mu_{\tilde{A}}(x)$ are measures, called membership functions of $\tilde{A}$ for giving the degree of the specific attribute for x (i.e. how old/young in terms of a value),defined by

  $$\mu_{\tilde{A}}(x) : \mathrm{D}_{\mathrm{X}} \rightarrow [0,1],$$

  and D_{X} an interval of scalars or a vector space associated with X. D_{X} may be [0, 130] as far as human lifespan is concerned or [1, 10] when talking about one’s LOOKING.

(2) Operators on Fuzzy Sets:

Operations for conventional crisp sets are extended in a way so that the aggregation, differentiation and other desired operations upon two or more fuzzy sets could be meaningful or meet the requirements of the applications. For example, the union and intersection of two fuzzy sets $\tilde{A}$ and $\tilde{B}$ defined respectively as

$$\mu_{OR}(\tilde{A}, \tilde{B}) = \max\{\mu_{\tilde{A}}, \mu_{\tilde{B}}\}$$

and

$$\mu_{AND}(\tilde{A}, \tilde{B}) = \min\{\mu_{\tilde{A}}, \mu_{\tilde{B}}\}$$

are quite common and the negation of $\tilde{A}$ is very often defined as

$$\mu_{NOT}(\tilde{A}) = 1 - \mu_{\tilde{A}}.$$

The fuzzy operators for generic operations on fuzzy sets are usually referred to as fuzzy connectives, based on which higher levels of logic analysis, inference and reasoning can be realized.

(3) Measure of Fuzziness:

A measure for the fuzziness of the fuzzy set in consideration, $Fuzz(\tilde{A})$, is often required, the formulation of which is of course function of $\mu_{\tilde{A}}(\cdot)$ and application-oriented and preferably possesses properties like

- $Fuzz(\tilde{A}) = 0$ if $\tilde{A}$ is a crisp set in X
- $Fuzz(\tilde{A})$ reaches a unique maximum if $\mu_{\tilde{A}}(x) = \frac{1}{2}$ $\forall \mathrm{x} \in \mathrm{X}$
- $Fuzz(\tilde{A}) \geq Fuzz(\tilde{A}')$ if $\tilde{A}'$ is crisper than $\tilde{A}$
- $Fuzz(\tilde{A}) = Fuzz(-\tilde{A})$ when $-\tilde{A}$ is the complement of $\tilde{A}$

The issues of applying the fuzzy set to logic control will be briefed in Section 3.2.

3.1 Fuzzy Schemes and Speech Recognition

Fuzzy approaches have been widely applied to the field of speech recognition for many years, playing a variety of roles from data clustering, logic reasoning, to neural network configuration for speech recognition.

(1) Fuzzy data clustering:

In [88], Bezdek developed a clustering algorithm for improving the weakness of K-Means clustering algorithm, in which fuzzy scheme was exploited to consider the relationship in data attributes. Bezdek's method later became quite popular and widely known as FCM (Fuzzy C-Means) algorithm. In [89], a revised version of FCM algorithm was used to generate phonetic tied-mixture HMM (FPTM) for reducing the parameter size and improving the robustness of parameter training. In the work by Li et al. [90], the FCM was applied to Mandarin four-tone recognition, where the tone value can be determined by the maximum memberships. Tran et al. presented a generalized fuzzy manipulation using FCM and fuzzy entropy in statistical modeling for speech recognition [91].

Another line of fuzzy data clustering concerns the use of vector quantization (VQ). VQ is a standard technique for quantizing a set of scalars (mathematically the vector components) among which statistical dependencies are to be exploited, if ever exist, for optimal reconstruction levels or steps in coding process; the result of VQ is effectively as data clustering from the perspective of data classification and has been widely employed in high-dimensioned data applications, including speech recognition [92-94]. VQ variants with fuzzy ingredients introduced into the quantization process have been seen for the purpose of speech recognition. In [95], a minimum FVQ error criterion was devised for unsupervised speaker adaptation, which showed that the same recognition accuracy as a supervised speaker adaptation could be achieved by minimizing the overall FVQ errors. Based on the concept of FVQ, Shikano et al., proposed a fuzzy codebook mapping algorithm to speaker adaptation for mapping from a speaker to a standard speaker

[96]. In addition, in the work by Lin et al., the FVQ technique is embedded in neural network for isolated word speech recognition [97, 98]. In [99], a composite of Multi-Layer Perceptron (MLP) neural network and FVQ was presented. Compared with MLP-VQ, MLP-FVQ will provide richer information about recognition results, an output vector whose components indicating the relative closeness of each label to the input.

(2) Fuzzy logic and reasoning applications:

Fuzzy logic and reasoning has also been applied to speech recognition recently. In [100, 101], Zhao and Woo proposed a fuzzy speech recognition approach based on the power distribution pattern of a speech segment using fuzzy logic. Compared to speech recognition using typical hidden Markov models, the work using fuzzy logic was simpler to implement in real-time recognition systems. In the work by Halavati et al. [102], speech spectrogram was conversed into a linguistic description based on arbitrary colors and lengths, following which, fuzzy measures, fuzzy reasoning and a genetic algorithm were used to describe phonemes, perform the recognition procedure and optimize phoneme definitions, respectively.

(3) Fuzzy neural network applications:

Fuzzy neural network (FNN) that combines both the fuzzy logic and the neural network is frequently seen in speech recognition lately. In contrast to the conventional HMM-based recognition, FNN has the advantages of efficient learning, adaptation and connectionist structure when carrying out speech recognition [103]. In [104], a neuro-fuzzy classifier is designed to perform SI model speech recognition, where the classifier is an MLP model incorporated with fuzzy operations and therefore inherits the strength of both neural networks and fuzzy systems. The work by Kasabov et al. applied FNN to model a phoneme-based speech recognition system, which acquired quite satisfactory recognition performance [105]. In the study of [106], a variant of FNN, called modular general fuzzy min-max (MGFMM) neural network, was proposed to modify the transfer function of the output layer of general fuzzy min-max neural network (GFMM) for improving the recognition accuracy of speech recognition. Other related works of speech recognition by FNN can be seen in [107-109]. In the specific area of speaker adaptation, however, the use of fuzzy scheme/mechanism is rarely seen. Lin et al. proposed a speaker adaptation scheme in a perceptron-NN for speech recognition [110], where the fuzzy perceptron approach is applied to generate hyperplanes which separate speech patterns of each class from the others. In particular, speaker adaptation is considered as a procedure of tuning the trained hyperplanes when there is recognition error caused by a new speaker. The work by Lin et al. is thus essentially more of a fuzzy-neural classification of speech patterns in the perceptron neural space, instead of an adaptation scheme as being proclaimed. In addition, Gales applied the fuzzy scheme to MLLR speaker adaptation (Fuzzy-MLLR) to further enhance the classification of regression matrices of MLLR [111], which in nature belongs to fuzzy clustering applications.

Although many fuzzy approaches have been widely used in various sub-areas of speech recognition as mentioned hereinbefore, it has not been seen for the use of FLC in HMM speaker adaptation, or even in speech recognition. Based on the methodology of FLC, a series of speaker adaptation computations under FLC regulation are designed in the dissertation. Basic concepts and architectures of the FLC underlying the main theme of the dissertation are to be introduced in the following sections.

3.2 Fuzzy Logic Controller (FLC)

As mentioned earlier fuzzy logic control concerns the automation of a control process for which the operator’s knowledge/expertise/experience regarding the process control imparted in oral or written form has to be translated so as to fit in the framework of fuzzy logic control, together with other accommodation or extension in the fuzzy set theory specific to this particular application. An excellent book on this subject by Zadeh et al. [112] is highly recommended.

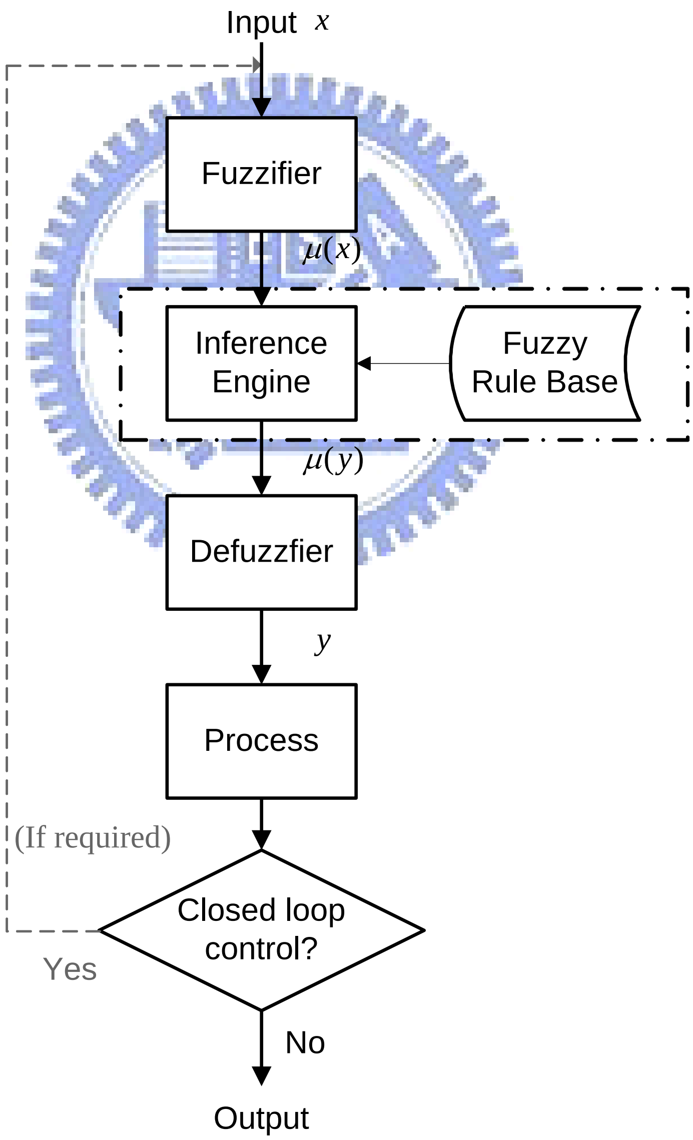


Fig. 3.1. Architecture of a typical FLC.

Fig. 3.1 shows the architecture of a typical FLC and the role of each constituent module and the input/output are described as follows.

(1) Input:

usually signals or quantities of certain attribute in precise magnitudes (e.g., temperature measured in Celsius)

(2) Fuzzifier:

the precise and exact values of the input have to be transformed by the fuzzifier through the use of membership functions such that fuzzy implications like MODERATE, VERY LOW or HIGH could be attached so as to be processed by the next module.

(3) Inference Engine:

performing analysis or reasoning on the input information for making control decision like “PUT ON A LIGHT JACKET”, “TURN ON THE HEATER A BIT MORE” or a conclusion like “THE AUTUMN HAS COME” under the constraints from the Fuzzy Rule Base.

(4) Fuzzy Rule Base:

a representation of the domain knowledge in fuzzy terms, typically in either of the two forms:

$$\begin{aligned} &\mathit{Rule}\ i: \text{IF } x^i \text{ is } a^i \oplus \cdots \oplus y^i \text{ is } b^i, \\ &\qquad \text{THEN } z^i = c^i, \end{aligned} \qquad \text{(state evaluation rules)}$$

$$\begin{aligned} &\mathit{Rule}\ j: \text{IF } z^j = c^j, \\ &\qquad \text{THEN } x^j \text{ is } m^j \otimes \cdots \otimes y^j \text{ is } n^j, \end{aligned} \qquad \text{(object evaluation rules)}$$

where,

- $\oplus$ and $\otimes$ are fuzzy connectives.
- a^i, b^i, m^j and n^j are elements in associated fuzzy sets $\tilde{A}$, $\tilde{B}$,

$\tilde{M}$ and $\tilde{N}$ respectively.

(5) Defuzzification:

the decision of control action made in the fuzzy context has to be transformed so that a corresponding exact value such as “open the valve of the heat outlet by 10 %” would be available for sending to the physical world of the process

In designing an FLC, various issues regarding the structure or operation of each module in Fig. 3.1 have to be considered, the mastery of which determines the success of the FLC operation, as are addressed in the following:

(a) Input x:

- how many input signals being required
- scaling of each signal, etc.

(b) Fuzzification:

the number and types of membership functions required

(c) Rule base:

- the number of rules
- the number of antecedents, weights and membership functions of the antecedent/ consequence associated with each rule
- the structure of the rule base

(d) Inference engine:

- connectives for aggregating antecedents
- inference /reasoning schemes to be employed
- operators for aggregating the consequence of individual rules for generating a decision

Various types of FLCs have been proposed with variations in the module design considerations. The renowned Mamdani and Sugeno FLC are, for instance, different in consequence design in every individual rules; the former generates the consequence as a member in the fuzzy set associated with linguistic variables pertaining to, say, control action, whereas the latter produces a consequence in crisp form (e.g., as a scalar function of inputs in the antecedent).

Another issue for FLC design has to do with taking into account from the temporal perspective the potential variations in the process itself, for which the use of time-variant parameters in the FLC design becomes unavoidable; i.e. the FLC is preferable to be adaptive in accordance with the time-varying process. Basically the adaptation can be done by modifying the rule sets or the fuzzy set, resulting in two classes of FLCs, respectively the self-organizing and self-tuning FLC.

3.3 Takagi-Sugeno (T-S) FLC

The Takagi-Sugeno fuzzy model proposed by Takagi and Sugeno has been widely in use since it is conceptually simple and straightforward [113]. This type of fuzzy system was early used in a famous parking control of a model car [114] where an FLC is designed for the task of driving a model car to a designated parking space as shown in Fig. 3.2.

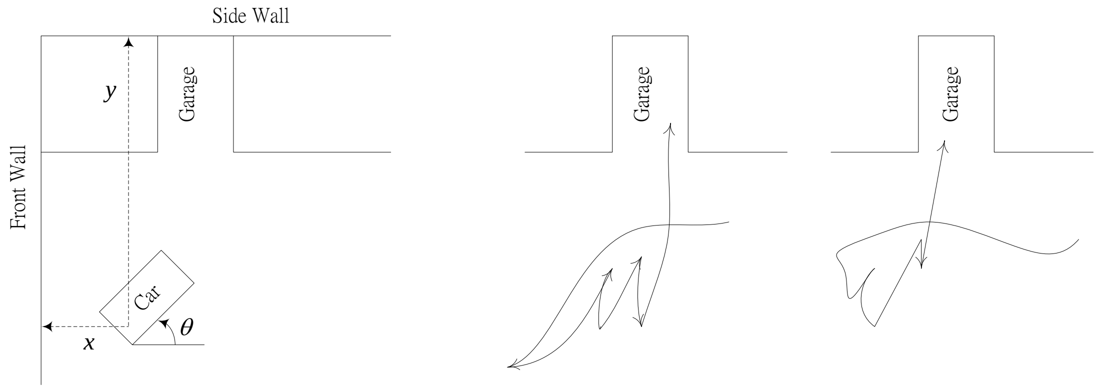


Fig. 3.2. Sugeno’s FLC for car parking.

The parking FLC by Sugeno was designed with the following specifications.

(1) Three inputs:

- (x , y) for the car position,
- θ for the car orientation.

Two outputs:

- f for the front wheels angle while driving forward,
- b for the front wheels angle while driving backward.

(2) A rule base:

- 18 rules for driving forward in which the antecedents involved x , y and θ , the consequence f is a function of x , y and θ too,
- 18 rules for driving backward with similar rule forms,
- 6 rules for speed control.

Based on which, the control goal is to construct a successive alternation of forward-backward driving actions with appropriate speed and turning such that the car can be properly parked in position. Fig. 3.2 shows two parking trajectories by the FLC which is amazingly similar to those done by human drivers. The performance is of coarse quite encouraging and thus it paves the road for subsequent applications to lots of general control problems with successfulness up to present days.

For a complex system, the T-S fuzzy design procedure presents a systematic framework of fuzzy modeling design for this system. Fig. 3.3 illustrates the design methodology. The system is decomposed into a set of subsystems for which local behaviors are identified by expressing the inputs-out mapping in terms of a fuzzy implication (or rule) where the inputs are specified in the antecedent part and the output as the linear combination of the associated inputs. The overall system output is then a function of the subsystem outputs which could be as simple as of a “linear” combination, where fuzziness of the system behaviors is to be taken care of in the coefficient handling, or of other more elaborated forms.

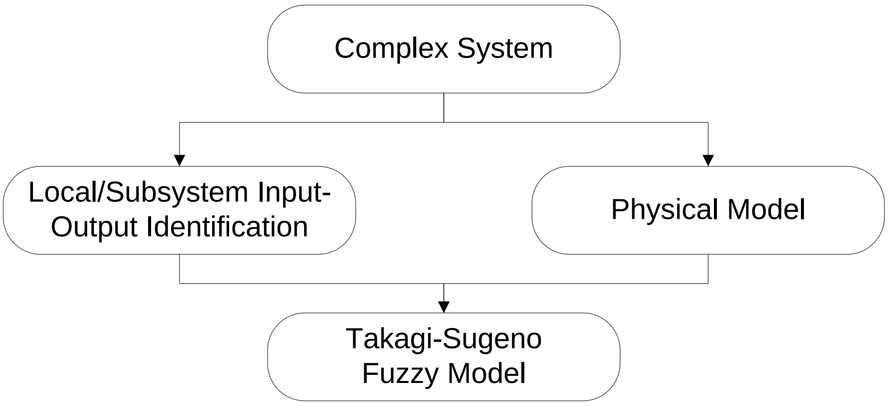

Complex System
Local/Subsystem Input-Output Identification
Physical Model
Takagi-Sugeno Fuzzy Model

Fig. 3.3. Designs of Takagi-Sugeno (T-S) fuzzy model.

Through the system decomposition, the system dynamics, which is generally complicated and nonlinear, is captured in a set of linear system models and fuzzy mechanisms are incorporated wherever necessary. The application of T-S fuzzy modeling is thus quite straightforward.

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. And the system representation would be of the form

$$\text{Rule 1: IF } x(1) \text{ is } A_1^1 \text{ and} \ldots \text{and } x(n) \text{ is } A_n^1$$

$$\text{THEN } y^1 = a_0^1 + a_1^1 x(1) + \ldots + a_n^1 x(n),$$

$$\cdot \quad \cdot \quad \cdot$$

$$\text{Rule } i\text{: IF } x(1) \text{ is } A_1^i \text{ and} \ldots \text{and } x(n) \text{ is } A_n^i$$

$$\text{THEN } y^i = a_0^i + a_1^i x(1) + \ldots + a_n^i x(n), \qquad (3\text{-}1)$$

. . .

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

THEN $y^l = a_0^l + a_1^l x(1) + \ldots + a_n^l x(n)$,

System output: $$y = \frac{\sum_{i=1}^{l} w^i y^i}{\sum_{i=1}^{l} w^i}, \text{ given that } w^i = \prod_{p=1}^{n} A_p^i(x(p)), \qquad (3\text{-}2)$$

for a system of n inputs and l implications. Note that A_p^i, $p = 0,1,\ldots,n,$ are fuzzy sets and $A_p^i(x(n))$ denotes 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 i-th consequence y^i is expressed as a linear combination of n inputs.

The output of this system is a weighted sum of functions. In Eq. (3-2), an interpolation procedure is performed among different linear functions (local models). Fig. 3.4 depicts the phenomenon of the smooth interpolation of the local models.

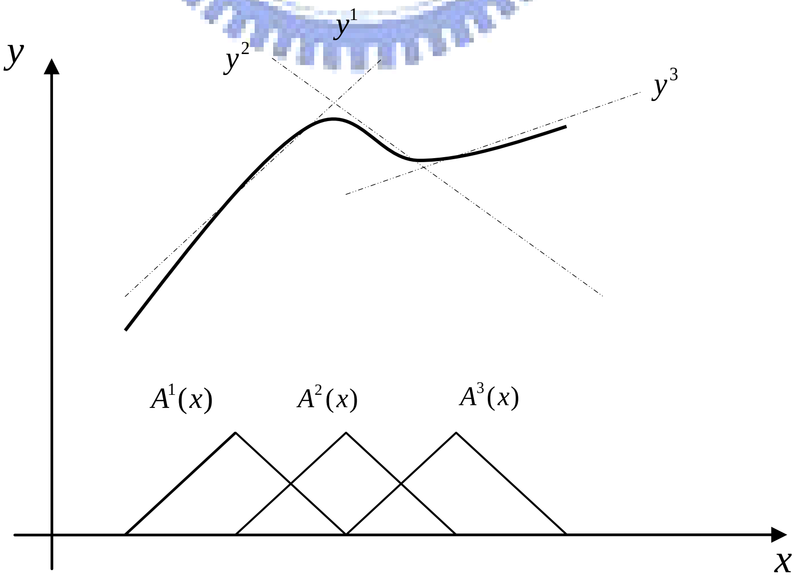


Fig. 3.4. System output of T-S fuzzy model in an interpolation form.

T-S fuzzy model has been seen in the control of the system as complicated as an electric power plant with success [115, 116], and is employed in the author's research in speaker adaptation schemes and audio event detection, as will be detailed in the next four chapters.

Chapter 4

Speaker Adaptation Based on MAP Estimation Using Fuzzy Controller

As mentioned in Section 2.2.1, MAP adaptation is a kind of direct model adaptation, which attempts to directly re-estimate the model parameters [58]. However, it is noted that MAP adaptation re-estimates only the portion of model parameter units associated the adaptation data, and therefore, MAP adaptation usually needs a large amount of data for adaptation and the performance will be improved as adaptation data increases and gets covering the model space. When the amount of data is sufficiently large, the MAP estimation yields as good recognition performances as that obtained using maximum-likelihood estimation [55]. As shown in Eq. (2-17),

$$\hat{\mu}_k = \frac{N_k}{\tau + N_k}\bar{y}_k + \frac{\tau}{\tau + N_k}\mu_k, \tag{2-17}$$

the MAP estimate of the mean is essentially a weighted average of the prior mean and the sample mean, and the weights are functions of the number of adaptation samples, given that τ being fixed. When N_k is equal to zero (i.e., no additional training data are available for adapting the *k*-th Gaussian), the estimate is simply the prior mean of the *k*-th Gaussian alone. Conversely, when a large number of training samples are used for the *k*-th Gaussian ($N_k \to \infty$, to be exaggerative), the MAP estimate in Eq. (2-17) then converges asymptotically to the maximum likelihood estimate, i.e., the sample mean parameter with the *k*-th Gaussian, $\bar{y}_k$.

Now consider the other way round with N_k being fixed, the parameter τ controls the balance in the interpolation between the $\bar{y}_k$-term and the μ_k-term, (as N_k does). It is referred to as the “adaptation speed parameter” in [65, 117] in that the speed of adaptation can be increased or held-back by choosing a small or a large value of τ. The parameter τ is also known as a "prior density parameter" since it determines, to which side of, and for how close to $\bar{y}_k$ or μ_k, the MAP-estimate of $\hat{\mu}_k$ would be.

As a general remark, that the recognition performance of adaptation, regardless of whatever adaptation schemes in consideration, would not be as good as desired given insufficient training samples N is a consensus among all. The robustness of MAP adaptation against relatively small N should not be overlooked either, and as yet in conventional schemes for MAP adaptation ([54, 117, 118], e.g.), a common value of τ was used for all the Gaussians of a given state, or for all states of an HMM, or even for all HMMs.

With all the aforementioned thoughts in mind and looking at Eq. (2-17), it would be quite natural for one to come out with the idea that $\hat{\mu}_k$ should stay in the vicinity of μ_k when N is somewhat small (by the choice of a large τ) to avoid the performance degrading caused by the potentially poor estimate of $\bar{y}_k$, and on the other hand when N is large enough, the adaptation should move toward $\bar{y}_k$ speedily. Putting such notion in terms of simple rules in plain words leads to statements as follows

(1) When N is small, τ should be large such that $\hat{\mu}_k$ sticks more to μ_k.

(2) When N is medium, τ should be medium such that $\hat{\mu}_k$ locates between $\bar{y}_k$ and μ_k accordingly.

(3) When N is large, τ should be small such that $\hat{\mu}_k$ adapts toward $\bar{y}_k$.

This is where fuzzy methodology comes into play, and how the statements of linguistic terms with uncertainty to some degree can be formulated in quantized forms for subsequent computations will be explained in the next subsection.

4.1 FCMAP Adaptation

Within the framework of fuzzy process, the formulation of the problem at hand is given as a set of three fuzzy IF-THEN rules and the system output $\tau(\cdot)$ [8].

Rule 1: If N is $M_1(N)$, then $\tau_L = f_1(N)$,

Rule 2: If N is $M_2(N)$, then $\tau_M = f_2(N)$,

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

where $M_1(N)$, $M_2(N)$ and $M_3(N)$ are the membership functions representing the degree of how much N is involved in the classes of linguistically “small”, “medium” and “large” respectively, and are defined as

$$M_1(N) = \begin{cases} 1 & N \le N_1, \\ \dfrac{N_2 - N}{N_2 - N_1} & N_1 \le N \le N_2, \\ 0 & N \ge N_2, \end{cases}$$

$$M_2(N) = \begin{cases} 0 & N \le N_1 \text{ or } N \ge N_3, \\ \dfrac{N - N_1}{N_2 - N_1} & N_1 < N \le N_2, \\ \dfrac{N_3 - N}{N_3 - N_2} & N_2 \le N < N_3, \end{cases} \qquad (4\text{-}1)$$

$$M_3(N) = \begin{cases} 0 & N \le N_2, \\ \dfrac{N - N_2}{N_3 - N_2} & N_2 < N < N_3, \\ 1 & N \ge N_3. \end{cases}$$

$f_i(N), i = 1, 2, 3$ are output functions in each rule for regulating the τ value and are defined as

$$f_1(N) = \frac{b}{\log(N) + a},$$

$$f_2(N) = \frac{N}{c}, \qquad (4\text{-}2)$$

$$f_3(N) = \frac{\log(N)}{N}.$$

Note that the definitions in Eq. (4-2) is an empirical choice among many possibilities. For the system output, $\tau(\cdot)$ is defined as [113]

$$\tau = \frac{\sum_{i=1}^{3} M_i(N) f_i(N)}{\sum_{i=1}^{3} M_i(N)}. \tag{4-3}$$

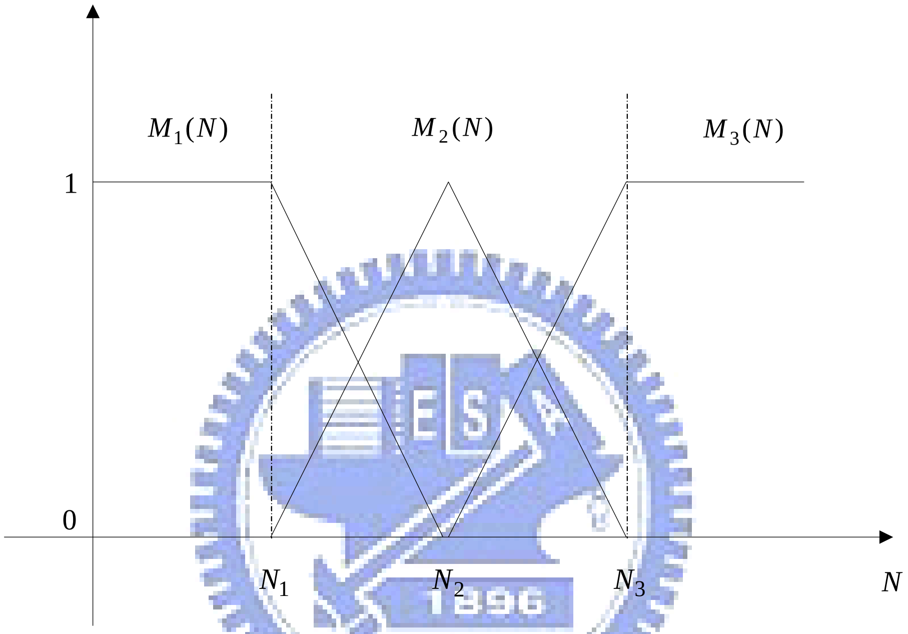


Fig. 4.1. Membership functions of fuzzy controllers for FCMAP adaptation.

By the formulation, the system now has six hyperparameters (a, b, c, N_1, N_2, and N_3) to be fixed, which will be done below:

STEP 1: Let $N_1 : N_2 : N_3 = 1:2:3$ and $N_1 = 500$. Also let $i = 0$, $a = initial value,$ and $b = initial value.$

STEP 2: Estimate the parameters a and b under the condition $N < N_1$.

Since $M_1(N) = 1$, $M_2(N) = 0$, $M_3(N) = 0$, then

$$\tau = \frac{M_1(N) f_1(N)}{M_1(N)} = f_1(N) = \frac{b}{\log(N) + a}.$$

```
$P^i$ = baseline_recognition_rate;
$a+=\Delta a$ ; $i$ ++;
$P^i$ = $speech\_recognition(\tau = \frac{b}{\log(N)+a}$, tunning_utterances);
if ($P^i > P^{i-1}$)
    Repeat
        { $a+=\Delta a$ ; $i$ ++;
          $P^i$ = $speech\_recognition(\tau = \frac{b}{\log(N)+a}$, tunning_utterances);
        } while ($P^i > P^{i-1}$);
else
    Repeat
        { $a-=\Delta a$; $i$ ++;
          $P^i$ = $speech\_recognition(\tau = \frac{b}{\log(N)+a}$, tunning_utterances);
        } while ($P^i > P^{i-1}$);
$b+=\Delta b$; $i$ ++;
$P^i$ = $speech\_recognition(\tau = \frac{b}{\log(N)+a}$, tunning_utterances);
if ($P^i > P^{i-1}$)
    Repeat
        { $b+=\Delta b$; $i$ ++;
          $P^i$ = $speech\_recognition(\tau = \frac{b}{\log(N)+a}$, tunning_utterances);
        } while ($P^i > P^{i-1}$);
else
    Repeat
        { $b-=\Delta b$; $i$ ++;
        $P^i$ = speech_recognition( $\tau = \frac{b}{\log(N)+a}$, tunning_utterances);
    } while ( $P^i > P^{i-1}$ );
return $P^i$;
```

STEP 3: Adjust the parameter N_3 under the condition $N > N_3$.

Since $M_1(N) = 0$, $M_2(N) = 0$, $M_3(N) = 1$, then

$$\tau = \frac{M_3(N) f_3(N)}{M_3(N)} = f_3(N) = \frac{\log(N)}{N}.$$

Based on the initial value of N_3 in step 2, the parameter N_3 is to be adjusted under the condition $N > N_3$ for maximizing the recognition performance by a procedure similar to the one for fixing parameter a (or b) in STEP 2. Once the suitable value of N_3 is selected, the parameters N_1 and N_2 will be further updated such that the ratio of $1:2:3$ is maintained.

STEP 4: Estimate the parameter c under the condition $N_1 < N < N_2$.

Since $M_1(N) = \frac{N_2 - N}{N_2 - N_1}$, $M_2(N) = \frac{N - N_1}{N_2 - N_1}$, and $M_3(N) = 0$,

then

$$\tau = \frac{M_1(N) f_1(N) + M_2(N) f_2(N)}{M_1(N) + M_2(N)}$$

$$= \frac{\frac{N_2 - N}{N_2 - N_1} \cdot \frac{b}{\log(N)+a} + \frac{N - N_1}{N_2 - N_1} \cdot \frac{N}{c}}{\frac{N_2 - N}{N_2 - N_1} + \frac{N - N_1}{N_2 - N_1}}$$

$$= \frac{(N_2 - N)\frac{b}{\log(N)+a} + (N - N_1)\frac{N}{c}}{N_2 - N_1}.$$

Since the values of a, b, N_1 and N_2 are already determined in previous steps, the value of c can be properly selected by utilizing the parameter fixing procedure in STEP 2.

STEP 5: Re-estimate the parameter N_3 under the condition $N_2 < N < N_3$.

Since $M_1(N)=0$, $M_2(N)=\dfrac{N_3-N}{N_3-N_2}$, and $M_3(N)=\dfrac{N-N_2}{N_3-N_2}$,

then

$$
\begin{aligned}
\tau &= \frac{M_2(N)f_2(N)+M_3(N)f_3(N)}{M_2(N)+M_3(N)} \\
&= \frac{\dfrac{N_3-N}{N_3-N_2}\cdot\dfrac{N}{c}+\dfrac{N-N_2}{N_3-N_2}\cdot\dfrac{\log(N)}{N}}{\dfrac{N_3-N}{N_3-N_2}+\dfrac{N-N_2}{N_3-N_2}} \\
&= \frac{(N_3-N)\dfrac{N}{c}+(N-N_2)\dfrac{\log(N)}{N}}{N_3-N_2}.
\end{aligned}
$$

Again, as the values of c, N_1 and N_2 have been fixed by now, a new suitable value of N_3 can be obtained in a similar way. As soon as the new value of N_3 is selected, the parameters N_1 and N_2 will be further updated again.

STEP 6: Let $\delta=\dfrac{|R^q-R^*|}{R^*}$ where R^* denotes the desired recognition rate and $R^0=R^q$. Repeat from STEP 2 until δ is less than a predefined threshold.

The FLC mechanism herein proposed for regulating the value of τ as a function of the number of adaptation samples during MAP adaptation is thus called as FCMAP.

4.2 Experiments

The experimental settings and results of the proposed FCMAP adaptation algorithm are respectively reported in the following subsections.

4.2.1 Database and Experiment Design

The speech signal was 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 was composed of a 12-dimensional mel-cepstral vector and a 12-dimensional delta-mel-cepstral vector.

Before conducting comparative experiments to illustrate the effectiveness of the FCMAP adaptation method proposed in Section 4.1, initial SI models have to be established first. The database MAT400 sub-database DB3 [119], which consists of 4800 utterances from native Mandarin speakers, was used to build up the initial SI models in the form of a set of HMM parameters. In Mandarin, a Mandarin utterance may contain one to several syllables, and each syllable consists, in terms of HMM states, of a 3-state initial part and a 6-state final part; thus the HMM of a Mandarin utterance includes the HMMs of the constituent syllables, which in turn includes an HMM of 3 states for the initial part (if exists) and an HMM of 6 states for the final part [50]. Together there are 440 states in the SI models.

A group of 10 speakers was summoned for utterance recording. 30 utterances of city names (one utterance for each of 30 cities) as adaptation data for setting up SA models and 60 utterances of city names (two utterances for each of the 30 cities) as tuning data for the FLC were collected from each of the 10 speakers. The pseudo-code sequence for tuning the hyperparameters of the fuzzy controller in the system training phase is given as follows

```
training_phase (SI_models, hyperparameters)
{     $k = 0$;
      $\overline{P}^{0}$ = Baseline recognition performance;
      Repeat
              { $k$ ++;
              $\overline{P}_{1}^{k}$ = speaker_training (SI_models, tuning_data$_1$, hyperparameters,
                                          adaptation_utterances$_1$);
                                          …
```

```
            $\overline{P}_j^k$ = speaker_training (SI_models, tuning_data$_j$, hyperparameters,
                adaptation_utterances$_j$);
            …
            $\overline{P}_{10}^k$ = speaker_training (SI_models, tuning_data$_{10}$, hyperparameters,
                adaptation_utterances$_{10}$);
            $\overline{P}^k = \frac{\sum_{j=1}^{10} \overline{P}_j^k}{10}$;
            $\Delta\overline{P} = \left|\overline{P}^k - \overline{P}^{k-1}\right|$;
        } until $\Delta\overline{P}$ < threshold;
    return $\overline{P}^k$;
};
speaker_training (SI_models, tuning_data$_j$, hyperparameters, adaptation_utterances$_j$)
// $j$ = 1,…, 10.
{
    $\overline{P}_j$ = Iterative_process (SI_models, tuning_data$_j$, hyperparameters,
    adaptation_utterances$_j$);
    // as described in Section 4.1 for maximizing the recognition rate $\overline{P}_j$.
    return $\overline{P}_j$;
};
```

where adaptation_utterances$_j$ and tuning_data$_j$ denote the adaptation utterances and the tuning utterances from the j-th speaker, $1 \le j \le 10$.

It is notable that during the execution of training_phase(·) procedure, the hyperparameter set of the FLC (a, b, c, N_1, N_2, and N_3) is continuously adapted through successive feeding of tuning_data collected from the 10 speakers; to be more specific, the state of the hyperparameter set tuned by the *j*-th speaker_training(·) is to be used as the initial state of the hyperparameters for the (*j*+1)-th speaker_training(·).

In the recognition experiments, adaptation and testing data were gathered from a new group of five speakers. All the uttered data were recorded by a close-talking microphone. Two designs were conceived for the recognition experiments:

(1) Adaptation and testing data being identical in contents:

The adaptation data consisted of 30 utterances from each speaker (one utterance for each of 30 cities). The testing data consisted of 60 utterances from the speakers, each uttering twice for 30 city names. For each speaker, 2, 6, 10, 14, 18, 22, 26, and 30 utterances were picked out from his/her 30-utterance adaptation data for SI model adaptation, and 8 sets of SA models are established per speaker. A total of 40 SA models are thus set up and used for performance comparison between MAP- and FCMAP-adaptation.

(2) Adaptation and testing data being different in contents:

The adaptation data consisted of 15 utterances from each speaker (one utterance for each of the first 15 cities of 30 cities). The testing data consisted of 30 utterances from the speakers, each uttering twice for the last 15 cities of 30 city names. For each speaker, 1, 3, 5, 7, 9, 11, 13, and 15 utterances were picked out from his/her 15-utterance adaptation data for SI model adaptation, and 8 sets of SA models are established per speaker. A total of 40 SA models are thus set up and used for performance comparison between MAP- and FCMAP-adaptation.

4.2.2 Experiment Results

The averaged recognition performance by 5 speakers of the conventional MAP with various settings of τ can be seen in Table 4.1. According to Table 4.1, the conventional MAP has better results when τ is fixed to 25 or 30. Thus, these two values of τ are chosen in the conventional MAP for comparison. FCMAP adaptation experiments were carried out for each of the 5 speakers, using the associated SA models of eight, and the recognition performances are given in Table 4.2, of which in the bottom row shows the averaged recognition rate by the 5 speakers at all eight test cases. Fig. 4.2 demonstrates the average recognition rate of the proposed FCMAP with adaptive τ against the conventional MAP with a fixed τ using identical adaptation and testing data as in the first design of the recognition experiment. It is clearly seen that the proposed FCMAP as well as MAP25 and MAP30 has an adaptive learning curve. For the conventional MAP, when the amount of training data is insufficient, the recognition rate is low, even lower than the baseline. In contrast, the recognition rate of the FCMAP is as or even better than the baseline when the amount of training data is not sufficient. Furthermore, when the amount of training data is increasing, the recognition performance of the conventional MAP becomes better than the baseline but still a little worse than that of the FCMAP. It is concluded that the FCMAP performs better than the conventional MAP especially when the amount of adaptation data is very limited. Fig. 4.3 demonstrates the average recognition rate of the proposed FCMAP with adaptive τ against the conventional MAP with a fixed τ , using testing data different from the adaptation ones as in the second design of the recognition experiment. It is clearly observed in the two recognition tests that, using identical or different data in training and testing, FCMAP behaves consistently and outperforms MAP adaptation. However, it is noted that both FCMAP and MAP adaptations have better performances in the second recognition experiment when adaptation utterances are below 15; the causes behind such behavior may be attributed to that the HMM components to be matched in the recognition phase are not adapted (or so to speak, “polluted”) by insufficient utterances during the training phase in the second experiment.

Table 4.1. Average recognition rates (%) of the conventional MAP with various τ .

Value of τ	**Average recognition rate (%)**								
	Numbers of utterances for adaptation								
	0	**2**	**6**	**10**	**14**	**18**	**22**	**26**	**30**
5	91.00	70.10	70.70	71.20	74.67	75.30	77.60	78.67	80.20
10	91.00	70.80	71.10	72.10	75.00	76.20	79.30	80.10	80.30
15	91.00	73.20	74.80	75.00	77.60	78.70	82.80	85.00	85.80
20	91.00	75.80	76.10	76.30	78.20	79.80	83.30	85.80	86.00
25	91.00	84.67	84.67	85.00	85.00	86.67	88.67	90.00	93.33
30	91.00	83.67	84.00	84.67	84.67	88.67	89.00	91.33	96.00
35	91.00	81.67	81.67	81.67	83.30	85.00	86.67	89.20	92.00
40	91.00	75.20	73.30	74.00	76.90	78.00	83.30	85.00	85.60
45	91.00	72.90	73.10	73.60	75.20	77.50	80.00	82.10	84.00
50	91.00	71.30	72.00	72.60	74.00	75.20	78.10	80.90	82.40

Table 4.2. Recognition rates (%) of FCMAP and MAP (with a fixed τ of 25 or 30).

Speaker	Adapt. method	Recognition rate (%)								
		Number of adaptation utterances								
		0	2	6	10	14	18	22	26	30
No. 1	FCMAP	90.00	90.00	90.00	90.00	90.00	93.33	95.00	96.67	98.33
	MAP25		83.30	82.50	83.60	83.70	85.85	88.67	88.90	93.33
	MAP30		82.10	82.30	83.00	83.00	88.33	88.83	90.30	96.67
No. 2	FCMAP	90.00	90.00	90.00	90.00	90.00	90.00	91.67	95.00	100
	MAP25		82.10	82.60	83.00	83.00	84.33	85.00	86.67	91.67
	MAP30		81.50	82.00	82.50	82.60	86.67	87.00	90.00	96.67
No. 3	FCMAP	91.67	91.67	91.67	91.67	91.67	93.33	91.67	93.33	95.00
	MAP25		84.67	85.00	85.00	85.10	86.67	88.67	90.00	93.33
	MAP30		83.67	84.67	85.00	85.00	88.33	88.50	91.67	95.00
No. 4	FCMAP	93.33	93.33	95.00	95.00	95.00	93.33	95.00	96.67	100
	MAP25		85.00	85.00	85.20	85.20	87.50	90.00	91.67	95.00
	MAP30		84.00	84.00	85.10	85.10	88.33	89.00	91.67	96.67
No. 5	FCMAP	90.00	90.00	91.67	91.67	91.67	93.33	91.67	93.33	96.67
	MAP25		88.30	88.30	88.20	88.10	89.00	91.00	92.80	93.33
	MAP30		87.10	87.10	87.70	87.70	91.67	91.67	93.00	95.00
Average	FCMAP	91.00	91.00	91.67	91.67	91.67	92.66	93.00	95.00	98.00
	MAP25		84.67	84.67	85.00	85.00	86.67	88.67	90.00	93.33
	MAP30		83.67	84.00	84.67	84.67	88.67	89.00	91.33	96.00

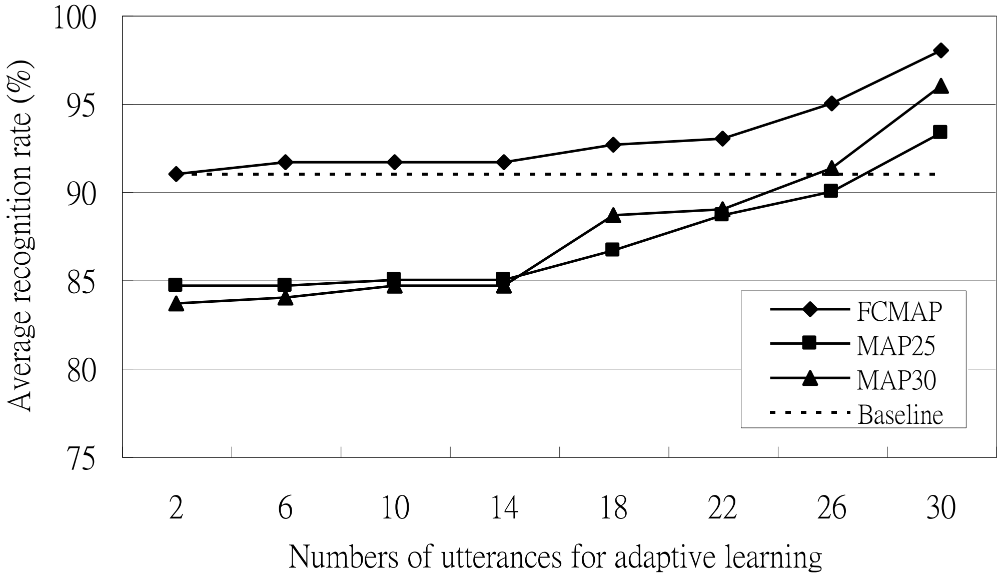


Fig. 4.2. Average recognition rates by 5 speakers using MAP with/without a fuzzy controller, using identical adaptation and testing data.

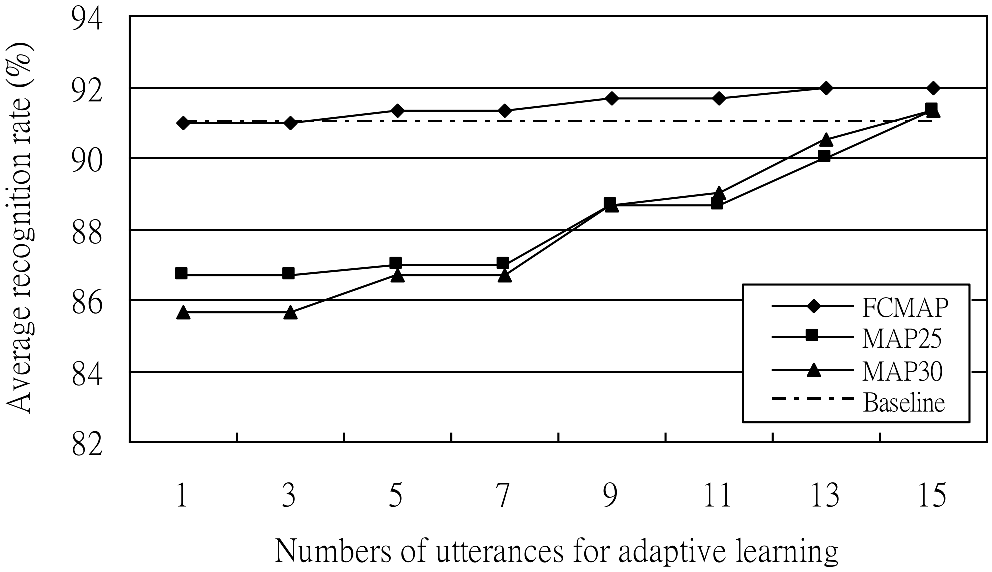


Fig. 4.3. Average recognition rates by 5 speakers using MAP with/without a fuzzy controller, using different adaptation and testing data.

When using a small number of utterances, say two utterances, for adaptation, the performance of MAP adaptation with various τ is shown in Fig. 4.4. It is seen that increasingτ tends to improve the performance. Fig. 4.5 shows the performance of MAP adaptation with various τ for a big number of adaptation utterances (thirty utterances), where the tendency that increasing the value of τ will cause the declining of recognition rate is observed. These further justify the rationale behind the design of FCMAP adaptation.

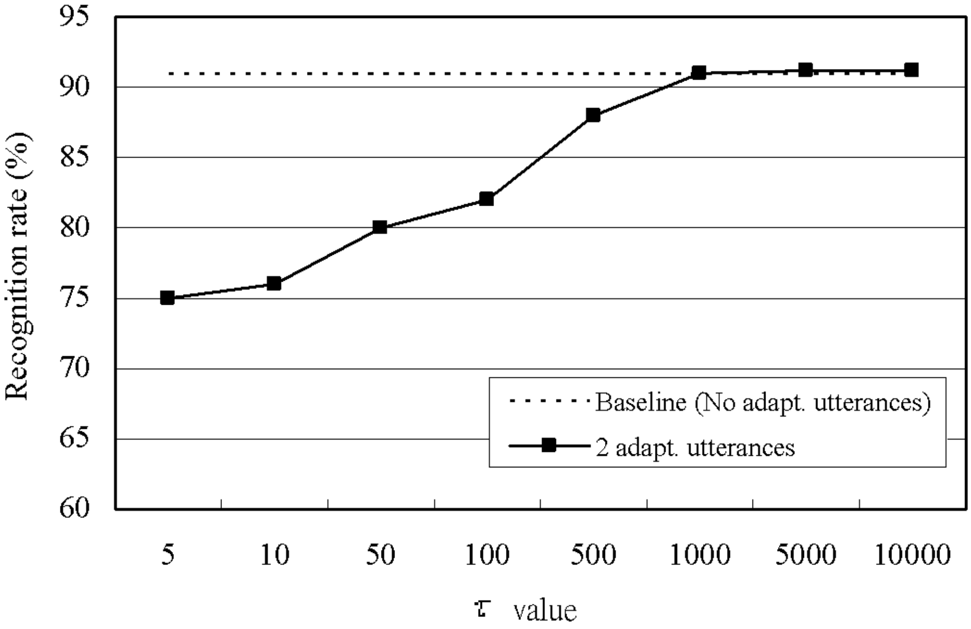


Fig. 4.4. The number of adaptation utterances = 2 (MAP testing experiments).

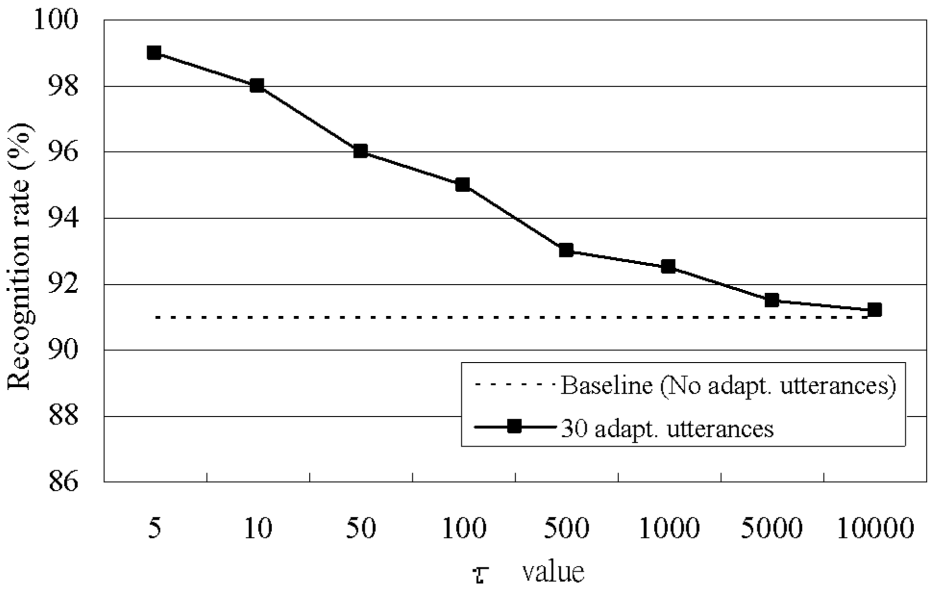


Fig. 4.5. The number of adaptation utterances = 30 (MAP testing experiments).

Compared with the conventional MAP estimate, the computing overhead of the FCMAP adaptation in calculating a proper value of τ is regarded as trivial, as is explained below.

For $N < N_1$, $\tau = \dfrac{M_1(N) f_1(N)}{M_1(N)} = f_1(N) = \dfrac{b}{\log(N)+a}$, which requires a division and a logarithm operation.

For the case when $N > N_3$, $\tau = \dfrac{M_3(N) f_3(N)}{M_3(N)} = f_3(N) = \dfrac{\log(N)}{N}$, which requires the same one division and one logarithm operation.

For the cases $N_1 < N < N_2$ and $N_2 < N < N_3$, the calculation of τ are respectively as follows:

$$\tau = \frac{M_1(N) f_1(N) + M_2(N) f_2(N)}{M_1(N) + M_2(N)}$$

$$= \frac{(N_2 - N)\dfrac{b}{\log(N)+a} + (N - N_1)\dfrac{N}{c}}{N_2 - N_1},$$

requiring two multiplications, three divisions and one logarithm operation;

and

$$\begin{aligned}\tau &= \frac{M_2(N) f_2(N) + M_3(N) f_3(N)}{M_2(N) + M_3(N)} \\ &= \frac{(N_3 - N)\dfrac{N}{c} + (N - N_2)\dfrac{\log(N)}{N}}{N_3 - N_2},\end{aligned}$$

again requiring the same amount of computation as in the former case.

Chapter 5

Enhancement of VFS Speaker Adaptation by Fuzzy Logic Control

As mentioned in Section 2.2.1, VFS adaptation is a post-processing after MAP-adaptation. MAP-adaptation, as aforementioned, 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. 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. 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.

As has been noted, the quality of MAP-adaptation depends largely upon the number of utterances acquired from adapting 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. The scheme for VFS adaptation proposed in this chapter works in the same line of thought by regulating the adaptation according to the amount of adaptation data.

As shown in Eq. (2-17), MAP adaptation is essentially a weighted average of the prior mean and the sample mean,

$$\hat{\mu}_k = \frac{N_k}{\tau + N_k}\bar{y}_k + \frac{\tau}{\tau + N_k}\mu_k . \tag{2-17}$$

And the idea of “collateral adaptation” behind VFS adaptation is best illustrated by Fig. 5.1, where the mean vector that isn’t adapted, μ_j, has three MAP-adapted neighbors μ_1, μ_2 and μ_3 in its vicinity of radius *R* denoted by $N_R(j)$. As shown in Eqs. (2-18) and (2-19), 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.

$$v_k = \hat{\mu}_k - \mu_k , \tag{2-18}$$

$$v_j = \frac{\sum_{k \in N(j)}^{K} \lambda_{j,k} \cdot v_k}{\sum_{k \in N(j)}^{K} \lambda_{j,k}} ,$$

$$\lambda_{j,k} = \exp\left(\frac{-d_{j,k}}{f}\right), \tag{2-19}$$

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

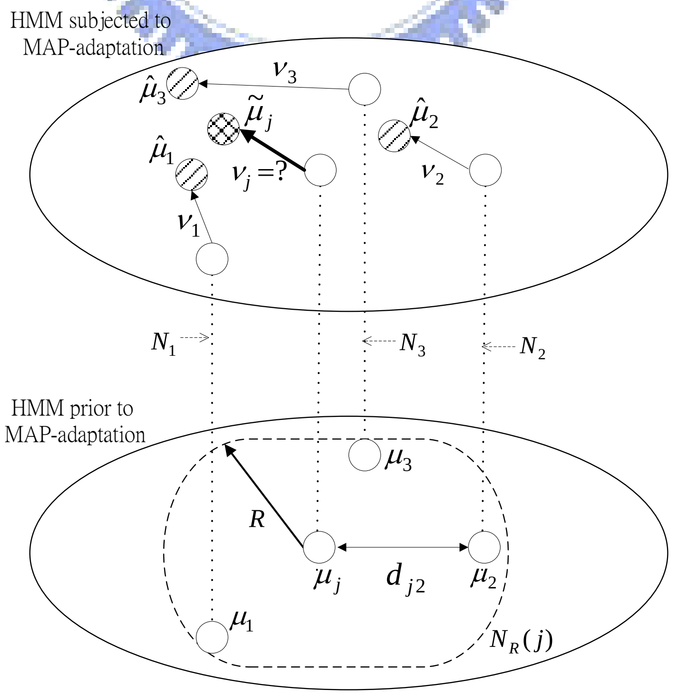


Fig. 5.1. Rationale behind VFS adaptation.

The *K*-nearest adapted neighbors μ_k's participating the estimation of ν_j can be selected purely geometrically by Euclidean distance d_{jk} in the mean vector space or more elaborated by choosing from a cluster-structure in which acoustic relations among parameter vectors are established.

Nevertheless, one key issue has been neglected in precedent VFS schemes [62-65] from the point of view regarding the quality of the transfer vectors ν_k's. Considering a specific mean vector μ_k adapted by very little adaptation samples N_k and yet very close to μ_j, in which case the associated ν_k very likely would be unreliable and accompanied by a significant weighting $\lambda_{j,k}$, and thus degrades the estimate of ν_j, as can be readily seen in Eqs. (2-18) and (2-19).

Therefore, 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 ν_k in the following way:

(1) When the transfer vector ν_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 ν_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 Eqs. (2-18) and (2-19), 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.

This is where fuzzy methodology comes into play, and how the statements of linguistic terms with uncertainty to some degree can be formulated in quantized forms for subsequent computations will be explained in the next subsection.

5.1 FLC-VFS Adaptation

For the specific problem in this work of 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 Fig. 5.2.

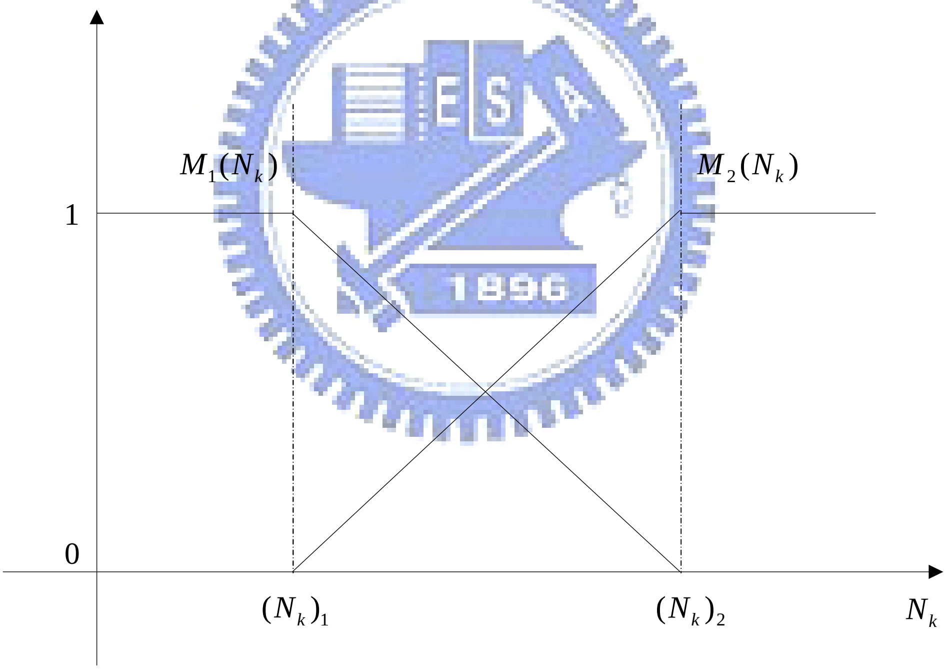


Fig. 5.2. Membership functions of the FLC for FLC-VFS adaptation.

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 $f = g_1(N_k)$, Rule 2: If N_k is $M_2(N_k)$, then $f = g_2(N_k)$,

where

$$M_1(N_k) = \begin{cases} 1 & N_k < (N_k)_1, \\ \dfrac{(N_k)_2 - N_k}{(N_k)_2 - (N_k)_1} & (N_k)_1 \le N_k \le (N_k)_2, \\ 0 & N_k > (N_k)_2, \end{cases}$$

$$M_2(N_k) = \begin{cases} 0 & N_k < (N_k)_1, \\ \dfrac{N_k - (N_k)_1}{(N_k)_2 - (N_k)_1} & (N_k)_1 \le N_k \le (N_k)_2, \\ 1 & N_k > (N_k)_2, \end{cases}$$

along with the implication functions

$$g_1(N_k) = a_1 \cdot N_k + b_1,$$

$$g_2(N_k) = a_2 \cdot N_k + b_2,$$

and the final system output as follows [113]

$$f = \frac{\sum_{i=1}^{2} M_i(N_k) \cdot g_i(N_k)}{\sum_{i=1}^{2} M_i(N_k)}. \tag{5-1}$$

Eq. (5-1) 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, $(N_k)_1$ and $(N_k)_2$) to be fixed. The following 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 $(N_k)_1 : (N_k)_2 = 1:3$.

```
$a_1$ = initial value; $b_1$ = 0; $q$ = 0;
/* The symbol $q$ denotes the iterative index while fixing $a_1$ and $b_1$. */
$R^0$ = baseline_recognition_rate;
```

STEP 2: Estimate the parameters a_1 and b_1 under the condition $N_k < (N_k)_1$, where $M_1(N_k) = 1$, $M_2(N_k) = 0$, and

$$f = \frac{M_1(N_k) \cdot g_1(N_k)}{M_1(N_k)} = g_1(N_k) = a_1 \cdot N_k + b_1.$$

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

```
$a_1 += \Delta a_1$; $q$ ++;
/* The symbol $\Delta a_1$ denotes an increment of $a_1$. */
$R^q$ = speech_recognition( $f = a_1 \cdot N_k + b_1$, testing_utterances);
/* The function speech_recognition($\cdot$) 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 testing_utterances. */
if ($R^q > R^{q-1}$) /* Increasing $a_1$ */
    Repeat
    { $a_1 += \Delta a_1$; $q$ ++;
    $R^q$ = speech_recognition( $f = a_1 \cdot N_k + b_1$, testing_utterances);
    } while ($R^q > R^{q-1}$);
else /* Decreasing $a_1$ */
    Repeat
    { $a_1 -= \Delta a_1$; $q$ ++;
    $R^q$ = speech_recognition( $f = a_1 \cdot N_k + b_1$, testing_utterances);
    } while ($R^q > R^{q-1}$);
$b_1 += \Delta b_1$; $q$ ++;
/* The symbol $\Delta b_1$ denotes an increment of $b_1$. */
```

```
    $R^q$ = speech_recognition( $f = a_1 \cdot N_k + b_1$, testing_utterances);
    if ( $R^q > R^{q-1}$ ) /* Increasing $b_1$ */
        Repeat
        { $b_1 += \Delta b_1$; $q$ ++;
         $R^q$ = speech_recognition( $f = a_1 \cdot N_k + b_1$, testing_utterances);
        } while ( $R^q > R^{q-1}$ );
    else /* Decreasing $b_1$ */
        Repeat
        { $b_1 -= \Delta b_1$; $q$ ++;
         $R^q$ = speech_recognition( $f = a_1 \cdot N_k + b_1$, testing_utterances);
        } while ( $R^q > R^{q-1}$ );
    return $R^q$;
```

STEP 3: Estimate the parameters a_2 and b_2 under the condition $N_k > (N_k)_2$, where $M_1(N_k) = 0$, $M_2(N_k) = 1$, and

$$f = \frac{M_2(N_k) \cdot g_2(N_k)}{M_2(N_k)} = g_2(N_k) = a_2 \cdot N_k + b_2.$$

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 $(N_k)_1 \le N_k \le (N_k)_2$, where $M_1(N_k) = \frac{(N_k)_2 - N_k}{(N_k)_2 - (N_k)_1}$, $M_2(N_k) = \frac{N_k - (N_k)_1}{(N_k)_2 - (N_k)_1}$, and

$$f = \frac{M_1(N_k) \cdot g_1(N_k) + M_2(N_k) \cdot g_2(N_k)}{M_1(N_k) + M_2(N_k)}$$

$$= \frac{\frac{(N_k)_2 - N_k}{(N_k)_2 - (N_k)_1} \cdot (a_1 \cdot N_k + b_1) + \frac{N_k - (N_k)_1}{(N_k)_2 - (N_k)_1} \cdot (a_2 \cdot N_k + b_2)}{\frac{(N_k)_2 - N_k}{(N_k)_2 - (N_k)_1} + \frac{N_k - (N_k)_1}{(N_k)_2 - (N_k)_1}}$$

$$= \frac{((N_k)_2 - N_k) \cdot (a_1 \cdot N_k + b_1) + (N_k - (N_k)_1) \cdot (a_2 \cdot N_k + b_2)}{(N_k)_2 - (N_k)_1}.$$

Since a_1 and b_1, together with a_2 and b_2, have already been determined at STEP 2 and STEP 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 $(N_k)_1 : (N_k)_2 = 1:3$,

$$\delta = \frac{\left|R^q - R^*\right|}{R^*}, \text{ /* } R^* \text{: desired recognition rate */}$$

$$R^0 = R^q .$$

Repeat from STEP 2 until δ 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.

5.2 Experiments

Experiments with MAP-FLCVFS adaptation were conducted in order to compare the recognition performance with MAP-VFS adaptation when encountering different amounts of adaptation data, from scarce to ample. In addition, MAP adaptation was also carried out alone in the comparative experiment to be a baseline of MAP-VFS and MAP-FLCVFS adaptations.

5.2.1 Database and Experiment Design

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.

The MAT-2000 database [120] 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. The details of the establishment of the initial SI models as a set of HMM parameters are entirely the same as the aforementioned adaptation experiments in Section 4.2.1.

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; please refer to Table 5.1 in Section 5.2.2 for the performance.

The same 5 lots of adaptation data were used for adapting the SI models through the proposed MAP-FLCVFS adaptation to acquire 5 $\mathrm{SA_{FLC}}$ 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-dimentional feature vector consisting of 12 mel-cepstral and 12 delta-mel-cepstral components was extracted for each frame.

Table 5.1. Average recognition rates (%) by conventional MAP-VFS adaptation with various *f*.

f	**Average recognition rates (%)**					
	Numbers of utterances for adaptation					
	0	**5**	**10**	**20**	**30**	**100**
5	93.2	92.3	93.6	95.3	96.3	97.0
10	93.2	92.2	93.7	95.6	96.8	97.6
15	93.2	92.2	94.1	96.1	97.3	98.0
20	93.2	92.7	94.8	96.6	97.9	98.5
25	93.2	92.3	94.2	96.3	97.4	98.3
30	93.2	92.0	93.8	95.8	96.9	98.3
35	93.2	91.7	93.6	95.5	96.5	98.4
40	93.2	91.9	93.8	95.5	96.6	98.2
45	93.2	91.8	93.6	95.2	96.2	98.4
50	93.2	91.6	93.6	95.0	96.8	98.5

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 SA$_{\text{MAP}}$ 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 SA$_{\text{MAP}}$ models
- MAP-VFS with 5 SA models built in the training phase
- MAP-FLCVFS with 5 SA$_{\text{FLC}}$ models built in the training phase

5.2.2 Experiment Results

The recognition performances of the conventional VFS with various *f* were recorded in Table 5.1, from which it can be seen that the best choice of *f* for VFS-adaptation would be 20 in all cases. The experiment records of MAP, MAP-VFS and MAP-FLCVFS adaptation were shown in Fig. 5.3, 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$_{\text{MAP}}$ 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$_{\text{MAP}}$ 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.

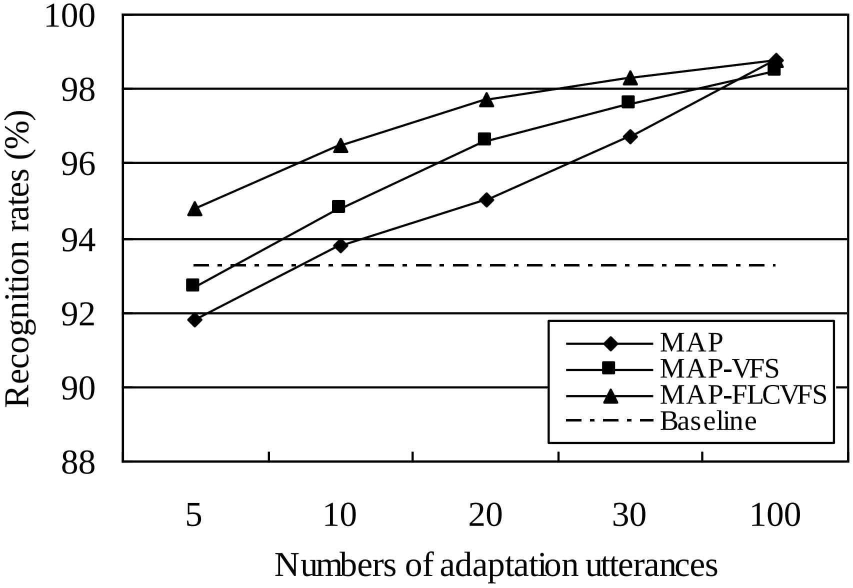


Fig. 5.3. Average recognition rates of MAP, MAP-VFS and MAP-FLCVFS with $f = 20$ for VFS and $\tau = 30$ for MAP.

The experiments of two extreme cases of adaptation data available, 10 utterances and 100 utterances, are also done for observing the variation of f on the recognition performance of VFS. In the case 10 utterances for adaptation, the performance of the VFS with various values of f is shown in Fig. 5.4. It is seen from Fig. 5.4 that the increasing f value tends to degrade the recognition performance when the training data are rare. On the other hand, when the training data are abundant, 100 utterances for instance, the increasing f tends to increase the recognition rate. As depicted in Fig. 5.5, the recognition rate is gradually increased when f is increased from 5 to 80, and approaches to a saturated value around 98.5 % when f is more than 80.

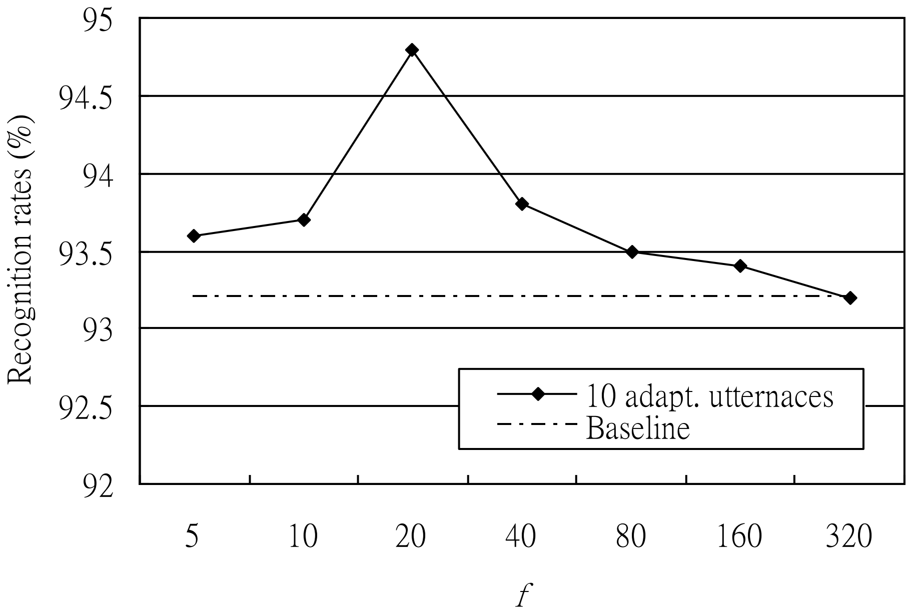


Fig. 5.4. Numbers of adaptation utterances = 10 (VFS testing experiments).

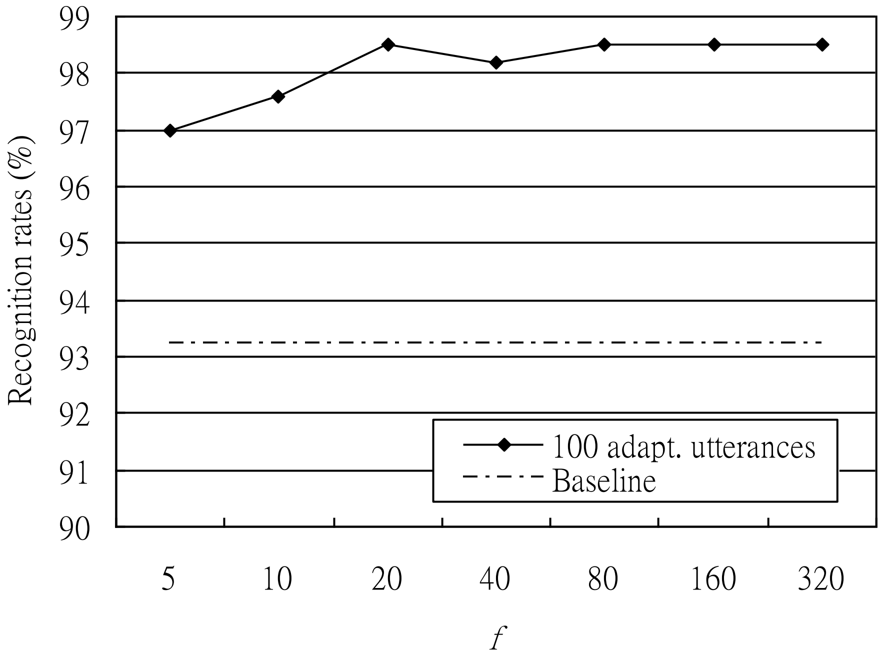


Fig. 5.5. Numbers of adaptation utterances = 100 (VFS testing experiments).

As the final observation, the computation overhead of FLC-VFS adaptation for calculating *f* compared to conventional VFS is practically minor, considering that at most 4 extra multiplications are required. The analysis is straightforward below:

For $N_k < (N_k)_1$, $f = a_1 \cdot N_k + b_1$ which takes only 1 multiplication, as is for the case when $N_k > (N_k)_2$, $f = a_2 \cdot N_k + b_2$.

And for the case $(N_k)_1 \le N_k \le (N_k)_2$,

$$\begin{aligned} f &= \frac{M_1(N_k) \cdot g_1(N_k) + M_2(N_k) \cdot g_2(N_k)}{M_1(N_k) + M_2(N_k)} \\ &= p \cdot (c_1 \cdot N_k{}^2 + c_2 \cdot N_k + c_3), \end{aligned}$$

which involves 4 multiplications.

Thus the computation of FLC-VFS adaptation is of the same order as that of conventional VFS adaptation.

Chapter 6

Incremental MLLR Speaker Adaptation by Fuzzy Logic Control

The transformation-based adaptation techniques have been described in Section 2.2.2, where Gaussians in an HMM system can be adapted rapidly by using the transformation matrix W_s. The MLLR speaker adaptation scheme proposed by Leggetter et al. [67] is the key role of such category of adaptation techniques and has been proven to be quite effective in many speech recognition applications. A series of variants of the MLLR scheme subsequently arise, aiming at the problem concerning the quality of the estimated transformation resulting from insufficient adaptation data. However, all those approaches for enhancing the robustness of MLLR are essentially complicated and time consuming in computation [68, 69, 70, 71, 73, 121], an adverse factor against on-line adaptation applications. The MAPLR adaptation scheme, for example, is a classic variant of MLLR adaptation, but it spends much more time in estimating the transformation matrix W_s than the MLLR adaptation scheme. 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 chapter 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 as much as the amount of training data allows, while the undesired effect of poor MLLR adaptation would be alleviated.

6.1 Incremental MLLR Adaptation (MAP-Like Adaptation)

As already mentioned, 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 is available, the quality of the acquired 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 [9]

$$\tilde{\mu}_s = \alpha \cdot \mu_s + (1-\alpha) \cdot W_s \cdot \xi_s, \quad 0 \le \alpha \le 1, \qquad (6\text{-}1)$$

where μ_s is the initial mean vector, ξ_S is the extended mean vector as defined in Eq. (2-23) and W_S is the transformation matrix derived from Eq. (2-24). The form of incremental MLLR adaptation in Eq. (6-1) is very similar to the one in Eq. (2-17), essentially an MAP-like adaptation. A weight parameter α is devised to govern the balance of the maximum likelihood estimate of the mean from the adaptation data and the prior mean, as is the role of τ in MAP estimate. By using a weighted sum of the initial mean vector and the MLLR adapted mean vector, it is expected that a satisfactory performance will also be achieved even when only a little amount of training data is available for adaptation. 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 goes with α approaching 1 so that $\tilde{\mu}_s$ 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 by W_s.

Fig. 6.1 depicts the training procedure of the incremental MLLR adaptation. The transformation matrix W_s from observed adaptation data can be obtained by using the standard MLLR method as described in Section 2.2.2. At the same time, according to the amount of data available for adaptation, the most proper α value for the incremental MLLR adaptation is to be determined by an FLC mechanism proposed in the subsequent sections.

Following the idea of incremental MLLR adaptation under α regulation, as formulated in Eq. (6-1), a precursory experiment was conducted to investigate the role of α against various amounts of adaptation data. A group of 15 speakers was recruited and every subject was asked to make 10 utterances, from which 1, 2, 4, 7 and 10 utterances were used to build 5 MLLR-adapted models for each speaker. For each speaker, the associated 5 MLLR-adapted HMM models were then put to recognition test with α varying from 0.1 to 0.9 at the interval of 0.1 respectively. Fig. 6.2 shows the procedure of the α-investigating experiment and the recognition rate for 15 speakers with various amounts of adaptation data (1, 2, 4, 7 and 10 utterances) and α settings (0.1 to 0.9) were tabulated in Table 6.1, where information regarding speaker 1 and 2 are explicitly stipulated. For a particular number of adaptation utterances (1, 2, 4, 7 or 10), the overall α^* for the 15 subjects is defined as [122]

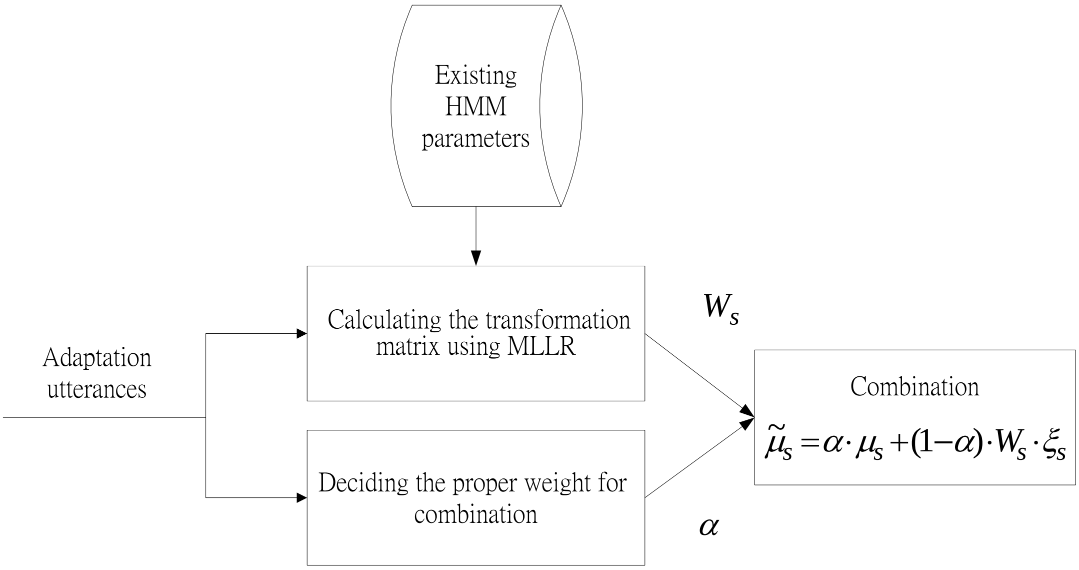
$$\alpha^* = \sum_{k=1}^{K} \sum_{j=1}^{9} w(k, j) \cdot \alpha(k, j), \tag{6-2}$$

$$w(k, j) = \frac{R(k, j)}{\sum_{k=1}^{K} \sum_{j=1}^{9} R(k, j)}, \tag{6-3}$$

where *k* is the speaker index and *j* is the index of α-setting, as explained in Fig. 6.2.

Table 6.1. α inclination w.r.t. the variation in the quantity of adaptation data in pursuit of recognition rate above baseline.

Speaker ID	**Utter. for MLLR**	**Recognition rates with various α settings**								
		0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
Speaker 1	**1**	82.5	84.3	86.0	87.8	91.5	91.5	91.5	91.0	90.5
	2	83.9	84.8	87.1	88.5	91.7	91.5	91.5	91.3	90.5
	4	91.1	91.1	92.3	93.2	93.1	92.1	91.1	90.9	90.5
	7	91.5	91.7	92.7	93.5	93.5	92.3	92.5	91.1	90.5
	10	93.3	93.3	93.3	93.8	93.6	92.5	91.3	91.1	90.5
Speaker 2	**1**	83.0	84.5	86.3	88.0	91.3	91.3	91.1	90.9	90.6
	2	84.2	85.0	86.8	88.9	91.7	91.7	91.3	91.0	90.5
	4	91.3	91.3	92.5	93.3	93.3	92.3	91.3	91.0	90.6
	7	91.7	91.8	92.8	93.6	93.5	92.6	91.7	91.2	90.5
	10	93.0	93.3	93.3	94.0	93.6	92.8	91.5	91.1	90.6
…	**…**	**…………………………………………**								
Speaker 15	…	……………………………………………								



Existing HMM parameters
Calculating the transformation matrix using MLLR
Adaptation utterances
Deciding the proper weight for combination
W_s
α
Combination
$\tilde{\mu}_s = \alpha \cdot \mu_s + (1-\alpha) \cdot W_s \cdot \xi_s$

Fig. 6.1. MLLR-adaptation under incremental α control (MAP-like adaptation).

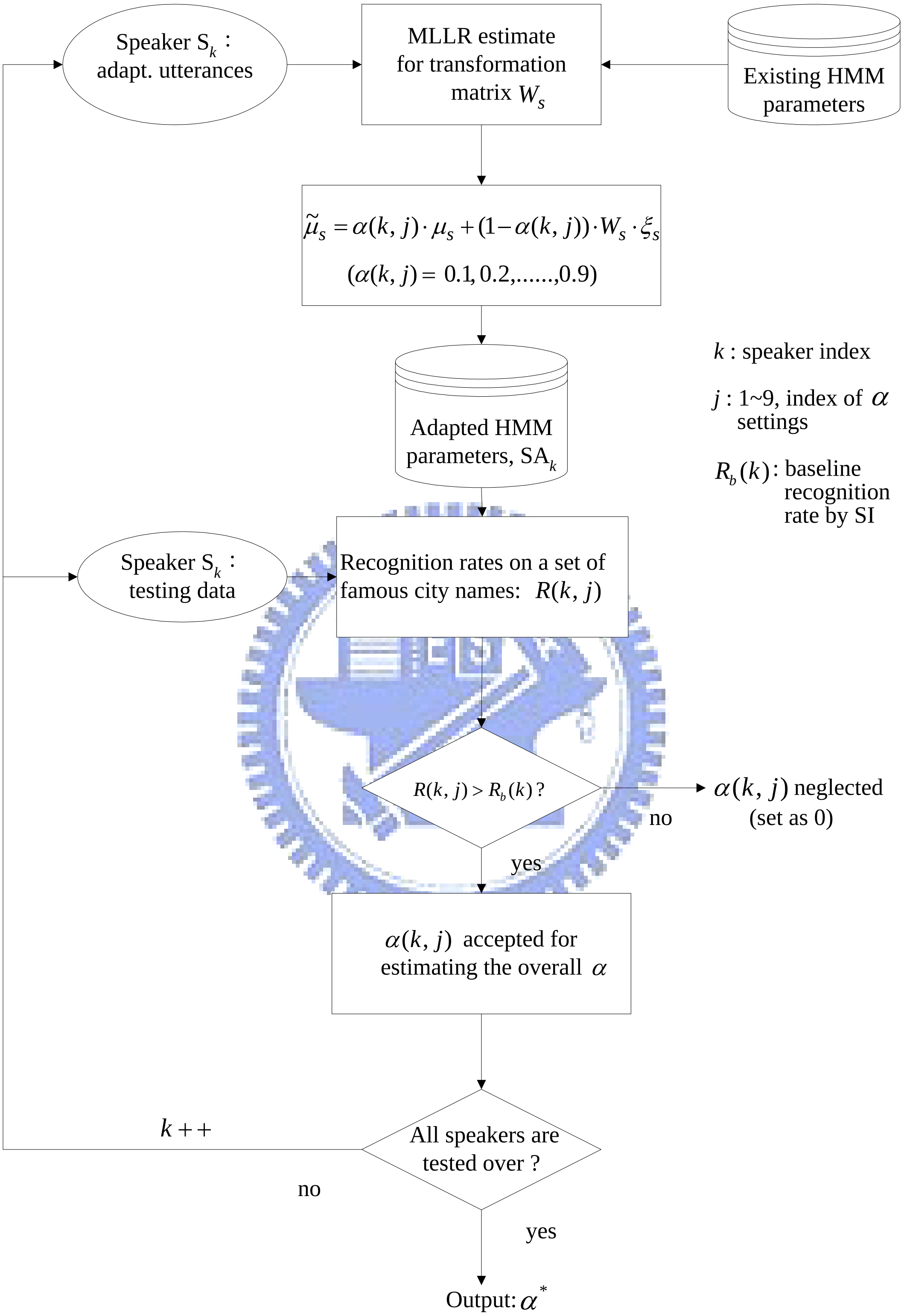

Speaker S$_k$: adapt. utterances
MLLR estimate for transformation matrix W_s
Existing HMM parameters
$\tilde{\mu}_s = \alpha(k,j)\cdot\mu_s + (1-\alpha(k,j))\cdot W_s\cdot\xi_s$
$(\alpha(k,j) =$ 0.1, 0.2,......,0.9)
k : speaker index
j : 1~9, index of α settings
$R_b(k)$: baseline recognition rate by SI
Adapted HMM parameters, SA$_k$
Speaker S$_k$: testing data
Recognition rates on a set of famous city names: $R(k,j)$
$R(k,j) > R_b(k)$?
no
$\alpha(k,j)$ neglected (set as 0)
yes
$\alpha(k,j)$ accepted for estimating the overall α
$k++$
All speakers are tested over ?
no
yes
Output: α^*

Fig. 6.2. An investigating procedure on the role of α in incremental MLLR adaptation given a specific amount of adaptation data.

It is noted in this preliminary investigation that in order to exceed the base-line recognition rate achieved by using SI models alone, the overall α^* value decreases from 0.6048 to 0.4515 when the number of adaptation utterances increases from 1 to 10, as shown in Fig. 6.3; specifically when the number of adaptation data is less than 4, the associated α^* is greater than 0.5, reflecting the potential inadequacy in the MLLR-estimated W_s.

In all, the investigation reveals that the key notion behind incremental MLLR-adaptation is intrinsically correct, which simply tells that in some way the effect of W_s estimate due to insufficient adaptation data should be held back such that the adapted HMM vectors do not drift away from the prior mean μ_s too much. Visualization of such a notion is illustrated in Fig. 6.4: which way to move and for how much.

The answer to the "Where to for $\tilde{\mu}_s$" question is conceptually plain: the more the adaptation data are, the smaller α will be. A formulation of the solution in the framework of fuzzy logic control is to be given in the following section.

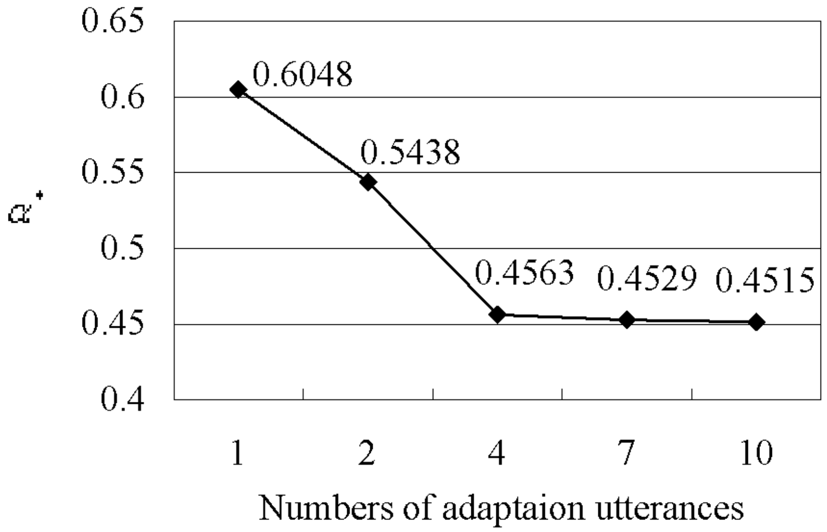


Fig. 6.3. α^* required for holding back W_s transformation effect under various numbers of adaptation utterances.

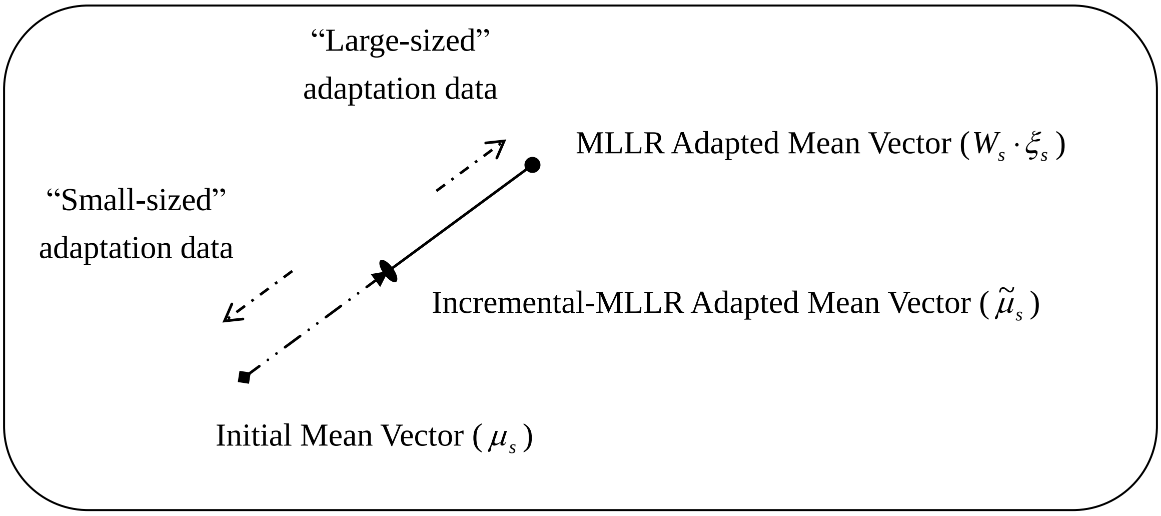

"Large-sized"
adaptation data
MLLR Adapted Mean Vector ($W_s \cdot \xi_s$)
"Small-sized"
adaptation data
Incremental-MLLR Adapted Mean Vector ($\tilde{\mu}_s$)
Initial Mean Vector (μ_s)

Fig. 6.4. Moving toward μ_s or $W_s \cdot \xi_s$? And for how much?

6.2 FLC-MLLR Adaptation (FCMAP-Like in Form)

The FLC-MLLR approach that performs incremental MLLR adaptation using fuzzy logic control will be presented herein. Formally speaking, FLC-MLLR is indeed an FCMAP-like adaptation in essence. The weight α in FLC-MLLR and the weight τ in FCMAP play the same role in respective adaptations, both being handled under associated FLC mechanisms according to the amount of adaptation data. According to the adaptation data size, each approach will adjust its weight parameter by the regulating FLC to move the adapted vector closer to the side of the maximum likelihood estimate of the mean vector or to the side of the initial mean vector.

A rule base with three fuzzy implications is given to govern α regulation under the circumstance of N training samples (in terms of acoustic frames) observed for all Gaussian mixture components [9] as follows.

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.

With Takagi-Sugeno FLC (T-S FLC) in consideration, 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. 6.5, 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

$$M_1(N) = \begin{cases} 1 & N \le N_1, \\ \dfrac{N_2 - N}{N_2 - N_1} & N_1 \le N \le N_2, \\ 0 & N \ge N_2, \end{cases}$$

$$M_2(N) = \begin{cases} 0 & N \le N_1 \text{ or } N \ge N_3, \\ \dfrac{N - N_1}{N_2 - N_1} & N_1 < N \le N_2, \\ \dfrac{N_3 - N}{N_3 - N_2} & N_2 \le N < N_3, \end{cases}$$

$$M_3(N) = \begin{cases} 0 & N \le N_2, \\ \dfrac{N - N_2}{N_3 - N_2} & N_2 < N < N_3, \\ 1 & N \ge N_3, \end{cases}$$

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 [113]

$$\alpha = \frac{\sum_{i=1}^{3} M_i(N) f_i(N)}{\sum_{i=1}^{3} M_i(N)}. \qquad (6\text{-}4)$$

By Eq. (6-4), 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 $M_3(N)$.

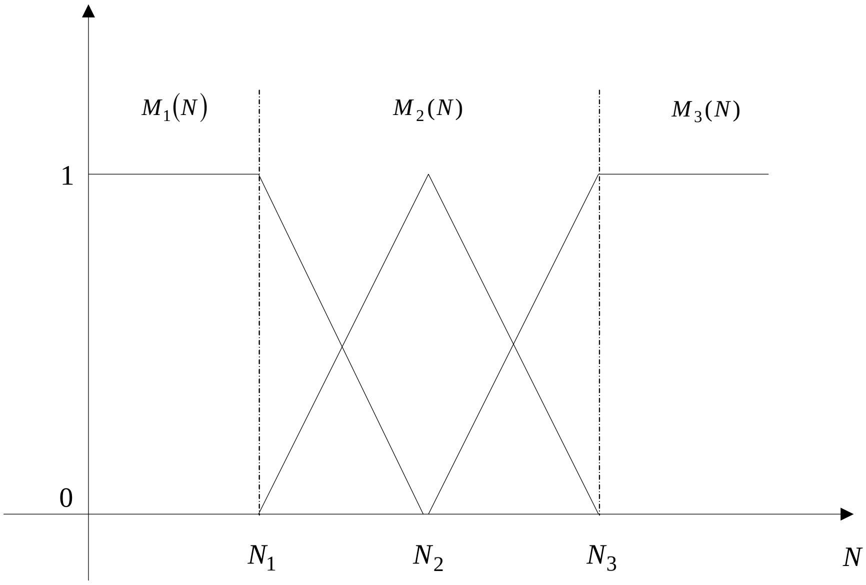


Fig. 6.5. Membership functions of the FLC for FLC-MLLR adaptation.

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.

```
$a_1$ = initial value;  $b_1$ = initial value; $k$ = 0;
$F^0$ = baseline_recognition_rate;
```

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:

```
$a_1 += \Delta a_1$; $k$ ++;
$F^k$ = 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 as 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 $M_1(N) = \frac{N_2 - N}{N_2 - N_1}$, $M_2(N) = \frac{N - N_1}{N_2 - N_1}$, $M_3(N) = 0$, and

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

With a_1 and b_1 already obtained at STEP 2, the parameters a_2 and b_2 is determined through the same tuning process as in STEP 2 for best recognition rate too.

STEP 5: Re-estimate the parameter N_3 under the condition $N_2 < N < N_3$, wherein

$$M_1(N) = 0, \quad M_2(N) = \frac{N_3 - N}{N_3 - N_2}, \quad M_3(N) = \frac{N - N_2}{N_3 - N_2}, \text{ and}$$

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

With a_2 and b_2 together with a_3 and b_3 already obtained at STEP 4 and STEP 3 respectively, a new value for N_3 can be found through a similar process for increasing recognition rates too.

STEP 6: Given the new estimate of N_3 from STEP 5, 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 can not further improve the recognition rate over the training data set.

6.3 Experiments

Experiments with FLC-MLLR adaptation were conducted in order to compare the recognition performance with MLLR- and MAPLR-adaptations when encountering different amounts of adaptation data, from scarce to ample.

6.3.1 Database and Experiment Design

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 the FLC (FLC-MLLR) in Section 6.2.

An 8 kHz sampling rate was set for speech signal acquisition. The analysis frames were 30-ms wide with a 20-ms overlap. For each frame, a 24-dimensional feature vector was extracted, which was made up of a 12-dimensional mel-cepstral vector and a 12-dimensional delta-mel-cepstral vector.

The initial models which were used as the speaker independent models were constructed using the database, MAT400 sub-database DB3 [119]. The details of the establishment of the initial SI models as a set of HMM parameters are entirely the same as the aforementioned adaptation experiments in Section 4.2.1.

The training data used for tuning the hyperparameters of the FLC were collected from 15 speakers in the training phase. 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 30 cities (two utterances for each) as FLC parameter tuning data; all utterances were recorded by an ordinary microphone. For readability and clearness, the training phase experiment procedure is described in the pseudo-code sequence below.

```
$\overline{F}^{0}$ = baseline recognition rate; $t$ = 0;
Repeat
        { $t$ ++;
        $\overline{F}_{2}^{t}$ = 2_utterances_training (SI_models, hyperparameters);
        $\overline{F}_{4}^{t}$ = 4_utterances_training (SI_models, hyperparameters);
        $\overline{F}_{6}^{t}$ = 6_utterances_training (SI_models, hyperparameters);
        $\overline{F}_{8}^{t}$ = 8_utterances_training (SI_models, hyperparameters);
        $\overline{F}_{10}^{t}$ = 10_utterances_training (SI_models, hyperparameters);
        $\bar{F}^{t} = \frac{\sum_{i=1}^{5} \bar{F}^{t}_{2\cdot i}}{5}$;

        $\Delta\bar{F}^{t} = \left|\bar{F}^{t} - \bar{F}^{t-1}\right|$;

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

where $2\cdot i$_utterances_training(·), $i =$ 1, 2, 3, 4, 5 is the procedure using $2\cdot i$ adaptation utterances from 15 speakers for fixing the 9 hyperparameters of FLC defined in Section 6.2 and thus returning a better-than-baseline overall recognition rate $\bar{F}^{t}_{2\cdot i}$ for the 15 training speakers, as is explained in the code-like sequence below.

```
$2\cdot i$_utterances_training (SI_models, hyperparameters) //  $i =$ 1, 2, 3, 4, 5.
{ $k = 0$;
  $\bar{F}^{0}_{2\cdot i}$ = baseline recognition rate;
  Repeat
        { $k$ ++;
          $\bar{F}^{k}_{(2\cdot i)1}$ = speaker_training (SI_models, test_data$_1$, hyperparameters,
                      $2\cdot i$_utterances$_1$);
                                    · · ·
          $\bar{F}^{k}_{(2\cdot i)j}$ = speaker_training (SI_models, test_data$_{j,}$ hyperparameters,
                      $2\cdot i$_utterances$_j$);
                                    · · ·
          $\bar{F}^{k}_{(2\cdot i)15}$ = speaker_training (SI_models, test_data$_{15,}$ hyperparameters,
                      $2\cdot i$_utterances$_{15}$);
```

```
                $\bar{F}_{2\cdot i}^{k} = \frac{\sum_{j=1}^{15} \bar{F}_{(2\cdot i)j}^{k}}{15}$;

                $\Delta\bar{F}_{2\cdot i} = \left|\bar{F}_{2\cdot i}^{k} - \bar{F}_{2\cdot i}^{k-1}\right|$;

        } *until* $\Delta\bar{F}_{2\cdot i}$ < threshold 1;

    *return* $\bar{F}_{2\cdot i}^{k}$;

    };
```

where $2\cdot i$ _utterances$_j$ and test_data$_j$ denote respectively the adaptation utterances in the number of 2, 4, 6, 8 and 10 for MLLR estimate of W_s and the 60 test utterances from the *j*th speaker, $1 \le j \le 15$, for the tuning of the 9 hyperparameters in the proposed FLC mechanism.

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 6.2, 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,…, 15.
    {
            Estimation_of_$W_s$ ($2\cdot i$ _utterances$_j$);

            $\bar{F}_{(2\cdot i)j}$ = Iterative_process (SI_models, test_data$_j$, $W_s$, hyperparameters);

            // as described in Section 6.2 for maximizing the recognition rate $\bar{F}_{(2\cdot i)j}$.

            return $\bar{F}_{(2\cdot i)j}$;

    };
```

As a result, a set of FLC hyperparameters {a_1, a_2, a_3, b_1, b_2, b_3, N_1, N_2 and N_3} was determined.

In the recognition phase, a group of 15 speakers that are entirely different from the previous group was recruited and again each being 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. For the recognition experiment with FLC-MLLR adaptation, five adapted models were constructed using 2, 4, 6, 8 and 10 adaptation utterances from each of the 15 speakers, and the α for each of the 5 adaptation will be calculated by Eq. (6-4) with $N_{utterances}$ = 2, 4, 6, 8 or 10 and the FLC hyperparameters were already determined in the training phase. 5 MLLR-adapted and 5 MAPLR-adapted models respectively using 2, 4, 6, 8 and 10 adaptation utterances were also constructed for performance comparison. Then 60 utterances from each of the 15 speakers were fed into the five adapted models for respective recognition rate evaluation.

6.3.2 Experiment Results

During the training phase, some experiment results and observations were acquired. It is observed that the weight α decreases as the number of adaptation utterances increases. As depicted in Fig. 6.6, α drops by a noticeable step when the number of utterances increases from 2 to 4, and then declines gradually, somewhat stabilized, as the number of utterances increases further.

It is seen that the tendency of the curve of α in FLC-MLLR is very similar to the one derived from the precursory investigation (Fig. 6.3). Both decrease quickly before the number of adaptation utterances reaches 4 and then fall progressively to a stable value around 0.47 when more and more adaptation utterances are available.

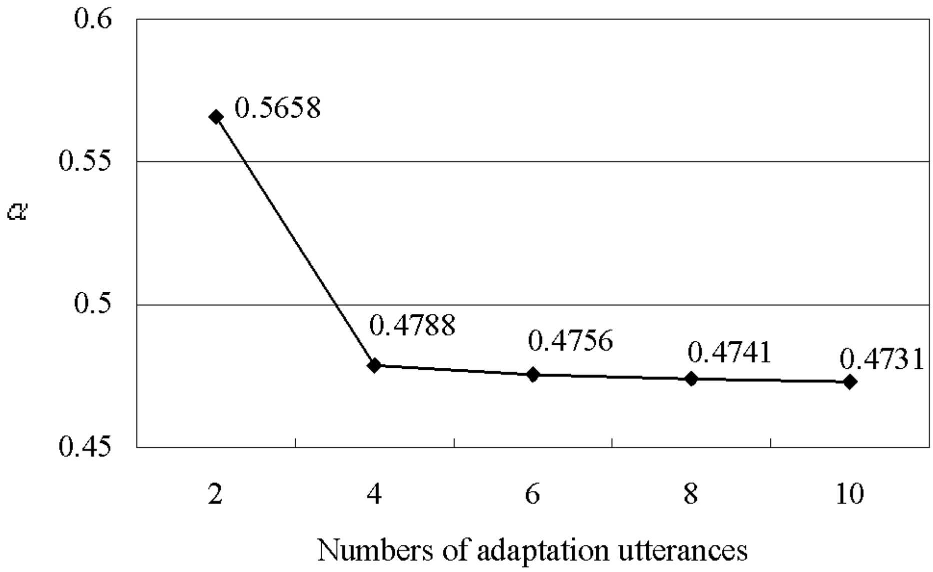


Fig. 6.6. The curve of the training values of α in FLC-MLLR adaptation.

In addition, 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. 6.7, it is observed that FLC-MLLR is better than MAPLR and MLLR for all cases, especially when training data are quite limited. It is also worth noting that the performance of MLLR falls below the baseline when 2 utterances were available for adaptation, indicating an improper adaptation may be worse than none at all. All three methods demonstrate improved recognition rate, and MAPLR tends to catch up FLC-MLLR, when the amount of training data increases.

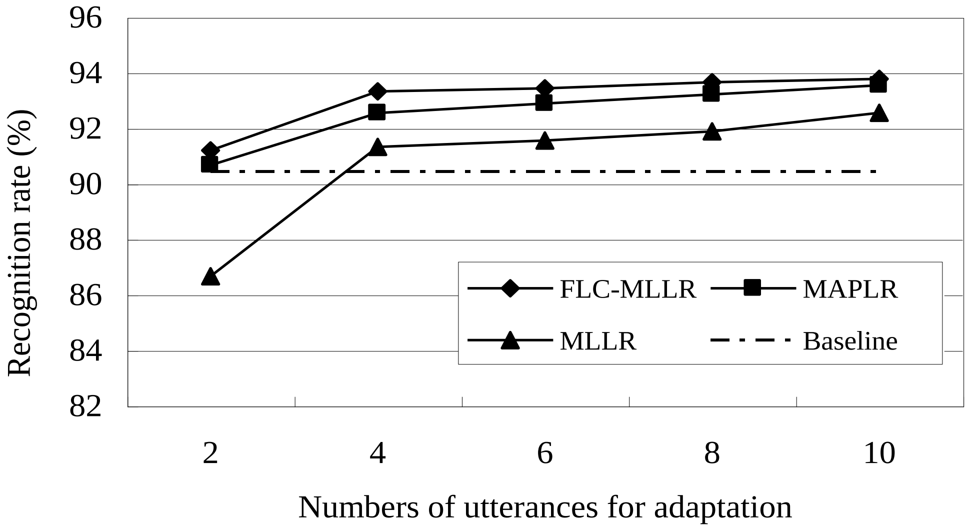


Fig. 6.7. The performance curves of FLC-MLLR, MAPLR and conventional MLLR in the recognition testing experiments with different amount of adaptation data.

Finally, the effects of α variation on the recognition performance of MLLR under extreme cases of training data availability are also observed, as shown in Fig. 6.8 and Fig. 6.9 respectively. The former shows that while the training data are scarce, 2 utterances say, the performance would go below the baseline if, for α being a bit 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 and further the performance degrades as if 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. 6.9.

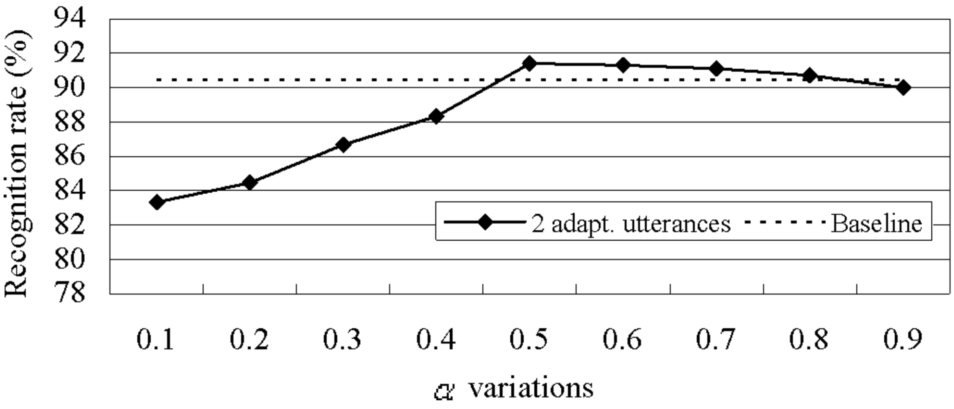


Fig. 6.8. Numbers of adaptation utterances = 2 (MLLR testing experiments).

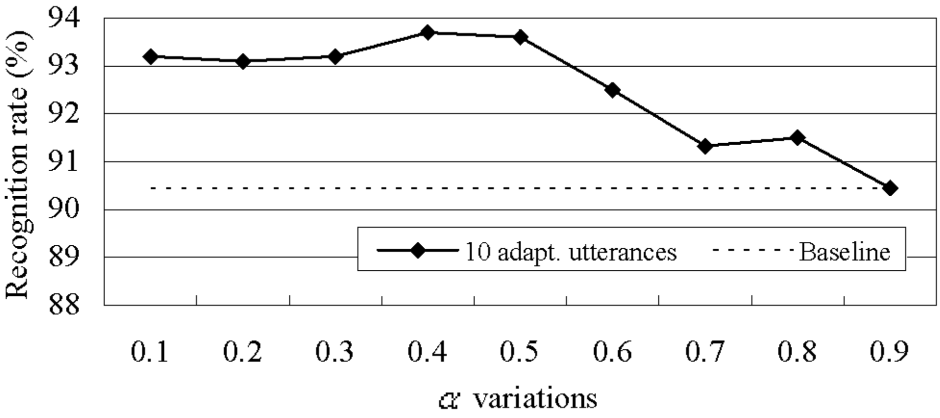


Fig. 6.9. Numbers of adaptation utterances = 10 (MLLR testing experiments).

As the final observation, the computing cost for FLC-MLLR involves the computation of α and W_s. Computing W_s is the same as in standard MLLR estimate.

The overhead of finding α in terms of the number of multiplications can be analyzed through its computation defined by Eq. (6-4).

For $N_1 < N < N_2$,

$$\begin{aligned}\alpha &= \frac{M_1(N) f_1(N) + M_2(N) f_2(N)}{M_1(N) + M_2(N)} \\ &= \frac{N^2(a_2 - a_1) + N(a_1 N_2 - a_2 N_1 + b_2 - b_1) + b_1 N_2 - b_2 N_1}{N_2 - N_1} \\ &= p \cdot (c_1 N^2 + c_2 N + c_3),\end{aligned}$$

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

$$\begin{aligned}\alpha &= \frac{M_2(N) f_2(N) + M_3(N) f_3(N)}{M_2(N) + M_3(N)} \\ &= \frac{N^2(a_3 - a_2) + N(a_2 N_3 - a_3 N_2 + b_3 - b_2) + b_2 N_3 - b_3 N_2}{N_3 - N_2} \\ &= q \cdot (d_1 N^2 + d_2 N + d_3).\end{aligned}$$

For $N < N_1$, $\alpha = a_1 \cdot N + b_1$ which requires 1 multiplication, as is for the case when $N > N_3$, $\alpha = a_3 \cdot N + b_3$.

Thus the computation of Eq. (6-1) is of the same order as computing Eq. (2-22), given that W_s being estimated by MLLR.

6.4 Fuzzy Mechanisms for the Context of Multiple Regression Classes

Whenever appropriate, the acoustic model space can be partitioned into a number of subspaces, each being a base class as referred in related works. In such a context, a transformation matrix is to be derived for each base class if in-class adaptation data are available such that a component in the class can be adapted accordingly.

Gales proposed a fuzzy clustering scheme [111] for determining the weight γ_p in the adaptation below, which is essentially a linear combination of MLLR transformation by matrices associated with every regression class

$$\hat{\mu}_s = \left[\sum_{p=1}^{P} \gamma_p \cdot W_s^{(p)} \right] \cdot \xi_s \,, \qquad (6\text{-}5)$$

where γ_p represents the degree of how much ξ_s belongs to the regression class *p*. Note that the role and purpose of the fuzzy techniques in Gales's work is completely different from the FLC mechanism for tuning α in FLC-MLLR herein.

Interestingly enough, Eq. (6-5) could be extended as

$$\tilde{\mu}_s = \alpha \cdot \mu_s + (1-\alpha) \cdot \left[\sum_{p=1}^{P} \gamma_p \cdot W_s^{(p)} \right] \cdot \xi_s \,, \qquad (6\text{-}6)$$

which could reduce to the form of Eq. (6-1) in the context of one regression class adaptation (i.e., *p* = 1), as is the case considered in the dissertation.

Chapter 7

Audio Event Detection Using Variable-Length Decision Windows

Detecting female screaming in three environments of different acoustic backgrounds was exploited in the research to examine the behavior of an FLC-regulating mechanism embedded in an audio event detection system for decision window length control.

A typical process for audio event detection would feed the stream of audio frames (vectors of extracted acoustic features, that is) into the event classifier by which successive analysis on a pre-determined number of audio frames is conducted and then the decision as to whether an audio event being detected over the associated time span, so called the decision window DW as mentioned in Section 2.3.2, is made. Fig. 7.1 depicts a stream of fixed-length decision windows, each of which covers the same number of audio frames and is thus of the same time span.

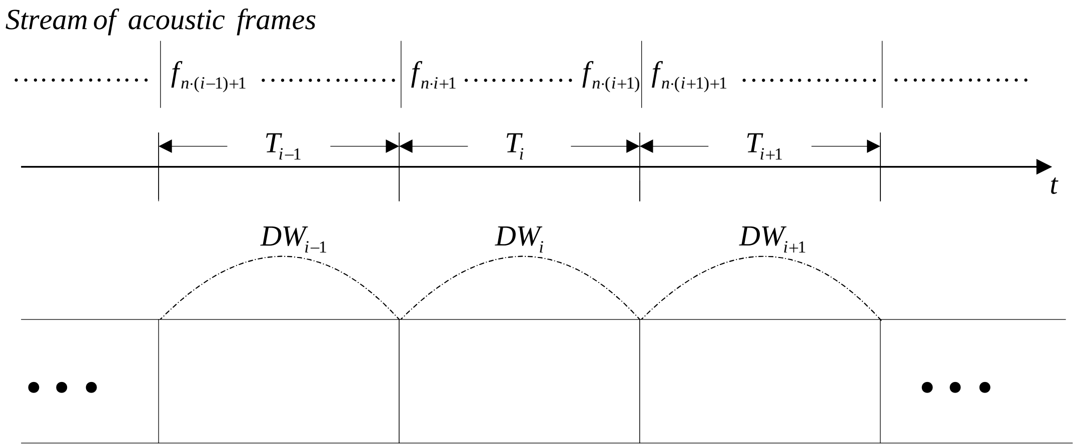


Fig. 7.1. DW with fixed-length, each covering the same number of audio frames, n, over the time span.

As is clearly seen in Fig. 7.1, for a fixed-length DW covering n audio frames of Δt ms time interval, the process makes a decision of event detection every $n \cdot \Delta t$ ms, regardless of the auditory situation in the context, which may be calm or tense. A too-long DW might face the concern of real-time response, which is essential to all surveillance and security applications, whereas a too-short one would instead encounter the problem of false alarms against sudden/intermittent acoustic changes in the background, which is equally undesired either.

7.1 Concepts of Short Timeslot Likelihood Difference (*STLD*)

The idea of variable-sized DW thus arises and is the core of the proposed audio event detection system in this dissertation. The length of the decision window should be small when encountering a somewhat "aurally hot" situation so that decision of event detection could be undertaken at a higher rate and be stretched at "aurally calm" moments for collecting more audio frames to ensure the reliability and correctness of the detection results. Such a situation-dependent behavior is essential to application where reliable and real-time response is the major concern, for which the fixed-length decision window may not suffice. An FLC mechanism is conceived for this purpose. The control of the decision window size is governed by an FLC, adjusting the window size by estimating the difference of likelihood scores between targeted audio event and normal acoustic background models over a short time-span. The design of the proposed variable-sized DW in audio event detection will be described in detail in the following sections.

An index *STLD* (Short Timeslot Likelihood Difference) for governing the length of the decision window in the case of two sound models is devised as follows:

$$STLD = \left| \sum_{i=1}^{m} \log f(x_i \mid \lambda_1) - \sum_{i=1}^{m} \log f(x_i \mid \lambda_2) \right| , \qquad (7\text{-}1)$$

where λ_1 and λ_2 are the sound models in consideration, $f(x_i \mid \lambda_1)$ and $f(x_i \mid \lambda_2)$ are given by Eq. (2-34), representing the likelihood of λ_1 and λ_2 model classification, respectively, for frame x_i.

The rationale behind Eq. (7-1) is that at the beginning stage covering *m* frames, say, of a decision window, if the class inclination of the frames has clearly exhibited, one term in Eq. (7-1) will be substantially greater than the other. As a consequence, a salient *STLD* value is acquired, indicating that a narrow decision window would suffice. If the class of the *m* frames can not be resolved, both terms in Eq. (7-1) would be trivial and lead to an insignificant *STLD* implying the need of a wider DW in order to collect more frames for classification. Fig. 7.2 illustrates the "phenomenon" implicated by Eq. (7-1).

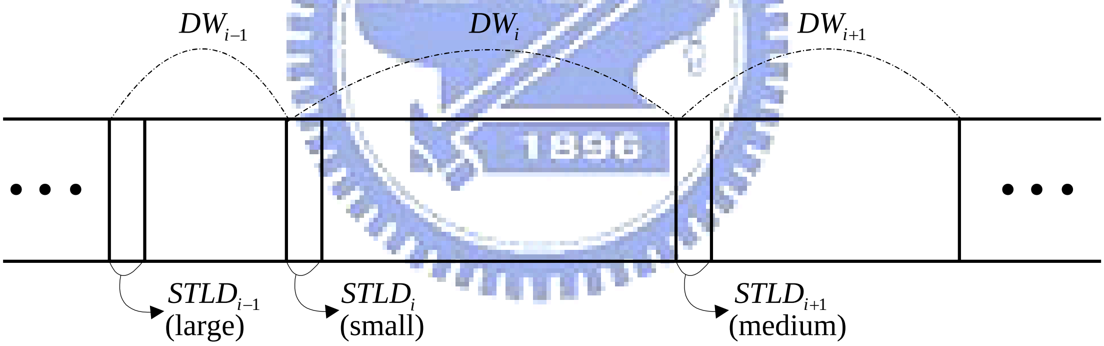


Fig. 7.2. DWs with variable length governed by *STLD* (Short Timeslot Likelihood Difference) indices.

7.2 Decision Windows Governed by an *STLD*-Driven FLC

As already explained, the *STLD* index can be used as the key to DW size control and, as a result, an FLC dictated by two IF-THEN fuzzy rules is designed accordingly:

Rule 1: If *STLD* is small,

Then *WL* is big,

Rule 2: If $STLD$ is big,

Then *WL* is small,

where $STLD$ is the input for the FLC and *WL*, the window length, is the output of the FLC.

Quantitatively, the FLC rule set is transformed into

Rule 1: If $STLD$ is $M_1(STLD)$,

Then $WL_B = f_1(STLD)$,

Rule 2: If $STLD$ is $M_2(STLD)$,

Then $WL_S = f_2(STLD)$, (7-2)

where

$$WL = \frac{\sum_{i=1}^{2} M_i(STLD) \cdot f_i(STLD)}{\sum_{i=1}^{2} M_i(STLD)}, \tag{7-3}$$

$$M_1(STLD) = \begin{cases} 1 & STLD \le STLD_1, \\ \dfrac{STLD_2 - STLD}{STLD_2 - STLD_1} & STLD_1 < STLD < STLD_2, \\ 0 & STLD \ge STLD_2, \end{cases} \tag{7-4}$$

$$M_2(STLD) = \begin{cases} 0 & STLD \le STLD_1, \\ \dfrac{STLD - STLD_1}{STLD_2 - STLD_1} & STLD_1 < STLD < STLD_2, \\ 1 & STLD \ge STLD_2, \end{cases} \tag{7-5}$$

$$f_1(STLD) = a_1 \cdot STLD + b_1, \tag{7-6}$$

$$f_2(STLD) = a_2 \cdot STLD + b_2. \tag{7-7}$$

In the formulation, $M_1(\cdot)$ and $M_2(\cdot)$ are membership functions of $STLD$, as shown in Fig. 7.3, and WL, the DW length to be determined by the $STLD$-controlled FLC, is a weighted sum of $f_1(\cdot)$ and $f_2(\cdot)$. It is observed from

Eqs. (7-3), (7-4) and (7-5) that for $STLD < STLD_1$, WL is solely determined by $f_1(\cdot)$, simply the case of Rule 1; whereas for $STLD > STLD_2$, WL is determined by $f_2(\cdot)$ alone, as is the case of Rule 2.

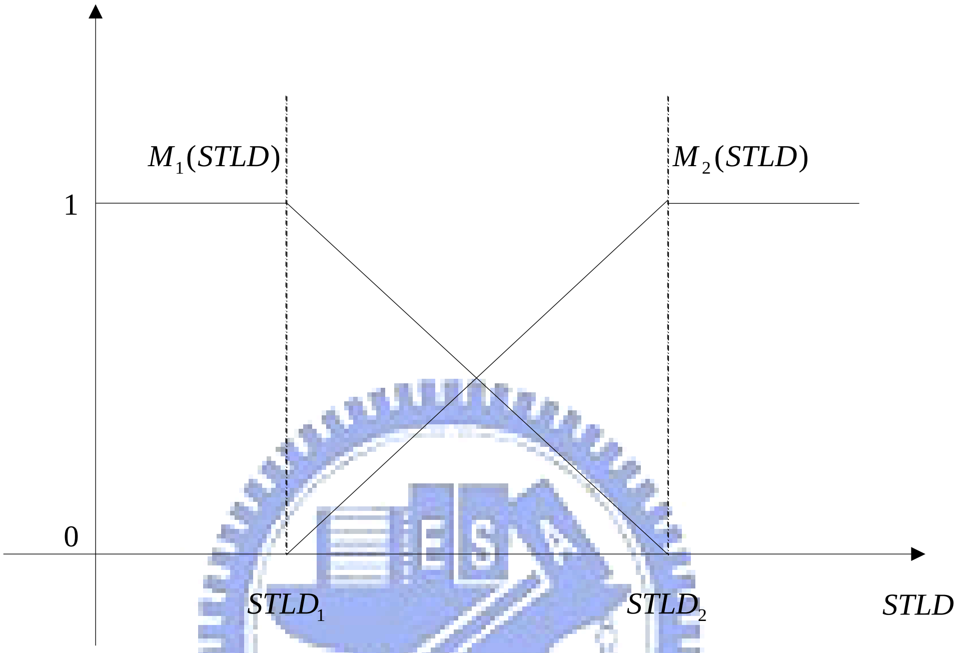


Fig. 7.3. Membership functions of the *STLD*-driven FLC.

The FLC now has six hyper-parameters (a_1, a_2, b_1, b_2, $STLD_1$ and $STLD_2$) to be fixed, for which an iterative process is devised as follows

STEP 1: Let $STLD_1 : STLD_2 = 1:3$ and give an initial value to $STLD_1$ in the experiment.

a_1 = initial value; b_1 = 0; k = 0;

F^0 = event_detection_ rate($WL = a_1 \cdot STLD + b_1$, training_database);

STEP 2: Estimate the parameters a_1 and b_1 under the condition $STLD < STLD_1$, wherein $M_1(STLD) = 1$, $M_2(STLD) = 0$, and

$$WL = \frac{M_1(STLD) \cdot f_1(STLD)}{M_1(STLD)} = f_1(STLD) = a_1 \cdot STLD + b_1,$$

by using the following pseudo-code sequence:

```
$a_1 += \Delta a_1$; $k$++;
$F^k$ = event_detection_ rate($WL = a_1 \cdot STLD + b_1$, training_database);
**if** ($F^k > F^{k-1}$)
        *Repeat*
         { $a_1 += \Delta a_1$; $k$++;
          $F^k$ = event_detection_ rate($WL = a_1 \cdot STLD + b_1$, training_database);
         } *while* ($F^k > F^{k-1}$);
**else**
        *Repeat*
         { $a_1 -= \Delta a_1$; $k$++;
          $F^k$ = event_detection_ rate($WL = a_1 \cdot STLD + b_1$, training_database);
         } *while* ($F^k > F^{k-1}$);
$b_1 += \Delta b_1$; $k$++;
$F^k$ = event_detection_ rate($WL = a_1 \cdot STLD + b_1$, training_database);
**if** ($F^k > F^{k-1}$)
        *Repeat*
         { $b_1 += \Delta b_1$; $k$++;
          $F^k$ = event_detection_ rate($WL = a_1 \cdot STLD + b_1$, training_database);
         } *while* ($F^k > F^{k-1}$);
**else**
        *Repeat*
         { $b_1 -= \Delta b_1$; $k$++;
          $F^k$ = event_detection_ rate($WL = a_1 \cdot STLD + b_1$, training_database);
         } *while* ($F^k > F^{k-1}$);
*return* $F^k$;
```

In the pseudo-code sequence, the rate of correct detection returned by event_detection_ rate(WL, X) is defined as

$$\text{detection rate} = \frac{\text{numbers of decision windows with correct detection}}{\text{numbers of all decision windows}} \times 100(\%). \qquad (7\text{-}8)$$

STEP 3: Estimate the parameters a_2 and b_2 under the condition $STLD > STLD_2$, wherein $M_1(STLD) = 0$, $M_2(STLD) = 1$, and

$$WL = \frac{M_2(STLD) \cdot f_2(STLD)}{M_2(STLD)} = f_2(STLD) = a_2 \cdot STLD + b_2,$$

by the same process as for a_1 and b_1.

STEP 4: Re-estimate the parameter $STLD_2$ for $STLD_1 < STLD < STLD_2$, wherein

$$M_1(STLD) = \frac{STLD_2 - STLD}{STLD_2 - STLD_1},$$

$$M_2(STLD) = \frac{STLD - STLD_1}{STLD_2 - STLD_1},$$

$$\begin{aligned} WL &= \frac{M_1(STLD) f_1(STLD) + M_2(STLD) f_2(STLD)}{M_1(STLD) + M_2(STLD)} \\ &= \frac{\frac{STLD_2 - STLD}{STLD_2 - STLD_1} \cdot (a_1 \cdot STLD + b_1) + \frac{STLD - STLD_1}{STLD_2 - STLD_1} \cdot (a_2 \cdot STLD + b_2)}{\frac{STLD_2 - STLD}{STLD_2 - STLD_1} + \frac{STLD - STLD_1}{STLD_2 - STLD_1}} \\ &= \frac{(STLD_2 - STLD)(a_1 \cdot STLD + b_1) + (STLD - STLD_1)(a_2 \cdot STLD + b_2)}{STLD_2 - STLD_1}, \end{aligned}$$

with a_1, b_1, a_2 and b_2 already fixed in STEP 2 and STEP 3, a new value for $STLD_2$ can be found through a similar tuning process as in STEP 3 for best recognition rate too.

STEP 5: Update $STLD_1$ such that $STLD_1 : STLD_2 = 1:3$. Repeat from STEP 2 until the settings of a_1, a_2, b_1, b_2, $STLD_1$ and $STLD_2$ can not further maximize the system performance over the training dataset.

7.3 Experiments

The experiments were to detect female screaming in three environments of different acoustic backgrounds: the office space, the parking lot and the living room.

7.3.1 Experiment Designs

In the training phase, three GMM models for “office space”, “parking lot” and “living room” were built as backgrounds using 10-minute recording in each environment. The recording was undertaken at 8K Hz sampling rate, from which LPC, LPCC and MFCC were extracted for each 20 ms frame (consisting of 160 samples, i.e.). Note that a 12-D LPC, a 12-D LPC/mel cepstrum and a 12-D delta cepstrum were utilized. Three GMM models for “female screaming” in each of the three environments were also built using two-thirds of a 180-second (60 sec. for each environment) recording from each of a group of 15 female subjects for extracting the same set of 3 acoustic features; the subjects were requested to scream in every possible way they could during the recording.

The rest one-third of the screaming data (20 sec. for each environment and totally 900 sec. for all 15 females in all the three environments) was used for FLC parameter-tuning as previously described.

In the event detection testing phase, an entirely new group of 15 females was recruited for the screaming recording of 60 sec. each (20 sec. for each of the three environments).

7.3.2 Experiment Results

During the testing phase, the GMM classifier with the proposed FLC-regulated DW was put to detect audio events occurring in a background audio stream of 15 minutes in length. Three experiments were conducted in “office space”, “parking lot” and “living room” respectively, and several observations on the effectiveness of the proposed approach are presented in tabulation for comparison, as are briefed below.

(1) Table 7.1 shows that, using LPC alone, the approach exploiting variable-sized DW governed by FLC achieves an average of 95%, 93.5% and 92% accuracy for event detection in the three testing contexts respectively, where the window size varies between $W_{\min}$ and $W_{\max}$, with an average of W_{avg}. With LPC alone, Table 7.2 shows the performance of the fixed-length DW scheme with a variety of fixed DW settings, from 0.5 sec. to 5 sec. at an increment of 0.5 sec., and in all cases the accuracy is inferior to the scores in Table 7.1. It is further noted that, against the variable-sized DW, the fixed DW reaches competitive scores of 91% at 3-sec. *WL*, 93.33% at 2.5-sec. *WL* and 95% at 1.5-sec. *WL*, respectively in the three testing contexts: the settings of DW fall within the corresponding ranges of DW variation $[W_{\min}, W_{\max}]$ associated with the FLC-regulated DW operation.

(2) Similar observations from the case of using LPCC alone are also made, as shown in Table 7.3 and 7.4.

(3) Table 7.5 and 7.6 present the experiment results in the case of using MFCC feature with the same observations.

(4) In the experiment, auditively the noisiest background is the living room (where family members exchanged conversation while children chasing/playing around, with TV set turned on aloud), followed by the parking lot, and then the office space. Such a phenomenon seems to be reflected by the range of *WL* variation, $WR = [W_{\min}, W_{\max}]$, when the $STLD$-driven FLC operated in the three contexts. To be specific,

$$WR\text{(office space)} < WR\text{(parking lot)} < WR\text{(living room)},$$

regardless of whichever of the three acoustic features used.

(5) For all the testing in the 3 backgrounds, MFCC leads to the best performance in audio event detection, LPCC the second and LPC the third, regardless of whichever control scheme on DW size being taken, as shown in Figs. 7.4, 7.5 and 7.6.

Table 7.1. Event detection by an FLC-regulated DW, using only LPC feature.

Variable-sized DW	**Living room**	**Parking lot**	**Office space**
$W_{min.}$	3.12 sec.	2.23 sec.	1.12 sec.
$W_{max.}$	3.96 sec.	2.88 sec.	1.58 sec.
$W_{avg.}$	3.55 sec.	2.56 sec.	1.33 sec.
Accuracy$_{avg.}$	92.00%	93.50%	95.00%

Table 7.2. Event detection by fixed-length DW, using only LPC feature.

DW length	**Living room**	**Parking lot**	**Office space**
0.5 sec.	80.83%	83.33%	91.67%
1 sec.	81.67%	86.00%	93.33%
1.5 sec.	84.00%	87.00%	95.00%
2 sec.	86.00%	91.33%	94.67%
2.5 sec.	89.33%	93.33%	95.00%
3 sec.	91.00%	93.00%	95.00%
5 sec.	91.67%	93.33%	95.00%
Average	86.36%	89.62%	94.24%

Table 7.3. Event detection by an FLC-regulated DW, using only LPCC feature.

Variable-sized DW	**Living room**	**Parking lot**	**Office space**
$W_{min.}$	3.18 sec.	2.31 sec.	1.15 sec.
$W_{max.}$	3.98 sec.	2.92 sec.	1.63 sec.
$W_{avg.}$	3.57 sec.	2.61 sec.	1.36 sec.
Accuracy$_{avg.}$	93.50%	95.00%	97.00%

Table 7.4. Event detection by fixed-length DW, using only LPCC feature.

DW length	**Living room**	**Parking lot**	**Office space**
0.5 sec.	83.67%	87.50%	92.50%
1 sec.	87.67%	90.00%	94.67%
1.5 sec.	89.00%	90.50%	96.50%
2 sec.	90.67%	93.33%	96.67%
2.5 sec.	91.67%	95.00%	96.67%
3 sec.	93.00%	95.00%	96.00%
5 sec.	93.33%	95.00%	96.67%
Average	89.86%	92.33%	95.67%

Table 7.5. Event detection by an FLC-regulated DW, using only MFCC feature.

Variable-sized DW	**Living room**	**Parking lot**	**Office space**
$W_{min.}$	3.15 sec.	2.18 sec.	1.17 sec.
$W_{max.}$	3.92 sec.	2.91 sec.	1.68 sec.
$W_{avg.}$	3.52 sec.	2.55 sec.	1.41 sec.
Accuracy$_{avg.}$	95.00%	98.50%	98.50%

Table 7.6. Event detection by fixed-length DW, using only MFCC feature.

DW length	Living room	Parking lot	Office space
0.5 sec.	84.50%	88.33%	93.33%
1 sec.	88.33%	90.67%	95.33%
1.5 sec.	90.50%	91.50%	98.00%
2 sec.	91.33%	94.67%	98.00%
2.5 sec.	92.50%	98.33%	98.33%
3 sec.	94.00%	98.00%	98.00%
5 sec.	95.00%	98.33%	98.33%
Average	90.88%	94.26%	97.05%

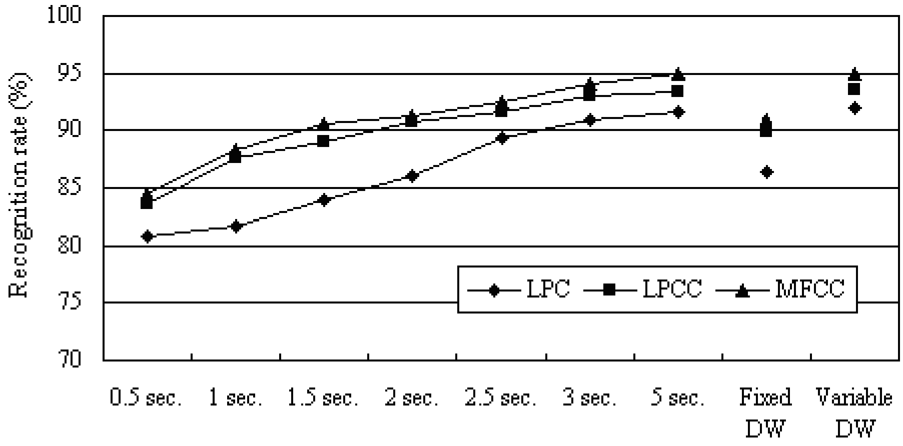


Fig. 7.4. Living room audio event detection.

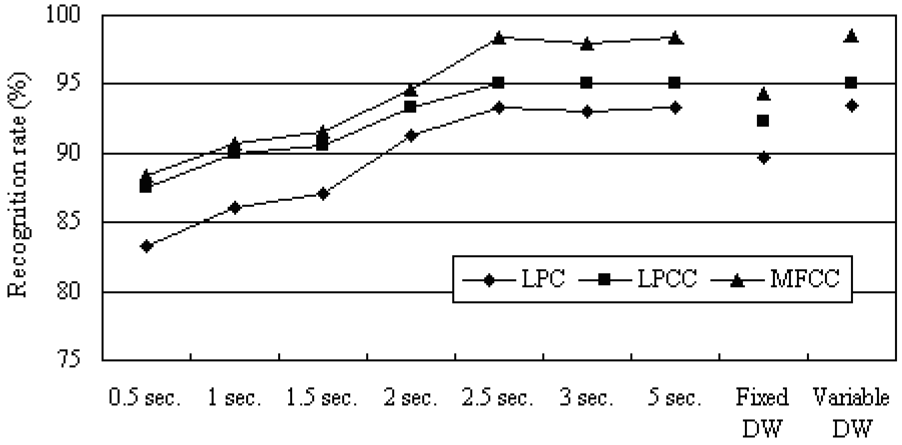


Fig. 7.5. Parking lot audio event detection.

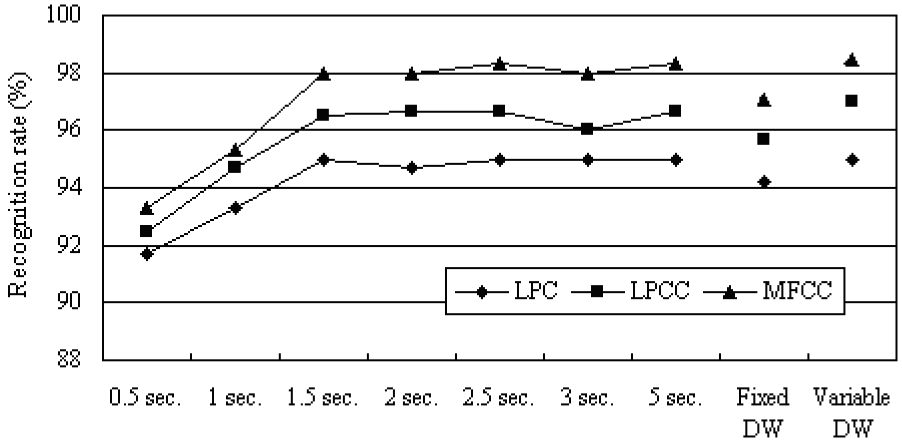


Fig. 7.6. Office space audio event detection.

Chapter 8

Conclusions and Future Works

In the following, the major contributions of the author’s work and some findings and observations of experiment results with FLC mechanisms are briefly summarized. In addition, some plausible developments in the future along the line of current researches are also mentioned.

8.1 HMM Speaker Adaptation with FLC

The quality of HMM speaker adaptation relies greatly on the amount of adaptation data acquired from the new speaker, be it an MAP or MLLR adaptation. It would be desired that the adaptation from either MAP or MLLR estimate to the prior distributions of HMM should be restricted when the adaptation data is limited, and adapts fully when the opposite occurs.

During the MAP estimate and the associated VFS process that follows up, the adaptation is governed by

$$\hat{\mu}_k = \frac{N_k}{\tau + N_k}\bar{y}_k + \frac{\tau}{\tau + N_k}\mu_k , \tag{2-17}$$

and

$$v_k = \hat{\mu}_k - \mu_k , \tag{2-18}$$

$$\tilde{\mu}_j = \frac{\sum_{k \in N(j)} \lambda_{j,k} \cdot v_k}{\sum_{k \in N(j)} \lambda_{j,k}} + \mu_j ,$$

$$\lambda_{j,k} = \exp\left(\frac{-d_{j,k}}{f}\right), \tag{2-19}$$

respectively in their original forms.

The author thus introduces FLC mechanism for the tuning of τ and f based on the following considerations.

τ and f^{-1} should be depressed in a certain way when ample adaptation data is at hand, and be enhanced otherwise, which is expected to adapt the HMM model without deteriorating the recognition performance even when the acquired data from the speaker is scarce.

The experiment records show that the proposed FLC design, FCMAP and FLC-VFS, meet the requirement quite well in two aspects:

1. As far as the speech recognition rate is concerned, FCMAP, FLC-VFS and FCMAP-FLCVFS respectively surpass the conventional MAP, VFS and MAP-VFS adaptation, regardless of the number of adaptation utterances, acquired from a new speaker.
2. The behaviors of τ and f^{-1} against adaptation data act as planned in the FLC mechanism design.

During the MLLR process in its original form, the adaptation in each iteration is governed by

$$\hat{\mu}_s = W_s \cdot \xi_s, \qquad (2\text{-}22)$$

The author thus proposed, again based on the same notion that adaptation should not deviate too much from the original means with little adaptation data at hands, a MAP-like MLLR adaptation, the FLC-MLLR adaptation as follows,

$$\tilde{\mu}_s = \alpha \cdot \mu_s + (1-\alpha) \cdot W_s \cdot \xi_s, \quad 0 \le \alpha \le 1, \qquad (6\text{-}1)$$

for which a T-S FLC mechanism is conceived for regulating α value in such a way that α would decrease when the W_s estimate is trust-worthy (as a result of sufficient training data) and increase otherwise to cope with the potential incorrectness lurking in W_s.

The experiment results reflect the success of FLC-MLLR design in two ways:

1. FLC-MLLR outruns MLLR and even MAPLR in the recognition performance, regardless of the amount of adaptation data at hands.
2. The behaviors of α with respect to the variation in adaptation data available do follow the requirement in the FLC design.

8.2 DW and GMM Adaptation with FLC

An *STLD*-driven FLC mechanism is devised for regulating the size of decision window (DW) in the application of audio event detection, and the performance of female screaming detection is examined in three operation backgrounds (office space, in-door parking lot and living room) with three individual acoustic features where the adaptation of GMM-based background models and screaming models are done by associated T-S FLC mechanism. The records show that the proposed scheme of variable-sized DW surpassed the one with fixed-length DW. Moreover, it is noted that the performance of the fixed-length DW reaches the score competitive against FLC-regulated DW at a DW setting that falls within the range of DW variation of the latter during the entire operation, which manifests the effectiveness of the proposed FLC-regulating design.

8.3 Future Developments

As a concluding remark, the author would like to point out some possible extensions to the use of FLC mechanism in the future study.

In the realm of speaker adaptation, there are many other techniques available for parameter tuning and not covered in the scope of the dissertation. Eigenvoice-adaptation, a younger cousin of eigenface methodology for face detection/recognition in image process, has been a new focus in recent years for instance. How FLC mechanism could be incorporated in the framework of eigenvoice process is an open issue and an FLC-eigen-MLLR or FLC-eigen-MAPLR would seem to be a promising subject for the next research.

Speaker adaptation by neural networks (NN), support vector machines (SVM) and genetic algorithm (GA) may also be considered for the bringing in of FLC mechanisms whenever plausible.

As for the matter of audio event detection, any audio events requiring the attention of a real surveillance application should be good research targets. The author starts with female screaming and there are many more to go on with: gunshot, explosion, noises arising from fiery quarrel or fighting or even the arson, just to list a few. Of course, surveillance applications do not always have to be that “serious” when it has nothing to do with public/private safety. Just imagine how to detect the singing of a specific kind of birds in the forest, the howling or roaring of a certain species of creatures in the grass land, and a duck quacking among a flock of honking geese, etc, which would make a lot more fun out of the research.

And as one final remark to end the writing of the dissertation, the use of T-S FLC mechanism is one choice from many fuzzy formulations in control by computation; Mamdani (linguistic) type fuzzy model [123], for instances, is an alternative that can be used in place of T-S FLC in the dissertation, of which the behaviors and performance can be examined if one wishes.

References

[1] L. G. Shapiro and G. C. Stockman, *Computer Vision*, Prentice-Hall, New Jersey, USA, 2001.

[2] D. A. Forsyth and J. Ponce, *Computer Vision: a Modern Approach*, Prentice-Hall, New Jersey, USA, 2003.

[3] B. H. Juang and L. R. Rabiner, “Automatic speech recognition – a brief history of the technology development,” *Encyclopedia of Language and Linguistics*, 2nd ed., Elsevier, 2005.

[4] L. R., Rabiner, “The power of speech,” *Science*, vol. 301, pp. 1494-1495, 2003.

[5] R. P. Lippmann, “Speech recognition by machines and humans,” *Speech Communication*, vol. 22, pp. 1-15, 1997.

[6] B. Atal, “Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification,” *Journal of Acoustical Society of America*, vol. 55, pp. 1304-1312, 1974.

[7] M. G. Rahim and B. –H. Juang, “Signal bias removal by maximum likelihood estimation for robust telephone speech recognition,” *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, pp. 19-30, 1996.

[8] Y. T. Juang, K. C. Huang and I. J. Ding, “Speaker adaptation based on MAP estimation using fuzzy controller,” *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2807-2813, 2003.

[9] I. J. Ding, “Incremental MLLR speaker adaptation by fuzzy logic control,” *Pattern Recognition*, vol. 40, no. 11, pp. 3110-3119, 2007.

[10] R. Kuhn, J.-C. Junqua, P. Nguyen and N. Niedzielski, “Rapid speaker adaptation in eigenvoice space,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 695–707, 2000.

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

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

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

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

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

[16] B. Mak, J. T. Kwok, S. Ho, "Kernel eigenvoice speaker adaptation," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 984–992, 2005.

[17] B. Zhou and J. Hansen, "Rapid discriminative acoustic model based on eigenspace mapping for fast speaker adaptation," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 4, pp. 554–564, 2005.

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

[19] B. Mak, R. Hsiao, S. Ho and J. T. Kwok, "Embedded kernel eigenvoice speaker

adaptation and its implication to reference speaker weighting," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1267-1280, 2006.

[20] B. Mak and R. Hsiao, "Kernel eigenspace-based MLLR adaptation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 784-795, 2007.

[21] R. J. Evans, E. L. Brassington and C. Stennett, "Video motion processing for event detection and other applications," in *Proceedings of International Conference on Visual Information Engineering*, pp. 93-96, 2003.

[22] A. Albiol, C. Sandoval, V. Naranjo and J. M. Mossi, "Robust motion detector for video surveillance applications," in *Proceedings of International Conference on Image Processing*, vol. 3, pp. II-379-382, 2003.

[23] T. Amano, S. Hiura, A. Yamaguti and S. Inokuchi, "Eigen space approach for a pose detection with range images," in *Proceedings of International Conference on Pattern Recognition*, pp. 622-626, 1996.

[24] E. M. DuPont, H. Yu and R. G. Roberts, "Object pose detection in the presence of background clutter and occlusion," in *Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory*, pp. 446-450, 2004.

[25] R. Cai, L. Lu, H.-J. Zhang and L.-H. Cai, "Highlight sound effects detection in audio stream," in *Proceedings of the IEEE International Conference on Multimedia Expo.*, vol. 3, pp. 37-40, 2003.

[26] O. Gillet and G. Richard, "Automatic transcription of drum sequences using audiovisual features," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. III-205-208, 2005.

[27] S. Pfeiffer, S. Fischer and W. Effelsberg, "Automatic audio content analysis," in *Proceedings of the 4th ACM international conference on Multimedia*, pp. 21-30, 1997.

[28] C. Clavel, T. Ehrette and G. Richard, “Event detection for an audio-based surveillance system,” in *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 1306-1309, 2005.

[29] A. Dufaux, L. Besacier, M. Ansorge and F. Pellandini, “Automatic sound detection and recognition for noisy environment,” in *Proceedings of European Signal Processing Conference (EUSIPCO)*, pp. 1033-1036, 2000.

[30] A. Harma, M. F. McKinney and J. Skowronek, “Automatic surveillance of the acoustic activity in our living environment,” in *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 634-637, 2005.

[31] L. Besacier, A. Dufaux, M. Ansorge and F. Pellandini, “Automatic sound recognition relying on statistical methods, with application to telesurveillance,” in *Proceedings of International Workshop on Intelligent Communication Technologies and Applications, with Emphasis on Mobile Communications*, pp. 116-120, 1999.

[32] P. K. Atrey, N. C. Maddage and M. S. Kankanhalli, “Audio based event detection for multimedia surveillance,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 813-816, 2006.

[33] A. Taylor, G. Grigg, G. Watson and H. McCallum, “Monitoring frog communities: an application of machine learning,” in *Proceedings of the Eighth Innovative Applications of Artificial Intelligence Conference*, pp. 1564-1569, 1996.

[34] T. Hsiao, B. Lee, T. Chou, H. Lien and Y. Chang, “Debris flow monitoring system and observed event in Taiwan: a case study at Aiyuzi River,” *Wuhan University Journal of Natural Sciences*, vol. 12, no. 4, pp. 610-618, 2007.

[35] M. Vacher, D. Istrate, L. Besacier, J. F. Serignat and E. Castelli, “Life sounds extraction and classification in noisy environment,” in *Proceedings of the*

International Association of Science and Technology for Development - Signal and Image Processing (IASTED-SIP), pp. 77-82, 2003.

[36] M. Cristani, M. Bicego and V. Murino, “On-line adaptive background modeling for audio surveillance,” in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 2, pp. 399-402, 2004.

[37] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 246-252, 1999.

[38] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice- Hall, New Jersey, USA, 1993.

[39] J. R. Deller, J. H. L. Hansen and J. G. Proakis, *Discrete-Time Processing of Speech Signals*, IEEE Press, 2000.

[40] Y. R. Wang, J. M. Shieh and S. H. Chen, “Tone recognition of continuous Mandarin speech based on neural network,” *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 2, pp. 146-150, 1995.

[41] F. Runstein and F. Violaro, “An isolated-word speech recognition system using neural networks,” in *Proceedings of the 38th Midwest Symposium on Circuits and Systems*, vol. 1, pp 550-553, 1995.

[42] W. Y. Chen, S. H. Chen and C. J. Lin, “A speech recognition method based on the sequential multi-layer perceptrons,” *Neural Networks*, vol. 9, no. 4, pp.655-669, 1996.

[43] A. Ganapathiraju, J. E. Hamaker and J. Picone, “Applications of support vector machines to speech recognition,” *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2348- 2355, 2004.

[44] R. Solera-Ureña, D. Martín-Iglesias, A. Gallardo-Antolín, C. Peláez-Moreno and F. Díaz-de-María, “Robust ASR using support vector machines,” *Speech*

Communication, vol. 49, no. 4, pp. 253-267, 2007.

[45] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state Markov chains,” *The Annals of Mathematical Statistics*, vol. 37, pp. 1554-1563, 1966.

[46] L. E. Baum and J. A. Egon, “An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology,” *Bulletin of the American Meteorological Society*, vol. 73, pp. 360-363, 1967.

[47] L. E. Baum, “An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov process,” *Inequalities*, vol. 3, pp. 1-8, 1972.

[48] S. E. Levinson, L. R. Rabiner and M. M. Sondhi, “An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition,” *The Bell System Technical Journal*, vol. 62, no. 4, 1983.

[49] L. R. Rabiner and B. H. Juang, “An introduction to hidden Markov models,” *IEEE ASSP Magazine*, pp. 4-16, 1986.

[50] C. H. Lin, C. H. Wu, P. Y. Ting and H. M. Wang, “Frameworks for recognition of mandarin syllables with tones using sub-syllabic units,” *Speech Communication*, vol. 18, no. 2, pp. 175-190, 1996.

[51] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” in *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, 1989.

[52] F. Jelinek, “Continuous speech recognition by statistical methods,” in *Proceedings of the IEEE*, vol. 64, no. 4, pp. 532-556, 1976.

[53] G. D. Forney, “The viterbi algorithm,” in *Proceedings of the IEEE*, vol. 61, no. 3,

pp. 268-278, 1973.

[54] P. C. Woodland, “Speaker adaptation: techniques and challenges,” in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 85-90, 1999.

[55] 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 Transactions on Acoustics, Speech and Signal Processing*, vol. 39, pp. 806–814, 1991.

[56] B. H. Juang and L. R. Rabiner, “The segmental k-means algorithm for estimating parameters of hidden Markov models,” *IEEE Transactions on Signal Processing*, vol. 38, no. 9, pp. 1639-1641, 1990.

[57] L. R. Rabiner, J. G. Wilpon and B. H. Juang, “A segmental k-means training for connected word recognition,” *AT&T Tech. J.*, vol. 65, pp. 21-32, 1986.

[58] J. L. Gauvain and C. H. Lee, “Maximum a *posteriori* estimation for multivariate Gaussian mixture observations of Markov chains,” *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 291-298, 1994.

[59] M. H. DeGroot, *Optimal Statistical Decisions*, McGraw-Hill, New York, 1970.

[60] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis (second ed.)*, Springer-Verlag, New York, 1985.

[61] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[62] K. Ohkura, M. Sugiyama and S. Sagayama, “Speaker adaptation based on transfer vector field smoothing with continuous mixture density HMMs,” in *Proceedings of International Conference on Spoken Language Processing*, pp. 369-372, 1992.

[63] H. Hattori and S. Sagayama, “Vector field smoothing principle for speaker adaptation,” in *Proceedings of International Conference on Spoken Language*

Processing, pp. 381-384, 1992.

[64] J. Ishii, M. Tonomura and S. Matsunaga, “Speaker adaptation using tree structured shared-state HMMs,” in *Proceedings of International Conference on Spoken Language Processing*, pp. 1149-1152, 1996.

[65] J.-I. Takahashi and S. Sagayama, “Vector-field-smoothed Bayesian learning for fast and incremental speaker/telephone-channel adaptation,” *Computer Speech and Language*, vol. 11, pp. 127-146, 1997.

[66] S. J. Cox and J. S. Bridle, “Unsupervised speaker adaptation by probabilistic spectrum fitting,” in *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, pp. 294-297, 1989.

[67] C. J. Leggetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models,” *Computer Speech and Language*, vol. 9, pp. 171-185, 1995.

[68] J. T. Chien, L. M. Lee and H. C. Wang, “Estimation of channel bias for telephone speech recognition,” in *Proceedings of International Conference on Spoken Language Processing*, vol. 3, pp. 1840-1843, 1996.

[69] J. T. Chien and H. C. Wang, “Telephone speech recognition based on Bayesian adaptation of hidden Markov models,” *Speech Communication*, vol. 22, pp. 369-384, 1997.

[70] C. Chesta, O. Siohan and C. H. Lee, “Maximum a *posteriori* linear regression for hidden Markov model adaptation,” in *Proceedings of the European Conference on Speech Communication and Technology*, pp. 211-214, 1999.

[71] W. Chou, “Maximum a *posteriori* linear regression with elliptically symmetric matrix priors,” in *Proceedings of the European Conference on Speech Communication and Technology*, pp. 1-4, 1999.

[72] A. P. Dempster, N. M. Laird and D. B. Rubin, “Maximum likelihood from

incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1-38, 1977.

[73] A. Gunawardana and W. Byrne, "Robust estimation for rapid speaker adaptation using discounted likelihood techniques," in *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, pp. 985-988, 2000.

[74] O. Siohan, C. Chesta and 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*, pp. 147-150, 1999.

[75] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72-83, 1995.

[76] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," *IEEE Acoustic, Speech and Signal Processing Magazine*, vol. 3, no. 1, pp. 4-16, 1986.

[77] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Francisco, 1988.

[78] M. Markou and S. Singh, "Novelty detection: a review-part 1: statistical approaches," *Signal Processing*, vol. 83, pp. 2481-2497, 2003.

[79] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, pp. 84-95, 1980.

[80] C. Fraley and A. E. Raftery, "How many clusters? Which clustering method? Answers via model based cluster analysis," *The Computer Journal*, vol. 41, pp. 578-588, 1998.

[81] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.

[82] L. A. Zadeh, "Fuzzy algorithms," *Information and Control*, vol. 12, pp. 94-102,

1968.

[83] L. A. Zadeh, “Outline of a new approach to the analysis of complex system and decision processes,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no.1, pp.28-44, 1973.

[84] L. A. Zadeh, “Making computers think like people,” *IEEE Spectrum*, vol. 21, no. 8, pp. 26-32, 1984.

[85] L. A. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning,” *Information Science*, vol. 8, pp. 43-80, 1975.

[86] L. A. Zadeh, “Fuzzy sets as a basis for a theory of possibility,” *Fuzzy Sets and Systems*, vol. 1, pp. 3-28, 1977.

[87] H.-J. Zimmermann, *Fuzzy Set Theory and Its Applications (3rd ed.)*, Kluwer Academic, 1996.

[88] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

[89] X.-H. Xu, J. Zhu and Q. Guo, “Fuzzy c-means clustering based phonetic tied-mixture HMM in speech recognition,” *Journal of Shanghai Jiaotong University*, vol. 10, no. 1, pp. 16-20, 2005.

[90] J. J. Li, X. D. Xia and S. S. Gu, “Mandarin four-tone recognition with the fuzzy c-means algorithm,” in *Proceedings of FUZZ-IEEE*, vol. 2, pp. 1059-1062, 1999.

[91] D. Tran and M. Wagner, “Generalised fuzzy hidden Markov models for speech recognition,” in *Proceedings of International Conference on Fuzzy Systems*, pp. 345-351, 2002.

[92] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer, 1992.

[93] P. C. Cosman, K. L. Oehler, E. A. Riskin and R. B. Gray, “Using vector quantization for image processing,” in *Proceedings of the IEEE*, vol. 81, pp.

1326-1341, 1993.

[94] W. J. Hwang, B. Y. Ye and L. Y. Lai, “Nonparametric classifier design using greedy tree-structured vector quantization technique,” *Pattern Recognition Letters*, vol. 18, pp. 409-414, 1997.

[95] H. Matsumoto and Y. Yamashita, “Unsupervised speaker adaptation of spectra based on a minimum fuzzy vector quantization error criterion,” *Second Joint Meeting of ASA and ASJ*, 1988.

[96] K. Shikano, S. Nakamura and M. Abe, “Speaker adaptation and voice conversion by codebook mapping,” *IEEE International Sympoisum on Circuits and Systems*, vol. 1, pp. 594-597, 1991.

[97] L. Cong, C. S. Xydeas and A. F. Erwood, “Combining fuzzy vector quantization and neural network classification for robust isolated word speech recognition,” in *Proceedings of IEEE ICCS-94*, pp. 884-887, 1994.

[98] C. S. Xydeas and L. Cong, “Combining neural network classification with fuzzy vector quantization and hidden Markov models for robust isolated word speech recognition,” in *Proceedings of the IEEE International Symposium on Information Theory*, 1995.

[99] P. Le Cerf, Weiye Ma and D. Van Compernolle, “Multilayer perceptrons as labelers for hidden Markov models,” *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 185-193, 1994.

[100] T. Zhao and P. Y. Woo, “Fuzzy speech recognition,” in *Proceedings of International Joint Conference on Neural Networks*, vol. 5, pp. 2959-2961, 1999.

[101] Y. Q. Ying and P. Y. Woo, “Speech recognition using fuzzy logic,” in *Proceedings of International Joint Conference on Neural Networks*, vol. 5, pp. 2962- 2964, 1999.

[102] R. Halavati, S. B. Shouraki, M. Eshraghi, M. Alemzadeh, P. Ziaie, “A novel

fuzzy approach to speech recognition," in *Proceedings of International Conference on Hybrid Intelligent Systems*, pp. 340-345, 2004.

[103] P. Melin, J. Urias, D. Solano, M. Soto, M. Lopez and O. Castillo, "Voice recognition with neural networks, fuzzy logic and genetic algorithms," *Engineering Letters*, vol. 13, no. 2, pp. 108-116, 2006.

[104] P. A. Nava and J. M. Taylor, "Speaker independent voice recognition with a fuzzy neural network," in *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, vol. 3, pp. 2049-2052, 1996.

[105] N. Kasabov, R. Kozma and M. Watts, "Phoneme-based speech recognition via fuzzy neural networks modeling and learning," *Information Sciences*, vol. 110, no. 1, pp. 61-79, 1998.

[106] D. Doye and T. Sontakke "Speech recognition using modular general fuzzy min-max neural network," *IETE Journal of Research*, vol. 48, no. 2, pp. 99-103, 2002.

[107] O. Grigore and I. Gavat, "Neuro-fuzzy models for speech pattern recognition in Romanian language" in *Proceedings of European Symposium on Intelligent Techniques (ESIT)*, pp. 98-103, 1999.

[108] N. Kasabov, "Evolving fuzzy neural networks - algorithms, applications and biological motivation," in *Proceedings of the Iizuka'98*, Iizuka, Japan, pp. 271-274, 1998.

[109] E. Khan, "Recurrent fuzzy logic in speech recognition," in *Proceedings of the WESCON Conference*, pp. 602-608, 1995.

[110] C. T. Lin, H. W. Nein and W. F. Lin, "Speaker adaptation of fuzzy-perceptron-based speech recognition," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 7, no. 1, 1999, 1-30.

[111] M. J. F. Gales, "The generation and use of regression class trees for MLLR

adaptation," *Tech. Rep. CUED/F-INFENG/TR263*, Cambridge University, 1996.

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

[113] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[114] M. Sugeno and K. Murakami, "Fuzzy parking control of model car," in *Proceedings of the IEEE 23rd Conference on Decision and Control*, pp. 902-903, 1984.

[115] S. Kermiche, M. L. Saidi, H. A. Abbassi and H. Ghodbane, "Takagi-Sugeno based controller for mobile robot navigation," *Journal of Applied Science*, vol. 6, no. 8, pp. 1838-1844, 2006.

[116] Marcelo C. M. Teixeira, Grace S. Deaecto, Ruberlei Gaino, Edvaldo Assunção, Aparecido A. Carvalho and Uender C. Farias, "Design of a fuzzy Takagi-Sugeno controller to vary the joint knee angle of paraplegic patients," in *Proceedings of International Conference on Neural Information Processing* , pp. 118-126, 2006.

[117] Y. Li, S. S. Narayanan and C.-C. J. Kuo, "Audiovisual-based adaptive speaker identification," in *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, vol. 5, pp. 812-815, 2003.

[118] J. L. Gauvain and C. H. Lee, "Speaker adaptation based on MAP estimation of hmm parameters," in *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, pp. II-558-661, 1993.

[119] H. C. Wang, "MAT – a project to collect Mandarin speech data through telephone networks in Taiwan," *Comput. Linguist. Chinese Lang. Process.*, vol. 2, pp. 73-89, 1997.

[120] H. C. Wang, F. Seide, C. Y. Tseng and L. S. Lee, "MAT-2000 - design,

collection, and validation of a mandarin 2000-speaker telephone speech database," in *Proceedings of International Conference on Spoken Language Processing*, vol. 4, pp. 460-463, 2000.

[121] O. Siohan, T. A. Myrvoll and C.-H. Lee, "Structural maximum a *posteriori* linear regression for fast HMM adaptation," in *Proceedings of the ISCA Workshop on Automatic Speech Recognition*, pp. 120-127, 2000.

[122] W. G. Cochran, "Problems arising in the analysis of a series of similar experiments," *Journal of the Royal Statistical Society*, vol. 4, pp. 102-118, 1937.

[123] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic systems," *Fuzzy Sets and Systems*, vol. 26, pp. 1182–1191, 1977.