論文 / 著書情報
Article / Book Information

題目(和文)	音声認識のためのデータ不足問題に対する頑健性の研究
Title(English)	A study on robustness against data insufficiency for speech recognition
著者(和文)	篠田浩一
Author(English)	Koichi Shinoda
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:乙第3529号, 授与年月日:2001年3月31日, 学位の種別:論文博士, 審査員:
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:乙第3529号, Conferred date:2001/3/31, Degree Type:Thesis doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

A Study on Robustness against Data Insufficiency for Speech Recognition

Koichi Shinoda

A Dissertation Submitted to
Department of Computer Science,
Graduate School of Information Science and Engineering
of Tokyo Institute of Technology
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Engineering

March 2001

Preface

In statistical pattern recognition, data insufficiency often causes serious degradation of recognition accuracy. This problem can be avoided by selecting models whose sizes are optimal for the given amount of data. In this thesis, a structural approach is proposed in which numerous models are prepared in the form of a tree structure and the model with the optimal size is selected by using information criteria. This approach was applied to two problems of speech recognition: acoustic modeling and speaker adaptation.

Context-dependent phone units, such as triphones, have recently come to be used for acoustic modeling in speech recognition systems. While most such systems cluster the model parameters (e.g., by subword clustering and state clustering) in order to limit the model size and thus avoid poor recognition accuracy due to a lack of training data, none of them provide effective criteria for determining the optimal number of clusters. This thesis therefore describes a method in which the minimum description length (MDL) criterion is used to optimize the number of clusters.

Speaker adaptation based on maximum *a posteriori* (MAP) estimation has been studied extensively, and this thesis presents a structural maximum *a posteriori* (SMAP) approach to improve the MAP estimates obtained when the amount of adaptation data is small. The model parameter space is assumed to be structured hierarchically and each model parameter is estimated as the weighted sum of the parameters in more than one tree layer.

These two methods increase the robustness of speech recognition against data

insufficiency, and their effectiveness was confirmed in a series of recognition experiments.

Acknowledgments

First and foremost, I would like to express my sincere appreciation to my supervisor, Prof. Sadaoki Furui (TITECH, Japan) for his guidance, support and encouragement throughout the years of my Ph.D studentship. He showed me the best way to put together all the studies I have done in speech recognition. His comments and suggestions have been crucial to the completion of my thesis. I would also like to thank other members of my thesis committee, Prof. Hidemitsu Ogawa, Prof. Hozumi Tanaka, Prof. Takao Kobayashi, and Prof. Takenobu Tokunaga (TITECH, Japan), who have been good enough to give this work a very serious review.

The research reported in Chapter 3 was conducted at Computer and Communication Media Research, NEC Corporation. I would therefore like to express my sincere gratitude to Dr. Satoshi Goto, Dr. Takao Nishitani, Mr. Masao Watari, Dr. Kazunori Muraki, Dr. Takao Watanabe, Mr. Kazunaga Yoshida, Mr. Kaichiro Hatazaki, and Mr. Ken'ichi Iso, for their continuous encouragement and support. I especially thank Dr. Watanabe and Mr. Iso for their advice. I also thank Mr. Ryosuke Isotani, Mr. Keizaburo Takagi, and Mr. Tadashi Emori who were kind enough to discuss problems and issues with me anytime I wished. Special thanks are also due to Dr. Jun'ichi Takeuchi and Dr. Li Hang. The idea of applying the MDL criterion to acoustic models was yielded from discussions with them.

The research reported in Chapter 4 was conducted at Dialogue System Research, Bell Laboratories, Lucent Technologies. Sincere appreciation goes to Dr.

Chin-Hui Lee for giving me the opportunity to work at Bell Laboratories and for providing much helpful advice with regard to research procedures. He showed me theoretical ways to solve problems, which have been of great help to me in my research activities since then. I am also grateful to Dr. Biing-Hwang Juang, Dr. Frank K. Soong, and Dr. Arun C. Surendran for giving me insightful comments to my study.

I owe a great many thanks to many people who were kind enough to help me over the course of this work. I would like to express here my sincere appreciation to all of them. Thanks also go to all the people who read this thesis and made many helpful comments, though the responsibility for any flaws and errors it may contain remains entirely mine.

Finally, I also would like to express a deep debt of gratitude to my parents, who instilled in me a love for learning and thinking, and to my wife, Miyuki, for her constant encouragement and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Data insufficiency problem

In statistical pattern recognition, a *model* with a number of parameters is prepared and those model parameters are estimated by using data. This process is called *training* and the data used for training is called training data. One might expect that the larger the number of the parameters is, the more accurate the recognition, since a model with more parameters can represent more features necessary for recognition. It is often observed, however, that when the number of model parameters exceeds a certain value, the recognition accuracy becomes worse. This is because there are too many parameters to be estimated precisely from the limited amount of training data. This is called the *data insufficiency problem* and is inevitable in statistical pattern recognition.

Let me explain this problem with a simple example. Consider the problem of English-letter speech recognition, in which the utterance of an English letter is assigned to one of the 26 letters in the English alphabet. The pattern recognition process is roughly divided into two parts: feature extraction and pattern matching. Suppose that the feature-extraction part processes each utterance into one feature vector with a fixed number of dimensions and that the standard pattern for each category is represented by a Gaussian-mixture distribution in which each

Gaussian component is multivariate. Suppose too that the number of mixture components is the same in all the categories and that the recognition accuracy is measured using test data different from the training data. When the number of mixtures is too small, the recognition accuracy may be very low because the number of mixtures is too small to represent the features of the training data. The recognition accuracy improves as the number of mixtures increases (i.e., as the model becomes larger), but after the number of mixtures exceeds a certain level a further increase results in less accurate recognition. This is because the model is *over-trained*: a significant portion of its parameters represent training data features unnecessary for recognition of the test data and this noise degrades the recognition accuracy.

This data insufficiency problem appears in many aspects of statistical pattern recognition. To avoid this, we need to obtain a model large enough to represent the features necessary for recognition but smaller than an over-trained model. Of models with the same performance, the simplest model is the most preferable, mainly because it requires less computational resources. The belief that the best model is the simplest has been stated in various ways:

- Pluralitas non est ponenda sine necessitas (plurality should not be posited without necessity). *William of Ockham*
- If I had more time I could write a shorter letter. *Blaise Pascal*
- Make everything as simple as possible but not simpler. *Albert Einstein*

It is difficult, however, to acquire such a parsimonious model.

There have been three different approaches to solving this data insufficiency problem, and they all utilize the *model selection* scheme: a number of models of various sizes are prepared beforehand and the *optimal* model is selected. These approaches are the following:

- Tuning to test data

- Cross validation
- Information criterion

In the first one all the prepared models are trained, each is used to recognize data and the one with the highest accuracy is selected. This approach is implicitly utilized in most pattern recognition systems since it is simple and needs no extra information. One significant problem with this approach, however, is that the amount of test data is usually limited, and tuning to this limited test data might result in poor recognition in actual use. The second approach, cross validation, was developed in response to this problem. In this approach the total data is divided into several portions and one portion is selected as test data. Then the model is trained using the rest of the data, and a recognition experiment is carried out using the test data. This process is repeated using each of the portions as the test data (i.e., the number of repetitions is the same as the number of portions). Since the amount of test data seemingly increases, it is expected that the selected model is more robust. The problem of this approach is that its computational cost is high and there are no established ways to divide the data. The third approach utilizes an information criterion of model complexity. A number of such criteria have been proposed and the most widely known are the Akaike information criterion (AIC) [1], the Bayesian information criterion (BIC) [46], and the minimum description length (MDL) criterion [43]. There are a number of applications of these criteria and some have been proved to be effective. The problem with this approach is that it is difficult to apply those criteria directly. In most case it is necessary to use a number of approximations which are not fully justified.

All these approaches based on the model selection scheme share two serious problems. The first is that the selected model may be very different from a truly optimal model when the number of models prepared is not large enough. And the second is that they cannot be used in cases for which the amount of training data often changed or in which the computational cost for estimating

the parameters of many models is prohibitively high (examples of these cases will be shown in the next section). The goal is to provide the optimal model for the given amount of training data and to do this with a small computational cost. It may be necessary to prepare a model set comprising numerous models with different sizes, to estimate the model parameters efficiently, and to select the optimal model without evaluating their performance on test data.

1.2 Structural approach to the data insufficiency problem

The following *structural approach* to attack the data insufficiency problem is proposed here. First prepare a *tree-structured* model set in which each layer in the tree represents a model: to each node in the tree is attached a probabilistic distribution function (pdf) for data, a *node pdf*. The root-layer model which consists only of the root node represents the smallest model, the leaf-layer model which consists of all the node of the leaf layers represents the largest model, and a layer closer to the root node represents a simpler model. A node in one layer and nodes in its subsidiary layer are connected by branches. Each training sample is associated with a node in the leaf layer. The pdf of each node is estimated by using the training samples associated with all the leaf nodes the node governs. After preparing this tree-structured set of models, select one layer by using a criterion which describes the relationship between the amount of data and the size of the model.

Let me explain this approach by using an example. One Gaussian distribution is attached to each node in the tree. The root layer, which consists only of the root node, represents the simplest model, a single Gaussian distribution. The leaf layer, comprising all the leaf nodes, represents the most complex model. The distribution in each node is estimated by using the data associated with the leaf nodes it governs. In most case, when a certain amount of training data is given,

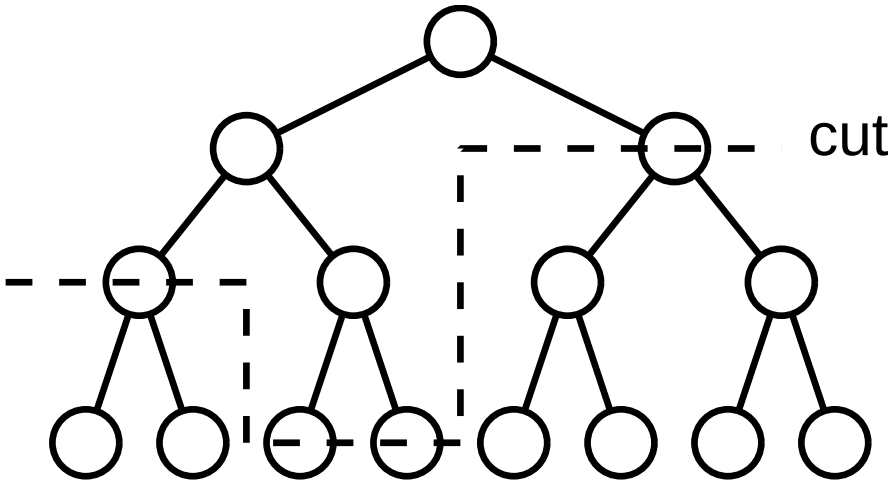

Figure 1.1: Tree structure.

the root-layer model is too small and the leaf-layer model is too large; the model represented by one of the intermediate layers is the optimal model. It is easily understood that models are not limited to the node sets in which all the nodes belong to the same layer. Any node sets which divide the tree into an upper part and a lower part can be used as a model. Such a node set is called a *cut* (see Figure 1.1).

As will be shown later, the problem of model selection can be resolved into many subproblems of node selection in sub-structures embedded in the tree. The computational cost of solving these subproblems will be shown to be much smaller than the computational cost of the greedy approach in which all the possible models have to be trained and evaluated. This subproblem approach is especially useful in cases where the training cost is high or the amount of training data changes frequently.

There are three issues important in this approach. First, the root node and the leaf layer should be designed appropriately. It can be generally assumed that the lower and upper bounds on the amount of data available are given. The parameters of the root-layer model can be precisely estimated from the least amount of data, while those of the leaf-layer model should be estimated from an amount of data large enough to represent all the features necessary for recognition. The

second issue is that the way to construct the tree structure should be specified. The tree structure can be made by clustering the leaf nodes hierarchically. The pdfs for the nodes nearer the root node should represent more global distributions in the space of data, while those of the nodes nearer to the leaf layer should represent more local distributions. Such trees cannot be made unless the measure of distance between the node pdfs is defined appropriately. Finally, the model selection framework needs to be provided: a framework not only providing a method to resolve the model selection problem into the node selection problem but also providing a criterion for selecting one node set.

This thesis is organized as follows. Chapter 2 briefly reviews speech recognition and describes the data insufficiency problem in speech recognition. For acoustic modeling it describes a structural approach using the MDL criterion, and for speaker adaptation it describes a structural Bayes approach. Chapters 3 and 4 describe these approaches in detail. Finally, Chapter 5 concludes this thesis.

Chapter 2

Speech Recognition

2.1 Overview

Nowadays there are a lot of commercial products using *speech recognition*, automatic recognition of human speech by machines. Examples are dictation software, call center systems, car navigation systems, educational software for learning foreign languages, video games, and electrical toys. Although the application areas of speech recognition are still limited, it is already regarded as one of the major tools for the man-machine interface.

Speech recognition has had a long history of research and development, and almost three decades passed before it was used in commercial products for general use. The recent advances in speech recognition have been sped by the following two factors.

- A probabilistic model, the hidden Markov model (HMM), was developed and used for speech recognition.
- A large speech database became available.

Of course, it should be noted that the rapid progress of computational hardware (CPUs, memories, etc.) also helped. But although speech recognition is already useful, there are still a lot of problems to be solved. Some of them are the

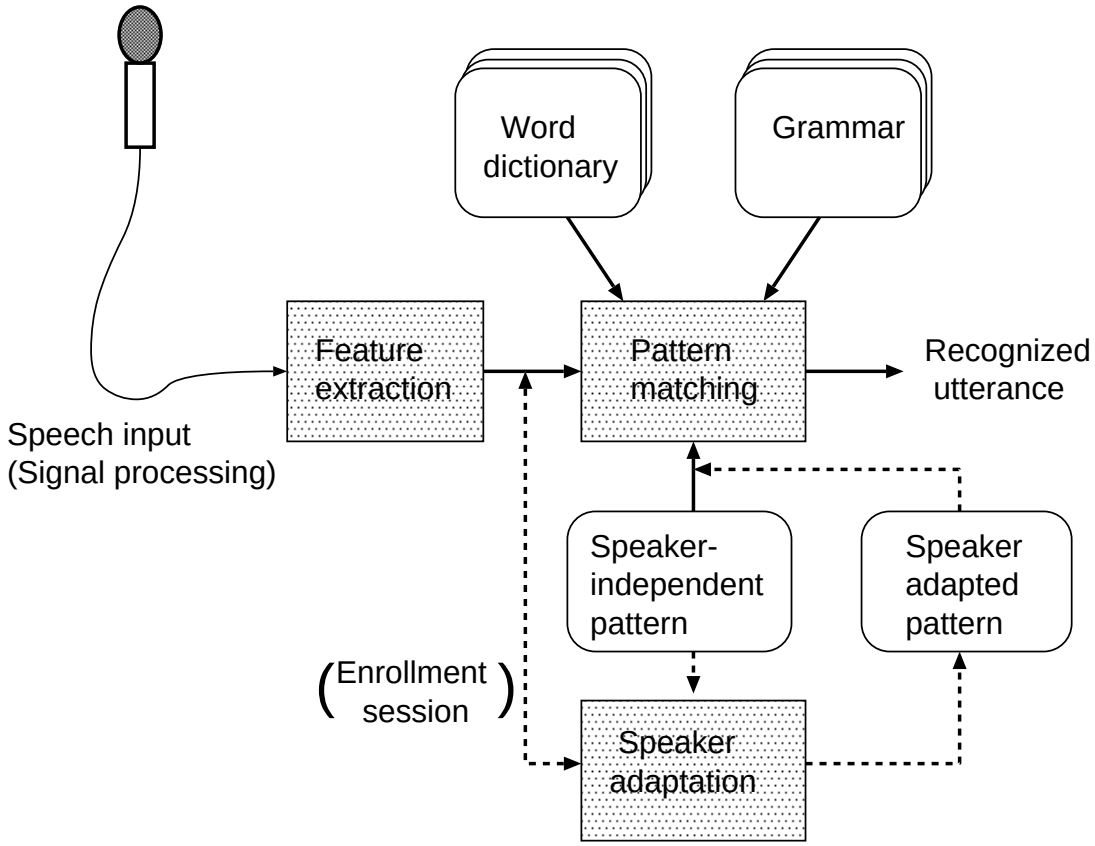

Figure 2.1: Speech recognition system.

following:

- Its performance is still much inferior to that of real people.
- Speaker-independent systems, which require no enrollment process for users, are less accurate than speaker-dependent systems, for which users have to enroll.
- Its performance is degraded under adverse conditions, noisy conditions such as those provided by telephone lines.
- Its performance is also degraded by changes in speaking style. Speech that is read is recognized better than spontaneous speech in lectures, meetings, and conversations.

Figure 2.1 shows a typical speech recognition system. The process of speech recognition is roughly divided into two parts: feature extraction and pattern matching. The input analog signals are first transformed into digital signals by analog-to-digital (AD) conversion. Then the features useful for speech recognition are extracted from the signals. A time series of extracted features is called an *input pattern*. Next, the pattern matching between the input pattern and each standard pattern is carried out. (One standard pattern for each word in the recognition vocabulary is prepared ahead of time.) The pattern matching yields the distance between the input pattern and each standard pattern, and the word corresponding to the standard pattern closest to the input pattern is selected as the recognized word.

2.2 Feature extraction

2.2.1 AD conversion

The input analog signals are first sampled for digital processing. While the frequency range perceptible to human beings is between 20 and 20 000 Hz, the range including the sound of human speech is only from 50 to 7000 Hz. A sampling frequency of 14 000 Hz is therefore high enough for speech processing. A frequency lower than 14 000 Hz, however, is sometimes used because of the limitations of the transmission channel or to reduce computational costs.

2.2.2 Short-term spectrum

Speech research has proved that *pitch* and *formant* are important features for human perception. In the simplest speech production model, the vocal code generates excited signals (which correspond the pitch) and the vocal tract (whose response corresponds the formant) filters the signals. Although it seems that speech recognition may be realized by simulating the human speech production system, for the following two reasons this is not possible with today's technology.

First, it is difficult to observe the real-time movement of the vocal organization. Second, it is not always possible to estimate the pitch and formant frequency from the acoustic signals observed.

Instead, in most cases the short-term spectrum is used for speech recognition in most case. Here it is assumed that speech signals are stationary during a short period, 10-100 ms. Then the spectrum for this period is computed using the fast Fourier transform (FFT).

2.2.3 Mel scale

The *mel* scale is defined in order to take into consideration the human perception of pitch. The frequency of 1000 Hz is defined as 1000 mel, and the frequency that people feel to be n times as high as 1000 Hz is defined as $(n \times 1000)$ mel. It should be noted that this mel scale is different from the octave scale used in music.

The mel scale can be well approximated by using the following equation:

$$f(\text{mel}) = \frac{1000}{\log_{10} 2} \log_{10}(\frac{f(\text{Hz})}{1000} + 1). \tag{2.1}$$

This mel scale transform has been often used in speech recognition, mainly because it emphasizes the lower frequencies more important for the classification of phones.

2.2.4 Cepstrum

The *cepstrum* of a signal is defined as a Fourier transform of the logarithm of the signal's power spectrum. It is especially useful when input signals are the superposition of excitation signals and linear filters. Speech signals are one such superposition: the excitation signals generated from the vocal code are filtered by the vocal tract response.

Let $y(n)$ be the speech signal at time n, $v(n)$ be the excited signal from the

vocal code, and $h(n)$ be the vocal tract response. Then

$$y(n) = v(n) * h(n), \tag{2.2}$$

$$Y(e^{jw}) = V(e^{jw}) * H(e^{jw}), \tag{2.3}$$

$$\log|Y(e^{jw})| = \log|V(e^{jw})| + \log|H(e^{jw})|, \tag{2.4}$$

where $Y(\cdot)$, $V(\cdot)$, $H(\cdot)$ are the Fourier transforms of $y(n)$, $v(n)$, and $h(n)$.

Then the cepstrum $c(k)$ is

$$c(k) = v(k) + h(k). \tag{2.5}$$

Although the dimension of cepstrum is the same as that of time, the term *quefrency* is used as the dimension of cepstrum. In the cepstrum, the component $v(k)$ of the vocal tract response is dominant in the lower quefrency, while the component $h(k)$ of the vocal code has a strong peak at high quefrencies, which corresponds to the pitch frequency and its harmonics

The influence of the pitch is efficiently removed by *liftering* (analogous to filtering in the spectral domain) the components $v(k)$ in the quefrency domain, and the components representing the vocal tract response are used as the features for speech recognition.

2.2.5 LPC cepstrum

The cepstral coefficients can also be derived by using the linear predictive coding (LPC) method in which speech is modeled as the output of an all-pole filter. The estimate of speech sample $s(n)$ at time n is approximated as a linear combination of the past p samples:

$$\hat{s}(n) = \sum_{i=1}^{p} a_i s(n-i), \tag{2.6}$$

where the coefficients $a_i, i = 1, \ldots, p$ are the prediction coefficients and are assumed to be constant over the analysis frame.

The prediction error is given by

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{i=1}^{p} a_i s(n-i). \tag{2.7}$$

What needs to be minimized is the mean squared error over a segment of speech, which is defined as follows:

$$E_n = \sum_m e^2(n+m) \tag{2.8}$$

$$= \sum_m [s(n+m) - \sum_{k=1}^{p} a_k s(n+m-k)]^2. \tag{2.9}$$

Differentiating E_n with respect to the coefficients a_k and setting it to zero,

$$\frac{\delta E_n}{\delta a_k} = 0, \qquad k = 1, \ldots, p, \tag{2.10}$$

we obtain

$$\sum_m s(n+m-k)s(n+m) = \sum_{i=1}^{p} a_i \sum_m s(n+m-i)s(n+m-k), \quad k = 1, \ldots, p. \tag{2.11}$$

If

$$\phi_n(i,k) = \sum_m s(n+m-i)s(n+m-k), \tag{2.12}$$

then

$$\phi_n(k,0) = \sum_{i=1}^{p} a_k \phi_n(i,k), \qquad k = 1, \ldots, p. \tag{2.13}$$

The LPC coefficients $a_k, k = 1, \ldots, p$ are obtained by solving this set of equations, and from these LPC coefficients can be derived the LPC cepstral coefficients:

$$c_0 = \log \sigma^2, \tag{2.14}$$

$$c_m = a_m + \sum_{k=1}^{m-1} \frac{k}{m} c_k a_{m-k}, \qquad 1 \le m \le p, \tag{2.15}$$

$$c_m = \sum_{k=1}^{m-1} \frac{k}{m} c_k a_{m-k}, \qquad m \ge p, \tag{2.16}$$

where σ^2 is the gain term in the LPC model.

2.2.6 Dynamic features

Dynamic features of the spectrum play an important role in human speech perception, and the delta cepstrum [16] developed to take these features into consideration is defined as follows:

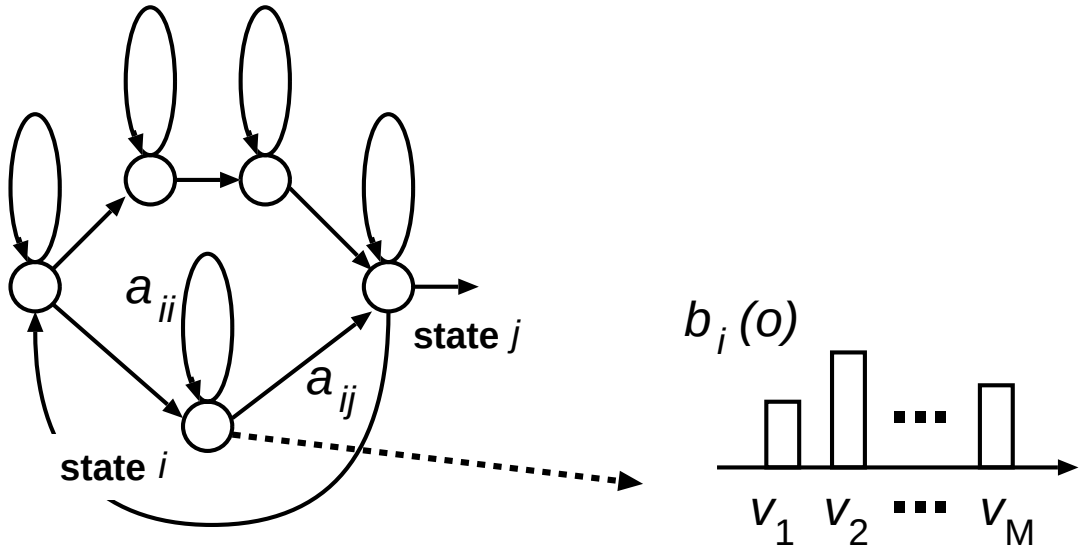

Figure 2.2: Discrete hidden Markov model (DHMM).

$$\Delta c_n(t) = \frac{\sum_{k=-K}^{K} k c_n(t+k)}{\sum_{k=-K}^{K} k^2}. \tag{2.17}$$

This delta cepstrum and the second derivative feature, delta-delta cepstrum, are often used in many speech recognition systems and are effective.

2.3 Hidden Markov models

2.3.1 Definition

A statistical approach using *hidden Markov models* (HMMs) has recently been widely used for speech recognition(e.g., [41]). In this approach the speech signals are characterized as outputs from Markov sources. In the recognition of words, for example, a HMM is assigned to each word in the vocabulary, and for each utterance the recognized word selected is the one whose HMM is most likely to produce that utterance.

HMMs are classified into three types according to the form of the output *probability density function* (pdf) in each state: the discrete HMMs, in which the output pdf is discrete; the continuous density HMMs, in which the output pdf

is continuous; and the semi-continuous HMMs which is the combination of the discrete HMMs and the continuous-density HMMs.

For the simplicity of explanation, the focus is on the discrete HMMs (DHMM) for a while (see Figure 2.2). Let $\mathbf{O} = (\mathbf{o}_1, \ldots, \mathbf{o}_t, \ldots, \mathbf{o}_T)$ be an observation sequence. Let S be the number of states, $A = \{a_{ij}\}$ be a set of transition probability distributions, in which a_{ij} is the probability of transition from state i to state j; let $B = \{b_i(\mathbf{o}_t)\}$ be a set of the output probability distributions, where $b_i(\mathbf{o}_t)$ be the output probability of the feature vector $\mathbf{o}_t$ at state i; let $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M\}$ be the set of the symbols, where M is the number of symbols; and let q_t be the state at time t; let $\mathbf{q} = \{q_1, \ldots, q_T\}$ be a state sequence in which q_t is the state at time t. Then

$$b_i(\mathbf{o}_t) = b_i(k) = P(\mathbf{o_t} = \mathbf{v_k}, | q_t = i), \qquad i = 1, \ldots, S, \tag{2.18}$$

where S is the number of state in the HMM. The initial probability distribution $\pi = \{\pi_i\}$ is also defined, where π_i is the probability of being state i at time 1:

$$\pi_i = P(q_1 = i), \qquad i = 1, \ldots, S. \tag{2.19}$$

The parameter set $\lambda = (A, B, \pi)$ is the complete parameter set for the model. From now on, the focus is on a left-to-right HMM, in which a number of states form a sequence and from each state only transitions to itself and to the next state on the right are allowed.

2.3.2 Recognition using HMMs

This subsection shows how the probability of an observation sequence for a given HMM is calculated. Let $\mathbf{O} = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$ be the observation sequence; $\lambda = (A, B, \pi)$ be the parameter set of the HMM; and let $P(\mathbf{O}|\lambda)$ be the probability of $\mathbf{O}$, given the model λ. In principle, $P(\mathbf{O}|\lambda)$ is obtained by adding up the probabilities of all the possible state transitions, each of which can be expressed as a path in a two-dimensional plane (see Figure 2.3).

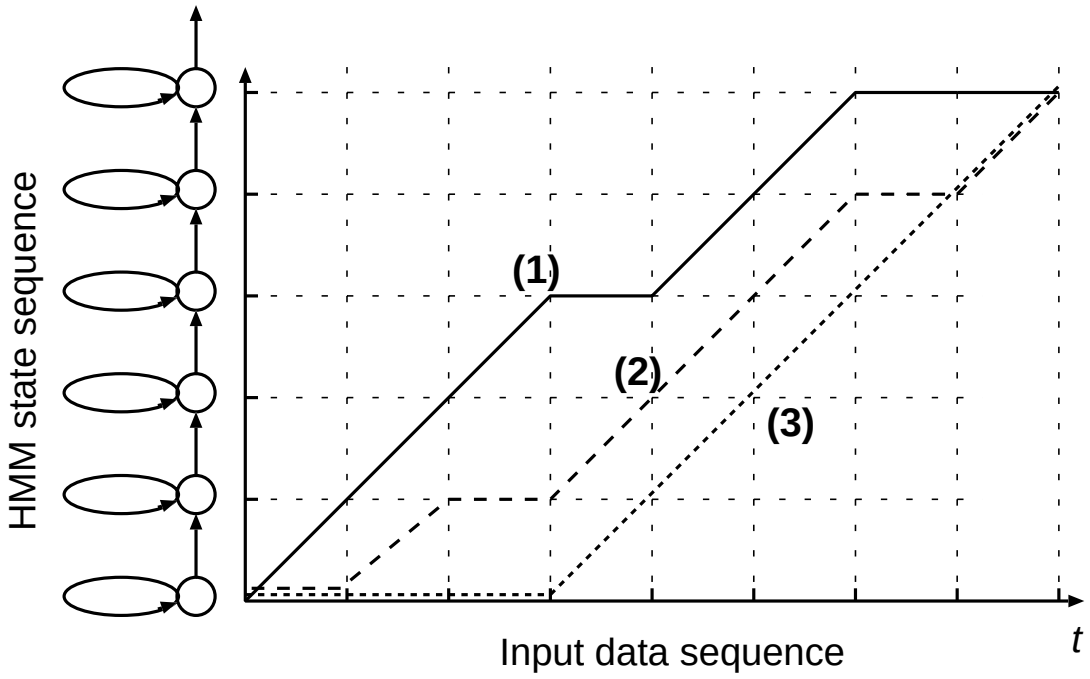

Figure 2.3: Forward algorithm.

There are two algorithms that can be used to calculate the probability: the Forward algorithm and the Viterbi algorithm. In the Forward algorithm the forward probability $\alpha_t(i)$ is defined as

$$\alpha_t(i) = P(\mathbf{o}_1, \ldots, \mathbf{o}_t, q_t = i|\lambda). \tag{2.20}$$

The forward probability is calculated as follows:

$$\alpha_1(i) = \pi_i b_i(\mathbf{o}_1), \quad 1 \leq i \leq S, \tag{2.21}$$

$$\alpha_{t+1}(j) = \left(\sum_{i=1}^{S} \alpha_t(i) a_{ij}\right) b_j(\mathbf{o}_{t+1}), \quad 1 \leq t \leq T-1, 1 \leq j \leq S, \tag{2.22}$$

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{S} \alpha_T(i). \tag{2.23}$$

This recursive process finally yields the probability $P(\mathbf{O}|\lambda)$.

In the Viterbi algorithm (see Figure 2.4) a maximization process is used instead of the summing procedure used in the Forward algorithm:

$$\delta_1(i) = \pi_i b_i(\mathbf{o}_1), \quad 1 \leq i \leq S, \tag{2.24}$$

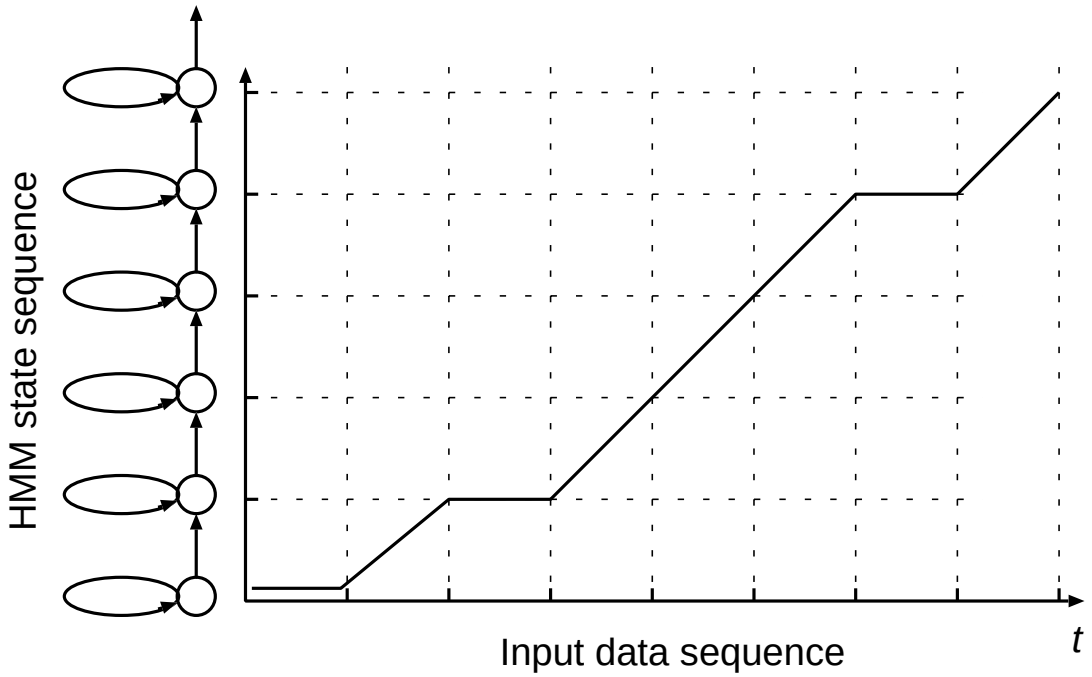

Figure 2.4: Viterbi algorithm.

$$\delta_t(j) = \max_{1 \le i \le S} (\delta_{t-1}(i) a_{ij}) b_j(\mathbf{o}_t), \quad 2 \le t \le T, 1 \le j \le S, \tag{2.25}$$

$$P^*(\mathbf{O}|\lambda) = \max_{1 \le i \le S} \delta_T(i). \tag{2.26}$$

Strictly speaking, the probability $P^*(\mathbf{O}|\lambda)$ obtained by the Viterbi algorithm is only an approximation of the probability obtained by the Forward algorithm. The Viterbi algorithm is often used, however, and it has been proved that the recognition accuracy obtained with the Viterbi algorithm is not significantly different from that obtained with the Forward algorithm. In the following, the Viterbi algorithm is used in the recognition process.

2.3.3 Estimation of HMM parameters

The ideal model parameter set is the one that maximizes the probability of the observation sequence. While no straightforward way to obtain the optimal parameter set is known, the *locally* optimized parameter set can be obtained by using the Expectation-Maximization (E-M) algorithm [11]. This parameter estimation using E-M algorithm is often referred to as the maximum likelihood (ML)

estimation.

Let $\mathbf{O} = (\mathbf{o}_1, \ldots, \mathbf{o}_t, \ldots, \mathbf{o}_T)$ be an observation sequence used for parameter estimation. As in the previous section, the forward probability $\alpha_t(i)$ is calculated for $t = 1, \ldots, T$, $i = 1, \ldots, S$. Additionally, the backward probability needs to be calculated. In a manner similar to that in which the forward probability is calculated, the backward probability $\beta_t(i)$ is calculated as follows:

$$\begin{aligned} \beta_T(i) &= 1, \qquad 1 \le i \le S, && (2.27) \\ \beta_t(i) &= \sum_{j=1}^{S} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j), \\ & \quad t = T-1, T-2, \ldots 1, \quad 1 \le i \le S, && (2.28) \end{aligned}$$

Then let $\gamma_t(i)$ be the probability of being in state i at time t, which is calculated as

$$\begin{aligned} \gamma_t(i) &= P(q_t = i | \mathbf{O}, \lambda) \\ &= \frac{P(\mathbf{O}, q_t = i | \lambda)}{\sum_{j=1}^{S} P(\mathbf{O}, q_t = j | \lambda)} \\ &= \frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^{S} \alpha_t(j) \beta_t(j)}. \end{aligned} \tag{2.29}$$

Given the model and the observation sequence, let $\xi_t(i,j)$ be the probability of being in state i at time t and state j at time $t+1$:

$$\begin{aligned} \xi_t(i,j) &= P(q_t = i, q_{t+1} = j | \mathbf{O}, \lambda) \\ &= \frac{P(q_t = i, q_{t+1} = j, \mathbf{O} | \lambda)}{P(\mathbf{O} | \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{S} \sum_{j=1}^{S} \alpha_t(i) a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}. \end{aligned} \tag{2.30}$$

One can easily see that

$$\gamma_t(i) = \sum_{j=1}^{S} \xi_t(i,j). \tag{2.31}$$

In the E-M algorithm the following auxiliary function $Q(\lambda', \lambda)$ is maximized in an iteration process:

$$Q(\lambda', \lambda) = \sum_{\mathbf{q}} P(\mathbf{O}, \mathbf{q} | \lambda') \log P(\mathbf{O}, \mathbf{q} | \lambda), \tag{2.32}$$

where the parameter set λ' is the current estimate for the HMM parameter set, and λ is the new estimate to be calculated. Since we can express P as

$$P(\mathbf{O}, \mathbf{q}|\lambda) = \pi_{q_0} \prod a_{q_{t-1}q_t} b_{q_t}(\mathbf{o}_t), \tag{2.33}$$

we can write $Q(\lambda', \lambda)$ as

$$Q(\lambda', \lambda) = Q_{\boldsymbol{\pi}}(\lambda', \boldsymbol{\pi}) + \sum_{i=1}^{S} Q_{\mathbf{a}_i}(\lambda', \mathbf{a}_i) + \sum_{i=1}^{S} Q_{\mathbf{b}_i}(\lambda', \mathbf{b}_i), \tag{2.34}$$

where $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_S\}$, $\mathbf{a}_i = \{a_{i1}, \ldots, a_{iS}\}$, $\mathbf{b}_i$ is the parameter vector that defines $b_i(\cdot)$, and

$$\begin{aligned} Q_{\pi}(\lambda', \boldsymbol{\pi}) &= \sum_{i=1}^{S} P(\mathbf{O}, q_0 = i|\lambda') \log \pi_i, & (2.35)\\ Q_{a_i}(\lambda', \mathbf{a}_i) &= \sum_{j=1}^{S} \sum_{t=1}^{T} P(\mathbf{O}, q_{t-1} = i, q_t = j|\lambda') \log a_{ij}, & (2.36)\\ Q_{b_i}(\lambda', \mathbf{b}_i) &= \sum_{t=1}^{T} P(\mathbf{O}, q_t = i|\lambda') \log b_i(\mathbf{o}_t). & (2.37) \end{aligned}$$

There are also the following stochastic constraints:

$$\begin{aligned} \sum_{j=1}^{S} \pi_j &= 1, & (2.38)\\ \sum_{j=1}^{S} a_{ij} &= 1, \qquad \forall i & (2.39)\\ \sum_{k=1}^{K} b_i(k) &= 1, \qquad \forall i & (2.40) \end{aligned}$$

where $b_i(k) = b_i(\mathbf{o_t} = \mathbf{v}_k)$.

Then the maximization leads to the model estimate $\bar{\lambda} = (\bar{\pi}, \bar{A}, \bar{B})$, where

$$\begin{aligned} \bar{\pi}_i &= \frac{P(\mathbf{O}, q_0 = i|\lambda)}{P(\mathbf{O}|\lambda)}, & (2.41)\\ \bar{a}_{ij} &= \frac{\sum_{t=1}^{T} P(\mathbf{O}, q_{t-1} = i, q_t = j|\lambda)}{\sum_{t=1}^{T} P(\mathbf{O}, q_{t-1} = i|\lambda)}, & (2.42)\\ \bar{b}_i &= \frac{\sum_{t=1}^{T} P(\mathbf{O}, q_t = i|\lambda)\delta(\mathbf{o}_t, \mathbf{v}_k)}{\sum_{t=1}^{T} P(\mathbf{O}, q_t = i|\lambda)}. & (2.43) \end{aligned}$$

Here

$$\begin{aligned} \delta(\mathbf{o}_t, \mathbf{v}_k) &= 1 \qquad \text{if } \mathbf{o}_t = \mathbf{v}_k\\ &= 0 \qquad \text{otherwise.} & (2.44) \end{aligned}$$

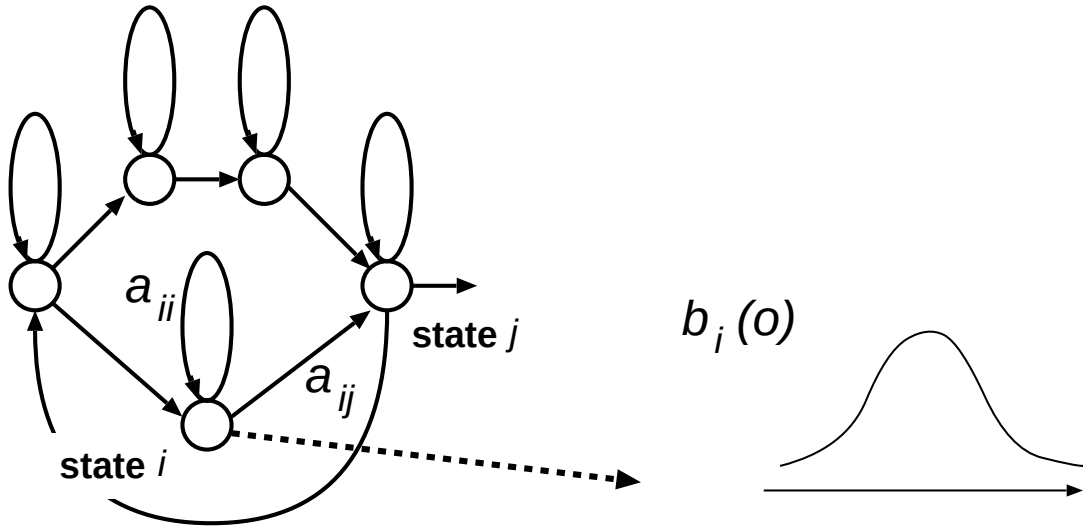

Figure 2.5: Continuous density hidden Markov model (CDHMM).

These re-estimation formulas can be rewritten as follows:

$$\bar{\pi}_i = \gamma_0(i), \tag{2.45}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} \xi_{t-1}(i,j)}{\sum_{t=1}^{T} \gamma_{t-1}(i)}, \tag{2.46}$$

$$\bar{b}_i = \frac{\sum_{t=1,\mathbf{o}_t=\mathbf{v}_k}^{T} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}. \tag{2.47}$$

Then the newly obtained λ is set to λ' and this process is repeated. This iteration process is stopped when the probability $P(\mathbf{O}|\lambda)$ converges.

2.3.4 Continuous-density HMMs

The output pdf of continuous-density HMMs (CDHMMs) is usually a mixture of the Gaussian distributions (Figure 2.5). In this case, the output probability $b_j(\mathbf{o})$ is

$$\begin{aligned} b_j(\mathbf{o}) &= \sum_{k=1}^{M} c_{jk} \mathcal{N}(\mathbf{o}|\boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})) \\ &= \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_{jk}|^{1/2}} (\mathbf{o}_t - \boldsymbol{\mu}_{jk})^T \boldsymbol{\Sigma}_{jk}^{-1} (\mathbf{o}_t - \boldsymbol{\mu}_{jk}), \end{aligned} \tag{2.48}$$

where $\boldsymbol{\mu}_{jk}$ is the mean vector and $\boldsymbol{\Sigma}_{ik}$ is the covariance for the pdf of mixture k at state j. The mixture weight c_{jk} for each mixture component has the following

constraint:

$$\sum_{k=1}^{M} c_{jk} = 1, \quad 1 \leq j \leq S. \tag{2.49}$$

The parameters c_{jk}, $\boldsymbol{\mu}_{jk}$, and $\boldsymbol{\Sigma}_{jk}$ are also estimated using the E-M algorithm:

$$\bar{c}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)}, \tag{2.50}$$

$$\bar{\boldsymbol{\mu}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \mathbf{o}_t}{\sum_{t=1}^{T} \gamma_t(j,k)}, \tag{2.51}$$

$$\bar{\boldsymbol{\Sigma}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)(\mathbf{o}_t - \boldsymbol{\mu}_{jk})(\mathbf{o}_t - \boldsymbol{\mu}_{jk})^T}{\sum_{t=1}^{T} \gamma_t(j,k)}, \tag{2.52}$$

where

$$\gamma_t(j,k) = \gamma_t(j) \frac{c_{jk} \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})}{\sum_{m=1}^{M} c_{jm} \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm})}. \tag{2.53}$$

Since it is generally known that CDHMMs perform better than DHMMs, we focus on CDHMMs from now on.

2.4 Large-vocabulary continuous-speech recognition

The previous section reviewed *isolated-word* speech recognition, in which a standard pattern is prepared for each word in the vocabulary and users utter only one word at a time. In this type of recognition, it is difficult to increase the number of words in the vocabulary because it is necessary to provide training data for each word. Furthermore, the recognition systems are not easy to use because it is not natural to speak only one word at a time. People usually utter phrases (concatenations of words) called *continuous* speech. This section therefore describes large-vocabulary continuous-speech recognition (LVCSR), in which the recognition vocabulary usually consists of more than 10,000 words.

2.4.1 Subword speech units

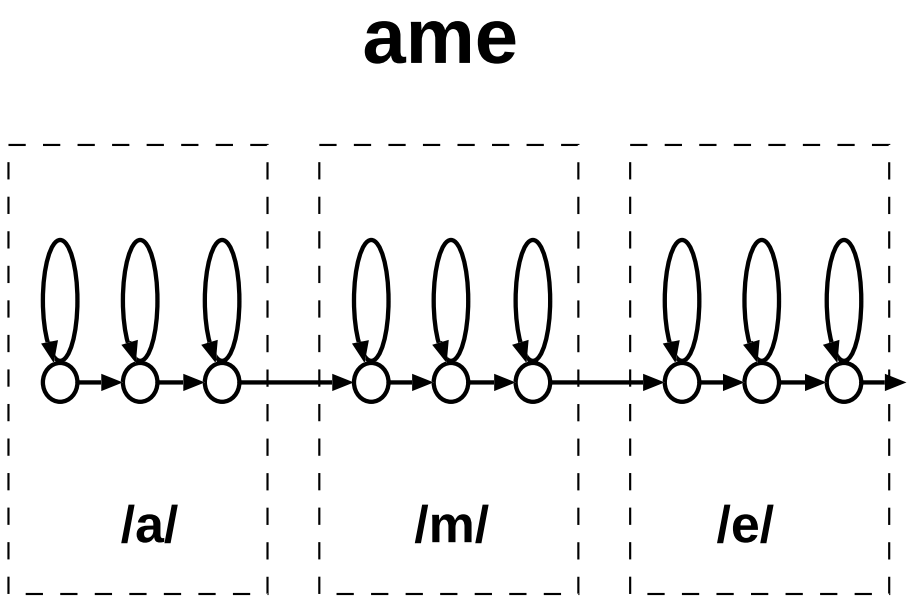

Figure 2.6: A model for "ame" using monophone HMMs.

There are many ways to define a set of *subword* speech units so that every word can be represented by their concatenation. Once such a set is given, a model for each unit can be defined and each word model can be represented by a concatenation of such models. Since it is not necessary to prepare training data for every word in the recognition vocabulary, large-vocabulary speech recognition becomes much easier.

A typical example of such a subword unit set is a set of *monophones*. A model is prepared for each phone, and every word model is a concatenation of phone models (Figure 2.6).

The features of a phone differ greatly from context to context and it is difficult to model all these differences by using a single monophone model. To overcome this problem, context-dependent phones, in which nearby (preceding or succeeding) phones are taken into consideration as the context, have come to be used as the subword units. In a context-dependent phone set, two units that correspond to the same phone but have different contexts are regarded as different units. Examples are *diphones*, in which the immediately right or the immediately left

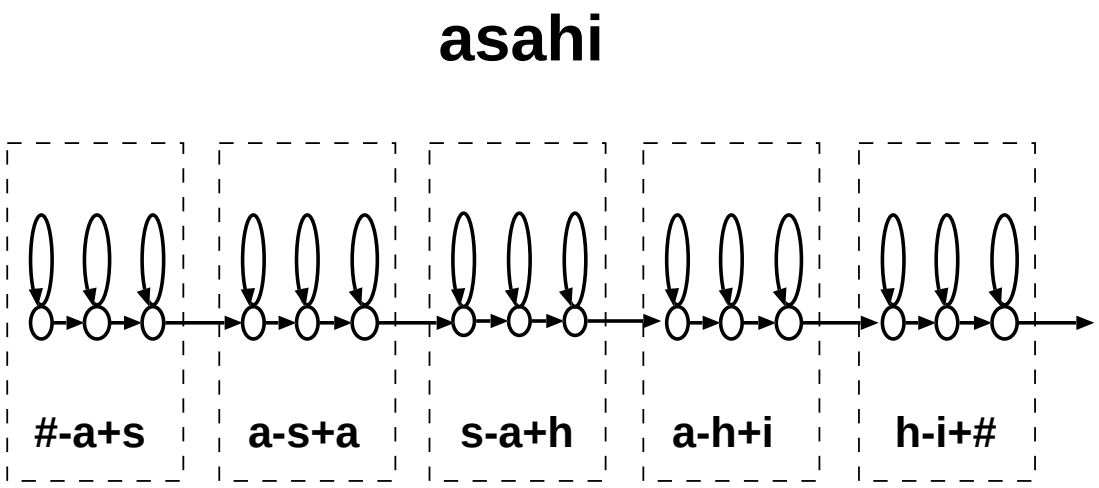

Figure 2.7: A model for "asahi" using triphone HMMs. For example, the symbol "s-a+h" stands for phone "a" whose preceding phone is "s" and whose succeeding phone is "h".

phone is taken into consideration, and *triphones*, in which both left and right phones are taken into consideration (see Figure 2.7).

2.4.2 Statistical language modeling

Let X be the observation and W be a word sequence. In the statistical recognition process, the word sequence which has the largest probability for the given data is chosen from all the possible word sequences,

$$\hat{W} = \mathrm{argmax}_W P(W|X). \tag{2.54}$$

Using Bayes' theorem, we can write:

$$P(W|X) = \frac{P(X|W)P(W)}{P(X)}. \tag{2.55}$$

Since $P(X)$ is the same for all W, Eq. (2.55) is rewritten as

$$\hat{W} = \mathrm{argmax}_W P(X|W)P(W). \tag{2.56}$$

So far in this chapter, we have dealt with isolated word recognition, in which it is assumed that each utterance consists of only one word and therefore the

recognition vocabulary $\{W\}$ is a set of isolated words. Additionally it is implicitly assumed that all words appear with the same probability; the probability $P(W)$ in Eq. (2.56) is the same for all words in the vocabulary.

In LVCSR, however, these assumptions are not realistic. First, a word sequence consists of more than one word, $W = \{w_1, w_2, \ldots, w_Q\}$, where Q is the number of words in W. Second, $P(W)$ differs among word sequences; some sequences are more likely to appear than others. The statistical model that gives $P(W)$ for each word sequence W is called a *language model* and its parameters are estimated from a very large linguistic corpus.

The probability $P(W)$ can be written as

$$\begin{aligned} P(W) &= P(w_1 w_2 \ldots w_Q) \\ &= P(w_1)P(w_2|w_1)\ldots P(w_Q|w_{Q-1}, \ldots, w_2, w_1). \end{aligned} \tag{2.57}$$

The probabilities for all the possible word sequences, are almost impossible to estimate from a limited amount of data. It is therefore often assumed that the probability of each word is affected only by the $(n-1)$ preceding words:

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) \simeq P_n(w_i|w_{i-1}, w_{i-2}, \ldots, w_{i-n+1}). \tag{2.58}$$

This model is called an n-gram model. Let $F(w_i, w_{i+1}, \ldots, w_{i+n-1})$ be the number of occurrences of the word sequence $w_i w_{i-1} \ldots w_{i-n+1}$ in the training corpus. Then the probability $P_n(w_i)$ for each word w_i is calculated as follows:

$$P_n(w_i|w_{i-1}, \ldots, w_{i-n+1}) = \frac{F(w_i, w_{i-1}, \ldots, w_{i-n+1})}{F(w_{i-1}, \ldots, w_{i-n+1})}. \tag{2.59}$$

Usually, a *bigram* in which $n = 2$ or a *trigram* in which $n = 3$ is used.

2.5 The data insufficiency problem in speech recognition

In speech recognition using CDHMMs, the recognition unit most widely used is the triphone, in which the right-hand phone and the left-hand phone are taken

into consideration as the context. The number of triphones is usually very large. In English, for example, the number of phones is roughly 50, so the number of triphones is about 50^3 (some combinations of phones are prohibited by the nature of the language). The number of states for each triphone HMM is usually three, and the number of mixture components for each state is about ten. Therefore the number of mean vectors in all the HMMs is $50^3 \times 3 \times 10 \simeq 4,000,000$. On the other hand, the amount of training data available is usually less than 100 hours' worth, which amounts to about forty million data frames when the frame interval for the short-term spectrum is 10 ms. Then the number of data samples per mean vector is $40,000,000 \div 4,000,000 = 10$. This number seems very small for precise estimation of the mean vectors. Furthermore, the training data is phonetically unbalanced; the number of occurrences of each triphone in the training data usually differs greatly from triphone to triphone, and some triphones do not appear in the training data. From these two reasons, it is clear that the data insufficiency problem greatly affects the recognition accuracy.

Many clustering methods have been developed in attempts to avoid this data insufficiency problem [34, 3, 31, 22, 73, 68, 74, 12, 37]. They reduce the number of parameters by grouping some of them. The parameters to be clustered are triphone units, states, and mixture components. These methods will be outlined in Section.3.1.

The trigram is often used for the language modeling in LVCSR systems. The trigrams are usually estimated from a large language corpus, typically comprising 10^8 words. Even if it is assumed that only 10^4 kinds of words are mainly used and the other kinds of words are ignored, the number of kinds of trigrams is $10^4 \times 10^4 \times 10^4 = 10^{12}$. This number is much larger than the number of trigrams in the language corpus, 10^8. It is clear that the data insufficiency problem is also inevitable in language modeling.

To tackle this problem, investigators have usually taken the following two approaches:

- Word clustering
- Smoothing with unigrams and bigrams

The first approach, word clustering, corresponds to the clustering approach in acoustic modeling: it groups *similar* words into one group and assumes that words in the same group have the same probability. The similarity measures usually used are mutual information [5] and Kullback-Leibler divergence (relative entropy) [39]. Other similarity measures are based on the structure of the language, such as those using information about parts of speech.

The second approach, smoothing, utilizes bigrams and unigrams, which are much less numerous and thus can be more precisely estimated from the same amount of data. Two methods used in this smoothing approach are the deleted interpolation method [23] and the back-off smoothing method [28]. The deleted interpolation method is a kind of cross validation in which the data are divided into training data and test data; it uses instead of the trigram probability the linear combination of the probabilities of the trigram, the bigram, and the unigram, and then these n-gram probabilities are estimated from the training data and the weighting factors among these n-gram probabilities are estimated so as to maximize the likelihood for the test data. The back-off smoothing method utilizes Good-Turing estimates [19], which are based on an empirical distribution of the number of occurrences of words in a natural language, for smoothing n-grams.

In both acoustic modeling and language modeling, the training procedure requires much computational time, since the amount of data available and the number of model parameters are both significantly large. We therefore need a way to obtain a model of optimal size without incurring excessive computational costs.

The data insufficiency problem also appears in adaptation, which is a technique of improving recognition performance by using a small amount of data and modifying the model parameters under test conditions. A number of utterances from a new user are used in acoustic model adaptation, and some recent writings

from a new user are used in language model adaptation. Often taken in both acoustic and language model adaptation is the Bayesian approach (e.g., [30]) in which the parameters of the initial model are used as *a priori* knowledge for the estimation of new parameters. Also often taken in acoustic model adaptation has been the transformation approach, in which some global transformations are employed and their parameters are estimated (e.g., [35]). Some of the acoustic model adaptation methods will be introduced in Section 4.1. An example of language model adaptation is the cache model approach, in which the parameters of the initial language model and those estimated from the sentences used by a new user are interpolated [29]. In the Bayesian approach, only the parameters corresponding to the data for adaptation are re-estimated. Therefore, as the model size increases, the adaptation tends to be less effective. It should also be noted that the computational cost of the model selection process in adaptation has to be very small, since the adaptation process should be completed almost instantaneously in actual applications.

2.6 Structural approach to data insufficiency problems in speech recognition

As mentioned in the previous section, context-dependent phone units such as triphones have recently come to be used to model units in speech recognition systems based on the use of HMMs. While most such systems cluster the HMM parameters (e.g., by subword clustering and state clustering) to control the HMM size and thus avoid poor recognition accuracy due to a lack of training data, none of them provide effective criteria for determining the optimal number of clusters. This thesis therefore describes a method in which state clustering is accomplished by using phonetic decision trees and in which the minimum description length (MDL) criterion is used to optimize the number of clusters. Large-vocabulary recognition experiments show that this method results in recognition more accu-

rate than that obtained when the maximum-likelihood estimation is used.

Maximum *a posteriori* (MAP) estimation is one Baysian approach and has been used for speaker adaptation in speech recognition systems using hidden Markov models [30, 18]. When the amount of data is sufficiently large, MAP estimation yields recognition performance as good as that obtained using the maximum-likelihood (ML) estimation explained in Subsection 2.3.3. This thesis describes a structural maximum *a posteriori* (SMAP) approach to improving the MAP estimates obtained when the amount of adaptation data is extremely small. The model parameter space is assumed to be structured hierarchically and the probability density functions for model parameters at one level are used as priors for those of the parameters at adjacent levels. Results of supervised adaptation experiments using non-native speakers' utterances showed that SMAP estimation reduced the error rate by half using only three utterances for adaptation, and that it yielded the same accuracy as conventional MAP and ML estimation when the amount of data was sufficiently large.

The acoustic modeling using the MDL criterion is described in the next chapter, and the SMAP adaptation method is described in Chapter 4.

Chapter 3

MDL-based Acoustic Modeling

3.1 Motivation

Over the past few years, extensive studies have been carried out on speaker-independent speech recognition systems that employ continuous density hidden Markov models. It is well known that in most such systems the use of context-dependent (CD) phone units (e.g., diphones, triphones) rather than context-independent (CI) phone units (monophones) provides greater recognition accuracy [1-10].

While the large number of CD models employed in a typical system can help to capture variations in speech data, the amount of available training data is likely to be insufficient to support the use of such a large number. Furthermore, there is great variation in the frequency with which individual CD phone units can be expected to appear in training data; in most CD phone unit sets, the frequencies for some units will be so small that they will be unlikely to appear in training data even when a very large amount of data is provided. Such lack of data can seriously degrade speech recognition performance and most recognition systems using CD models cluster the model parameters to try to alleviate the problem.

Various clustering methods have been developed for this purpose. One varia-

tion among them is the choice of parameter to be clustered: K.F. Lee *et al.* [34], for example, use subword clustering, Hwang *et al.* [22] use state clustering, and Digalakis et al. [12] cluster the mixture components of the HMMs with Gaussian-mixture state observation densities. There is also variation in the approach to selecting the acoustically-similar parameters to be clustered. One approach is to use only the acoustic characteristics of the data [73, 68, 22, 12, 37]. Another approach is to utilize *a priori* knowledge about acoustic similarities (usually represented in the form of decision trees) between the parameters, in addition to the acoustic characteristics themselves [34, 3, 31, 74].

However clustering is performed, the accuracy with which the acoustic similarities are measured will be extremely important. One of the most successful approaches in this regard is that based on the maximum-likelihood (ML) criterion (e.g., [74]). In this approach, a calculation is made for each parameter cluster in the model to determine the degree to which the splitting of that cluster would increase the likelihood of the model's outputting the training data; the cluster giving the greatest increase is then split. (Here, for the sake of simplicity, only the "splitting" method (i.e., top-down clustering) is considered, but an explanation of the application of ML to bottom-up clustering would be quite similar.)

The difficulty with the ML approach, however, is determining when to halt the splitting process, which could be carried on until the model simply consisted of a full set of individual, unclustered parameters. Most methods limit splitting by imposing a threshold value on the increase in the likelihood or on the number of parameter clusters, but the process required to optimize such thresholds (a series of recognition experiments; cross-validation; etc.) is computationally expensive.

In this chapter a new approach that uses the *minimum description length*(MDL) criterion [43] for state splitting [55, 58] is proposed. This MDL approach is effective for deciding when to stop splitting.

The MDL approach is based on an information

This chapter is organized as follows: Section 3.2 briefly reviews the MDL cri-

terion; Section 3.3 outlines state splitting using a phonetic decision tree; Sections 3.4 and 3.5 explain in detail how the MDL criterion is applied to state splitting; Section 3.6 describes the results of an experimental evaluation of the proposed method of state splitting. Finally, Section 3.7 discusses several issues related to the proposed method.

3.2 MDL Criterion

The MDL criterion [43] has been proven to be effective in selecting the optimal model from among various probabilistic models. It selects the model with the minimum description length for given data. When a set of models $\{1, \ldots, i, \ldots, I\}$ is given, the description length $l_i(\mathbf{x}^N)$ for data $\{\mathbf{x}^N = x_1, \ldots, \mathbf{x}_N\}$ and an underlying model i is given by

$$l_i(\mathbf{x}^N) = -\log P_{\hat{\boldsymbol{\mu}}^{(i)}}(\mathbf{x}^N) + \frac{K_i}{2}\log N + \log I, \tag{3.1}$$

where K_i is the dimensionality (the number of free parameters) of model i and $\hat{\boldsymbol{\mu}}^{(i)}$ represents the maximum likelihood estimates for the parameters $\boldsymbol{\mu}^{(i)} = (\theta_1^{(i)}, \ldots, \theta_{K_i}^{(i)})$ of model i. The first term on the right-hand side of (3.1) represents the code length for data $\mathbf{x}^N$ when model i is used as a probabilistic model. This term is identical to the negative of the log likelihood used in the ML criterion. The second term is related to the complexity of model i and the number of data samples, N. The third term is the code length required for choosing model i and is assumed here to be a constant. As a model becomes more complex, the value of the first term decreases and that of the second term increases. The second term works as a penalty imposed for employing a large model size (see Figure 3.1). In a comparison among models, the model with the shortest description length l may be considered the one having the most appropriate size and complexity. As may be seen in (3.1), the MDL criterion does not need any externally given parameters; the optimal model for the data is automatically obtained once a set of models has been specified. The derivation of the description length is briefly

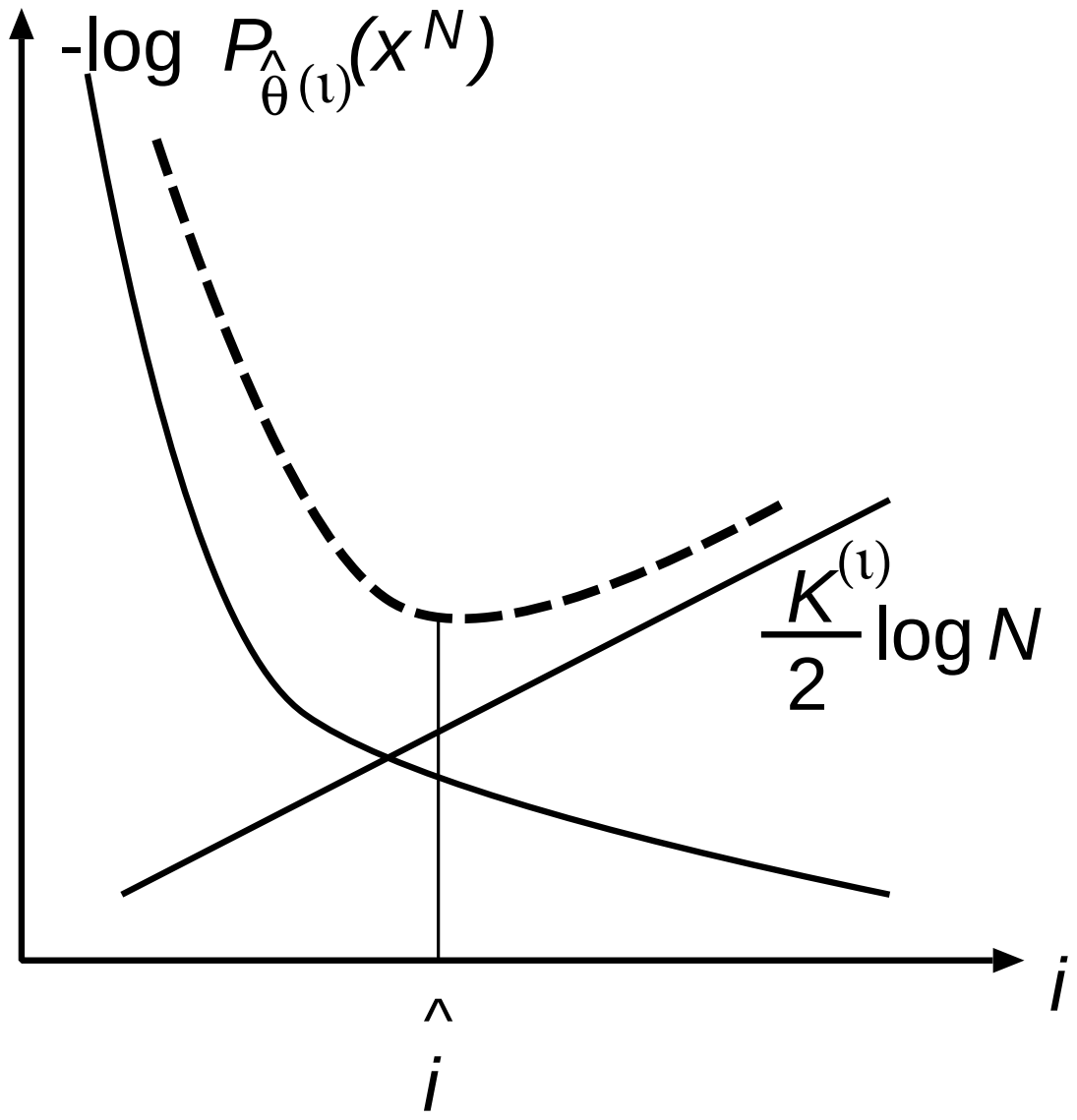

Figure 3.1: The MDL criterion.

summarized in Appendix A.

With complex models of the type used in speech recognition, it is often impractical to calculate the description length for all the possible models because to do so would involve high computational costs. To avoid this, a number of reasonable assumptions are introduced, which are explained in Section 4.

3.3 Tree-based state clustering

In this section, the outline of the proposed method is described.

For modeling CD phone units, triphones [46] are used, in which a central phone has left- and right-hand neighbors. Each triphone model is a left-to-right HMM in which states are placed in a line from the start state to the end state, and the transitions with respect to a state consist of that to itself and that to the next state to the right. The output density function for each state is a Gaussian probability density function (pdf) for which a diagonal covariance is assumed. All HMMs of triphones whose central phones are the same are assumed to have the same number of states.

As a clustering scheme, state splitting based on phonetic decision trees [74] is used. An example of the phonetic decision trees for triphones is shown in Figure 3.2. In the state splitting, those states at the same position in triphone HMMs having the same central phone are pooled into one set, and one phonetic decision tree is constructed for each set. Starting from the root node which represents the whole set, each node from top to bottom splits off into two other nodes representing, respectively, "yes" or "no" answers to such phonetic-context related questions as: "Is the previous phone unvoiced?" (*L-unvoiced?*) and "Is the next phone a fricative?" (*R-fricative?*) (see Figure 3.3). The MDL criterion is used to choose the optimal question to be asked at each node and to decide when to stop splitting. When all splitting has stopped, the pdf parameters of each leaf node are copied to the pdf parameters of the triphone states in the corresponding subset and used for recognition.

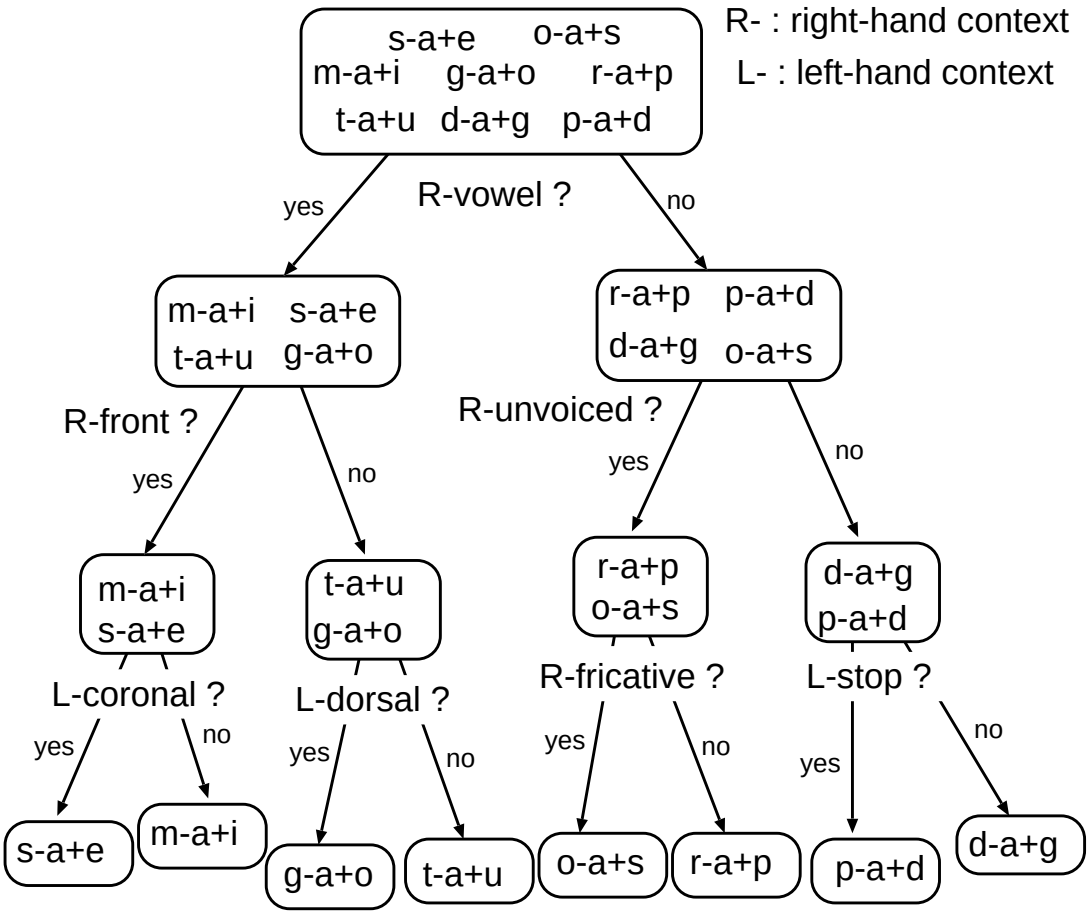

Figure 3.2: A phonetic decision tree for triphones. Eight triphones are split by phonetic-context related questions.

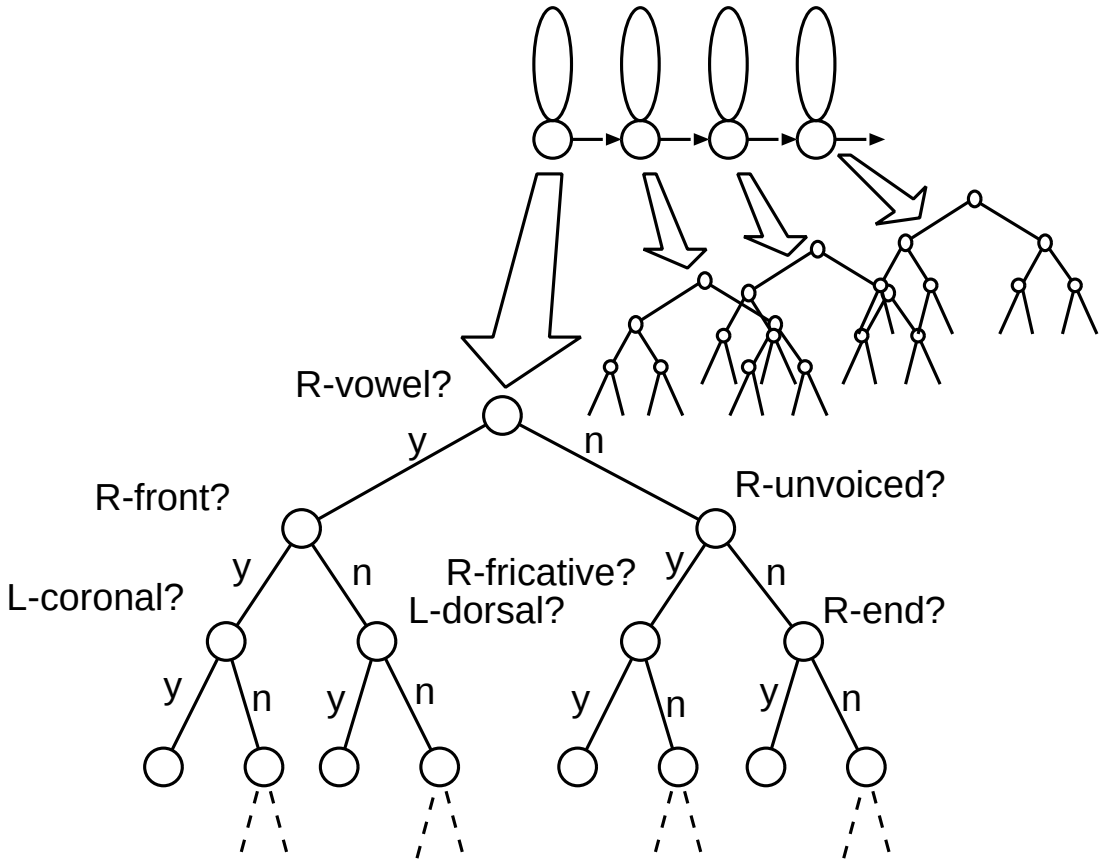

Figure 3.3: State splitting using phonetic decision trees.

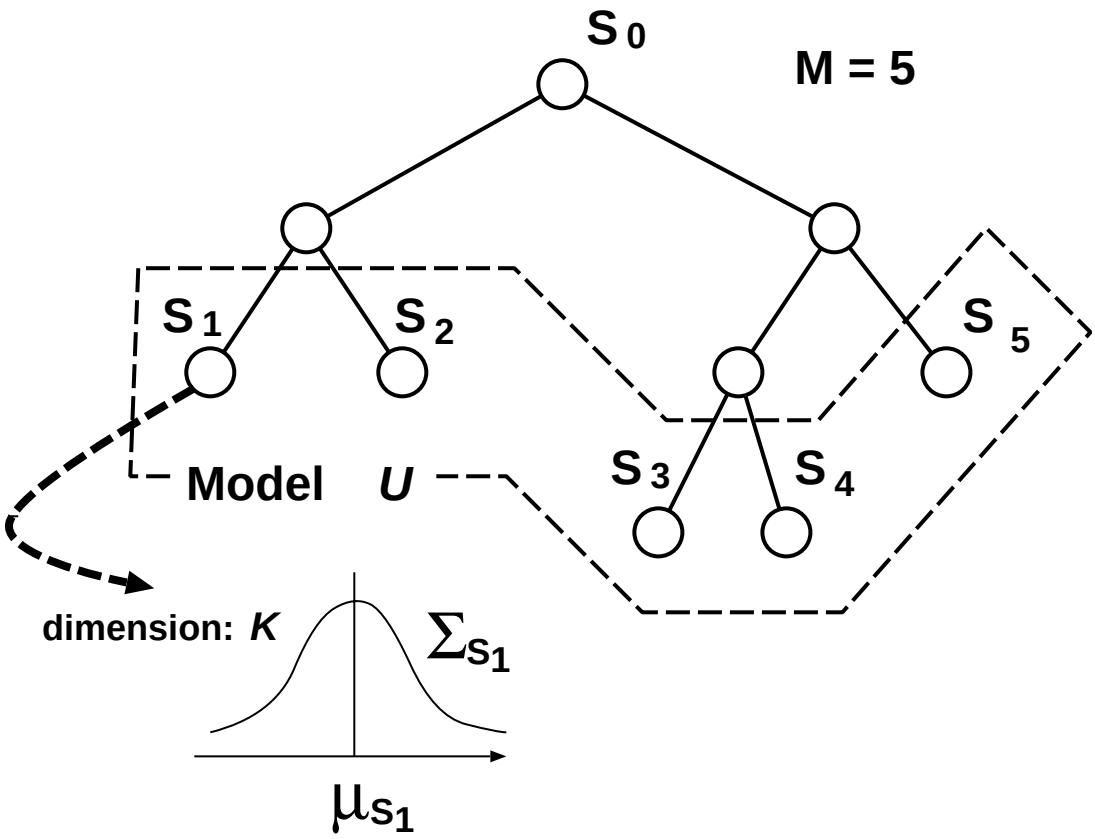

Figure 3.4: Model (node set) in the decision tree.

3.4 Description length for HMMs

3.4.1 Definition of a model set

As explained in Section II, the MDL criterion is used to select an optimal model from among a set of various models. Thus, it is first necessary to prepare the model set from which that optimal model is to be selected. For speech recognition using CDHMMs, it is impossible to prepare all the possible models because of the large number of possible structures of CDHMMs; the number of subword units and/or the number of states in each unit, for example, differ among recognition systems. In this study, the focus is on the clustering of the states in CDHMMs and constant values are given to those parameters unrelated to state clustering, such as the number of states in a single unit.

Here a *model* is defined as a node set in a phonetic decision tree in which a Gaussian pdf is assigned for each node. When the root node S_0, which represents the whole set of the triphone states in the tree, is split into M nodes, $S_1, \ldots, S_M$, as shown in Figure 3.4, one model $U(S_1, \ldots, S_M)$ is defined for the node set $\{S_1, \ldots, S_M\}$. Different node sets correspond to different models. The description length for each node set is calculated and the node set with the minimum description length is selected from among various node sets as being the optimum model.

3.4.2 Calculation of description length

Before clustering is performed, an estimate of each HMM parameter is calculated by using the Forward-Backward algorithm [41]. Let speech data for training consist of E examples and each example e be analyzed and represented by a time series of feature vectors, $\{\mathbf{o}_1^e, \ldots, \mathbf{o}_t^e, \ldots, \mathbf{o}_{T_e}^e\}$, where T_e is the number of data frames for example e. ML estimates for the Gaussian distribution at state s_l can then be written as:

$$\boldsymbol{\mu}_l = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_l(e,t)\mathbf{o}_t^e}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_l(e,t)}, \tag{3.2}$$

$$\boldsymbol{\Sigma}_l = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_l(e,t)(\mathbf{o}_t^e-\boldsymbol{\mu}_l)(\mathbf{o}_t^e-\boldsymbol{\mu}_l)^T}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_l(e,t)}, \tag{3.3}$$

where $\boldsymbol{\mu}_l$ is the mean vector and $\boldsymbol{\Sigma}_l$ is the covariance of the Gaussian distribution at state s_l, $(\mathbf{o}_t-\boldsymbol{\mu}_l)^T$ is the transpose of $(\mathbf{o}_t-\boldsymbol{\mu}_l)$, and $\gamma_l(e,t)$ is the *a posteriori* probability of the data being in state s_l at the t-th frame of example e, which is calculated as follows:

$$\gamma_l(e,t) = \frac{\alpha_l(e,t)\beta_l(e,t)}{\sum_{l'=1}^{L}\alpha_{l'}(e,t)\beta_{l'}(e,t)}, \tag{3.4}$$

where L is the total number of all the triphone states in the HMMs, $\alpha_l(e,t)$ is the forward probability, and $\beta_l(e,t)$ is the backward probability of state s_l at the t-th frame of example e.

The first term on the right-hand side of Eq. (3.1) is the negative of the log-likelihood of a probabilistic model with respect to given data. It is possible to

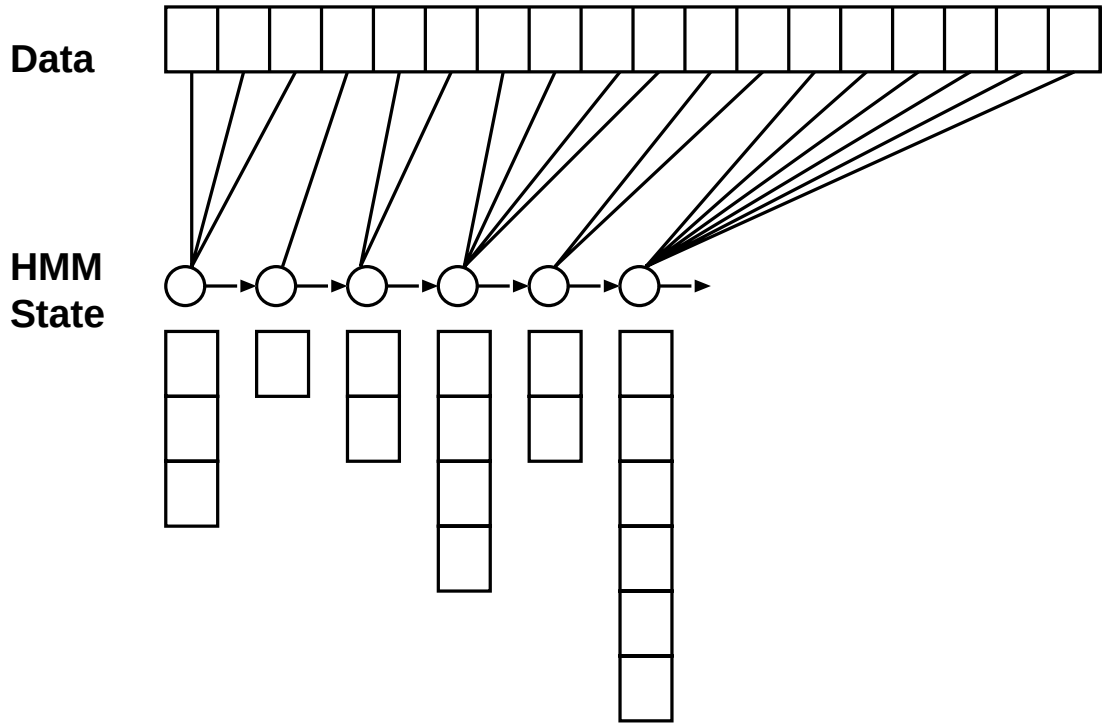

Figure 3.5: Viterbi alignment. Each data sample is assigned to one state.

calculate the log-likelihood of the training data for all the possible node sets, but to do so involves huge computational costs. To reduce these costs, the following three assumptions are introduced [74]:

1. The transition probabilities of HMMs can be ignored in the calculation of the log-likelihood for a node set.

2. State splitting does not change the frame/state alignment between the data and the model.

3. The log-likelihood of generating the data for each state is the sum of the log-likelihoods of generating each data frame, with each log-likelihood being weighted by the *a posteriori* probability of the data being in the state.

The third assumption is fully justified when the Viterbi algorithm is used for parameter estimation because in this algorithm the posterior probability is either one or zero (see Figure 3.5).

Let us next consider the problem of calculating the description length for the node set U defined in the previous subsection. All the triphone states pooled into the set corresponding the root node S_0 are renumbered as $\{s_1^1, \ldots, s_{L_1}^1, \ldots, s_1^m, \ldots, s_{L_m}^m, \ldots, s_1^M, \ldots, s_{L_M}^M\}$, where $\{s_1^m, \ldots, s_l^m, \ldots, s_{L_m}^m\}$ is the subset of states that are merged into node S_m, and L_m is the number of states in the subset. Using Eqs. (3.2) and (3.3), the mean vector and the covariance of state s_l^m are written as:

$$\boldsymbol{\mu}_l^m = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_l^m(e,t)\mathbf{o}_t^e}{\Gamma_l^m}, \tag{3.5}$$

$$\boldsymbol{\Sigma}_l^m = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_l^m(e,t)(\mathbf{o}_t^e-\boldsymbol{\mu}_l^m)(\mathbf{o}_t^e-\boldsymbol{\mu}_l^m)^T}{\Gamma_l^m}, \tag{3.6}$$

$$\Gamma_l^m = \sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_l^m, \tag{3.7}$$

where $\gamma_l^m(e,t)$ is the *a posteriori* probability of the data being in state s_l^m at the t-th frame of example e. Then, under the first and second assumptions, the ML estimates for the pdf parameters of node S_m are given by [26]

$$\begin{aligned}\boldsymbol{\mu}_m &= \frac{\sum_{l=1}^{L_m}\sum_{e=1}^{E}\sum_{t=1}^{T}\gamma_l^m(e,t)\mathbf{o}_t^e}{\sum_{l}^{L_m}\sum_{e=1}^{E}\sum_{t=1}^{T}\gamma_l^m(e,t)} \\ &= \frac{\sum_{l=1}^{L_m}\Gamma_l^m\boldsymbol{\mu}_l^m}{\sum_{l}^{L_m}\Gamma_l^m}, \end{aligned}\tag{3.8}$$

$$\begin{aligned}\boldsymbol{\Sigma}_m &= \frac{\sum_{l=1}^{L_m}\sum_{e=1}^{E}\sum_{t=1}^{T}\gamma_l^m(e,t)(\mathbf{o}_t^e-\boldsymbol{\mu}_l^m)(\mathbf{o}_t^e-\boldsymbol{\mu}_l^m)^T}{\sum_{l=1}^{L_m}\sum_{e=1}^{E}\sum_{t=1}^{T}\gamma_l^m(e,t)} \\ &= \frac{\sum_{l=1}^{L_m}\Gamma_l^m(\boldsymbol{\Sigma}_l^m+(\boldsymbol{\mu}_l^m)(\boldsymbol{\mu}_l^m)^T)}{\sum_{l=1}^{L_m}\Gamma_l^m} - (\boldsymbol{\mu}_m)(\boldsymbol{\mu}_m)^T, \end{aligned}\tag{3.9}$$

$$\gamma_m(e,t) = \sum_{l=1}^{L_m}\gamma_l^m(e,t), \tag{3.10}$$

where $\boldsymbol{\mu}_m$ is the mean vector and $\boldsymbol{\Sigma}_m$ is the covariance of the Gaussian distribution at node S_m. Then, from the third assumption, the approximated log-likelihood L of node S_m generating data $\{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$ is given by

$$\begin{aligned}L(S_m) &\simeq \sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_m(e,t)\log\left(\frac{1}{\sqrt{(2\pi)^K|\boldsymbol{\Sigma}_m|}}e^{-\frac{1}{2}(\mathbf{o}_t^e-\boldsymbol{\mu}_m)^t\boldsymbol{\Sigma}_m^{-1}(\mathbf{o}_t^e-\boldsymbol{\mu}_m)}\right) \\ &= -\sum_{e=1}^{E}\sum_{t=1}^{T_e}\frac{1}{2}\gamma_m(e,t)(K\log(2\pi)+\log|\boldsymbol{\Sigma}_m| \end{aligned}$$

$$
\begin{aligned}
& +(\mathbf{o}_t^e - \boldsymbol{\mu}_m)^t \boldsymbol{\Sigma}_m^{-1} (\mathbf{o}_t^e - \boldsymbol{\mu}_m)) \\
= \; & -\frac{1}{2} \Gamma_m (K + K \log(2\pi) + \log |\boldsymbol{\Sigma}_m|), \qquad (3.11) \\
\Gamma_m \; = \; & \sum_{t=1}^{T} \gamma_m(t), \qquad (3.12)
\end{aligned}
$$

where K is the dimensionality of the data vector $\mathbf{o}_t^e$, and Γ_m is the total state occupancy count at node S_m, which is the sum of $\gamma_m(e,t)$ over all data frames of all the examples. The log-likelihood of the data for all the nodes in set U is:

$$
\begin{aligned}
L_{all} \; = \; & \sum_{m=1}^{M} L(S_m) \\
\simeq \; & -\sum_{m=1}^{M} \frac{1}{2} \Gamma_m (K + K \log(2\pi) + \log |\boldsymbol{\Sigma}_m|). \qquad (3.13)
\end{aligned}
$$

The second term on the right-hand side of Eq. (3.1) represents the complexity of a model. In the proposed approach, it is assumed that the covariance of each Gaussian pdf is diagonal. The number of parameters to be estimated for model U is $2KM$ (with model U containing M mean vectors and M diagonal covariances). The total number of data samples is the sum of $\Gamma(S_m)$ over m. With this total, the second term may be approximated as:

$$
R \; = \; KM \log W, \qquad (3.14)
$$

where $W = \sum_{m=1}^{M} \Gamma_m$. As has been previously noted, the third term on the right-hand side of (3.1) is fixed at a constant value, C, for all possible models.

Finally, using (3.13) and (3.14), the description length for model U is calculated as follows:

$$
\begin{aligned}
l(U) \; = \; & -L_{all} + R + C \\
\simeq \; & \sum_{m=1}^{M} \frac{1}{2} \Gamma_m (K + K \log(2\pi) + \log |\boldsymbol{\Sigma}_m|) \\
& + KM \log W + C. \qquad (3.15)
\end{aligned}
$$

3.5 State splitting using the MDL criterion

In order to get an optimal model, it is needed to calculate description lengths for all possible models, which would involve prohibitively high computational costs. Instead, an algorithm that obtains only a suboptimal solution is used.

Let us first assume that node S_m of model U splits into two nodes S_{mqy} and S_{mqn},in response to question q, and then let $\Delta_m(q)$ be the difference between the description lengths after the splitting and before it (i.e., $l(U') - l(U)$). The description length of model U' is:

$$\begin{aligned} l(U') &= \sum_{m'=1,m'\neq m}^{M} \frac{1}{2}\Gamma_{m'}(K + K\log(2\pi) + \log|\mathbf{\Sigma}_{m'}|) \\ &\quad + \frac{1}{2}\Gamma_{mqy}(K + K\log(2\pi) + \log|\mathbf{\Sigma}_{mqy}|) \\ &\quad + \frac{1}{2}\Gamma_{mqn}(K + K\log(2\pi) + \log|\mathbf{\Sigma}_{mqn}|) \\ &\quad + K(M+1)\log W + C, \end{aligned} \tag{3.16}$$

where the number of nodes for U' is $M+1$, Γ_{mqy} is the state occupancy count for node S_{mqy}, and Γ_{mqn} is that for node S_{mqn}. The difference $\Delta_m(q)$ will then be given by the following equation:

$$\begin{aligned} \Delta_m(q) &= l(U') - l(U) \\ &= \frac{1}{2}(\Gamma_{mqy}\log|\mathbf{\Sigma}_{mqy}| + \Gamma_{mqn}\log|\mathbf{\Sigma}_{mqn}| \\ &\quad - \Gamma_m\log|\mathbf{\Sigma}_m|) + K\log W. \end{aligned} \tag{3.17}$$

In state splitting, the question q' which would minimize $\Delta_0(q')$ when used to split root node S_0 is first determined. If $\Delta_0(q') > 0$, then no splitting is conducted. If $\Delta_0(q') < 0$, then node S_0 is split into two nodes, $S_{q'y}$ and $S_{q'n}$, and the same procedure is repeated for each of these two nodes. This node splitting is carried out until there remain no nodes to be split and is conducted for the root nodes of all the phonetic decision trees in all the HMMs.

For the purpose of comparison, let us also consider here the ML approach [74]. Letting δ_m(q) be the increase in the log-likelihood when node S_m is split into two

in response to using question q,

$$\begin{aligned}\delta_m(q) &= L(S_{mqy}) + L(S_{mqn}) - L(S) \\ &= -\frac{1}{2}(\Gamma_{mqy} \log \mid \mathbf{\Sigma}_{mqy} \mid + \Gamma_{mqn} \log \mid \mathbf{\Sigma}_{mqn} \mid \\ &\quad -\Gamma_m \log \mid \mathbf{\Sigma}_m \mid). \end{aligned} \tag{3.18}$$

In the ML approach, question q' which would maximize $\delta_0(q')$ is first chosen from among all the questions, and then it is used to split root node S_0 into two nodes $S_{0q'y}$ and $S_{0q'n}$. This splitting process will continue until stopped by some externally given parameters used to control the number of clusters, since the increase δ is positive in all the splitting. Most methods apply a threshold value to the total occupancy count Γ_m and/or to the log-likelihood increase $\delta_m(q)$. However, the optimization of these parameters requires a series of recognition experiments which are computationally expensive and require additional data. The MDL approach needs no external control parameters; the term $K \log W$ in (3.17) corresponds to the threshold for increase δ in (3.18), and this term is estimated automatically on the basis of the training data. Additionally, the threshold term $K \log W$ is specified for each phone in the MDL approach, while the threshold *delta* is shared among all the phones. This indicates that the MDL approach is more robust against the data imbalance among phones than the ML approach.

3.6 Experiments

The proposed method was evaluated in experiments testing the recognition of 5000 Japanese words. Each utterance was digitized at a sampling rate of 16 kHz, and analyzed in 10-ms frame periods. The analysis yielded a vector of 21 components (a power derivative, 10 mel-scaled cepstral coefficients, and 10 corresponding mel-scaled cepstral time derivatives). 37 Japanese CI phones were used, from which 4309 triphones were derived. The number of states in each HMM was set to four. A Gaussian output pdf with a diagonal covariance was

assumed for each state. The number of questions used in the node splitting was 106. Two data sets, Data A and Data B, were prepared for training. Data A consisted of 250 phonetically-balanced words uttered by each of 46 male speakers. Data B consisted of 2150 phonetically-balanced words uttered by each of 36 male speakers. Speech data from five other male speakers, none of whom was involved in the production of Data A or Data B, was used for the evaluation tests. Each of these test speakers uttered 250 words. None of the words in the test vocabulary were used in the training vocabulary.

Table 3.1 shows recognition results obtained with the proposed MDL method (averaged for the five test speakers) and those obtained with the ML approach when Data A was used for training. The various results for the ML approach reflect the different values used for two thresholds, D and V, for the state occupancy count and for the increase in likelihood, respectively.

Let us consider a specific instance in the ML approach. For a node, S, the algorithm determines the set of questions for which neither of the resulting two response nodes would have a "total occupation count" less than or equal to D, and for which $\delta(q)$ would be larger than V. It determines the question among this set for which δ is largest and used it to split node S.

Table 3.1 shows the results of the ML approach for 17 combinations of D and V(ML 1–17). As shown in Table 3.1, the proposed method achieved higher recognition accuracy than any using the ML approach. In each instance, the computational cost required for the proposed method was roughly the same as that with the ML approach. In order to determine a model of a size optimal for the amount of training data, however, the ML approach must be performed repeatedly over a range of parameter values. Therefore, the total computational cost is much less with the MDL approach proposed here.

Table 3.2 shows the frequency of the questions used, summed over all the phonetic decision trees of all the HMMs, when the MDL approach was employed. "*L-begin*" corresponds to the question, "Is the phone located at the beginning of

Table 3.1: MDL and ML performance.

	D	V	# of nodes	Recog. rate(%)
MDL	–	–	2069	80.4
ML 1	60	0	3739	75.4
ML 2	100	0	3000	76.4
ML 3	200	0	2001	76.7
ML 4	300	0	1943	75.4
ML 5	400	0	1200	73.4
ML 6	500	0	1018	71.9
ML 7	1000	0	591	66.6
ML 8	60	200	2777	76.2
ML 9	60	400	2034	77.0
ML 10	60	600	1488	77.8
ML 11	60	800	1248	77.9
ML 12	60	1000	1071	77.4
ML 13	200	200	1782	77.3
ML 14	200	400	1533	77.2
ML 15	200	600	1326	77.0
ML 16	200	800	1142	77.8
ML 17	200	1000	1049	76.5

Table 3.2: Distributions of questions asked.

Vowel		Consonant	
L-coronal	67	L-begin	130
L-dorsal	55	L-back	69
L-begin	40	R-a	63
R-coronal	39	R-high	62
L-h	35	L-high	60
L-back	34	L-a	53
L-sonorant	33	L-front	45
R-dorsal	31	R-e	34
L-unvoiced	27	R-back	32
L-n	27	L-e	30
L-fricative	27	L-consonant	28
Total	1110	Total	822

Table 3.3: Recognition rates (%) for Data A and Data B.

Training set	Data A	Data B
No. of nodes	2069	6223
Male 1	72.8	84.8
Male 2	76.8	84.4
Male 3	89.2	92.4
Male 4	81.6	83.6
Male 5	81.6	84.8
Average	80.4	86.0

the word?". Questions related to left phones were used more often than those related to right phones. For a consonant, the question most frequently used was whether or not it was located at the beginning of a word.

Let us next consider how the optimal model size changes as the amount of data increases. Table 3.3 shows results for Data A and Data B. Data B, which is seven times larger than Data A, resulted in roughly a threefold increase in the number of nodes.

In order to evaluate the optimality of this size, A weight coefficient c was added to the second term on the right-hand side of (3.15), which results in:

$$\begin{aligned} l''(U) &= \frac{1}{2}\sum_{m=1}^{M} \Gamma_m (K + K\log(2\pi) + \log \mid \mathbf{\Sigma}_m \mid) \\ &\quad + cKM \log W + C. \end{aligned} \tag{3.19}$$

As c increases, so does the penalty for a large model size. Table 3.4 shows results for a range of c values of from 0.1 to 10.0. While the highest recognition accuracy was achieved for a c value of 2.0, it was only 0.7% higher than that for $c = 1$ (i.e., for the case in which the penalty for increased mode size is the same as that in (3.15), which expresses the description length in the proposed approach.)

Data B was also used to evaluate the recognition performance when single-

Table 3.4: Recognition rates (%) as a function of coefficient c.

c	0.1	0.5	1.0	2.0	4.0	10.0
No. of nodes	13927	9798	6223	3949	2418	1341
Male 1	84.0	84.4	84.8	83.6	82.4	79.6
Male 2	81.6	83.6	84.4	84.4	84.8	80.8
Male 3	92.0	92.0	92.4	92.8	92.4	91.2
Male 4	84.8	85.2	83.6	85.2	84.8	82.0
Male 5	84.4	84.4	84.8	87.6	85.2	86.8
Average	85.4	85.9	86.0	86.7	85.9	84.1

Table 3.5: Recognition rates (%) with mixture-Gaussian output pdfs.

	1 Gauss	2 Gauss
Male 1	84.8	86.8
Male 2	84.4	87.6
Male 3	92.4	94.4
Male 4	83.6	88.8
Male 5	84.8	87.2
Average	86.0	89.0

Gaussian output pdfs were replaced with mixture-Gaussian output pdfs. In this experiment, the number of Gaussian pdfs assigned to each state was increased to two, and the model was re-trained using the same training data. The increase in recognition rates shown in Table 3.5, indicates that further splitting of some of the nodes in models constructed using the MDL criterion might result in improved recognition rate if a better set of questions were prepared beforehand. Such a set might include, for example, questions regarding second-to-left and/or second-to-right phones, characteristics of individual speakers, recording conditions, etc.

3.7 Discussion and summary

A significantly useful method of optimizing the model size without using any externally given parameters was proposed. In an evaluation test it resulted in recognition more accurate than that obtained when a conventional approach was used and it had a much lower overall computational cost.

In the real-world application of speech recognition, the model size often must be small enough to operate in real time using limited hardware(CPU, memory). It would seem that this MDL approach is not applicable to this case because no hardware limitations are taken into consideration. However, this approach can be applied to this case by controlling the weight of each data sample in training data. For example, when the weight is assumed to be halved, the amount of data for each state is halved. Although one control parameter (the weight of each data sample) has to be introduced, the advantage of our approach in that the model size for each phone is controlled individually still remains valid.

A number of problems remain to be solved, however. First, the degree to which the assumptions implicit in the proposed method affect its performance with regard to the control of model size has to be determined. A second problem is that the set of models provided beforehand may not include the most optimal model ("true model") for the given data. A third problem is that, since it is assumed that the amount of data is sufficiently large in the MDL criterion, it

may not apply to the case where the amount of data available is small. These latter two problems are of course true not only for the proposed method but also for other model selection strategies using the MDL criterion, and further theoretical research addressing these problems is needed. A fourth problem is that the minimization of the description length does not necessarily minimize recognition error. Conventional ML approaches encounter the same problem: maximization of likelihood does not necessarily minimize recognition error. The MDL criterion has an advantage over the ML criterion in that it has an effective penalty used for model size control, one that has good theoretical support.

Two other widely known information criteria used for controlling model size are the Bayesian information criterion (BIC) [45] and the Akaike information criterion (AIC) [1]. The formula for the BIC is

$$l_i^{BIC}(\mathbf{x}^N) = -\log P_{\hat{\boldsymbol{\mu}}^{(i)}}(\mathbf{x}^N) + \frac{K_i}{2}\log N. \tag{3.20}$$

Comparing this criterion with the MDL criterion (Eq.3.1), one can easily see that the first and the second terms are identical and that the only difference is that the MDL criterion has a third term. Since throughout this thesis the third term is assumed to be constant, the BIC gives exactly the same results as the MDL criterion here. After the result of our research was first published [55], the approach using the BIC to control the model size in speech recognition was extensively studied [6, 7, 8, 70]. It has been successfully applied to speaker clustering [6], Gaussian mixture modeling [7], modeling of mixture of Gaussian pdf for HMM [8], and segmentation of speech data [70]. Since the BIC gives exactly the same results as the MDL criterion does, the results of these studies strongly support the effectiveness of our approach. They also proved that our approach can be applied to many other data insufficiency problems in speech recognition.

In the AIC, the second term in (3.1) is replaced by K_i and there is no third term:

$$l_i^{AIC}(\mathbf{x}^N) = -\log P_{\hat{\boldsymbol{\mu}}^{(i)}}(\mathbf{x}^N) + K_i. \tag{3.21}$$

Practically speaking, it is well known that in many applications the results given by the AIC differ little from those given by the MDL criterion. The MDL criterion and the AIC are therefore not compared in this thesis. In theory the difference between the MDL criterion and the AIC is still controversial but it is not discussed here because it is not an important issue here. One typical claim supporting the MDL criterion is that the AIC tends to overestimate the number of parameters needed [47]; while the AIC is likely to select the correct model when the complexity of the true model grows with sample size [48, 49], such a case is unlikely to happen in actual applications.

Recently a new information criterion was proposed: the subspace information criterion (SIC) for model selection from the functional analytical viewpoint [64]. Since it was reported to work well even when the number of training examples is small, it may be promising to apply this criterion to acoustic modeling.

Chapter 4

Structural MAP Approach to Speaker Adaptation

4.1 Motivation

Automatic speech recognition systems using continuous density hidden Markov models (HMMs) have been recently used in various applications. Speaker-independent (SI) systems are typically constructed using speech samples collected from many speakers. It has been reported, however, that the performance of SI HMMs is often degraded when there is a mismatch between the training and testing environments. For example, when the acoustic characteristics of a new speaker are very different from those of the speakers in the training data, the recognition accuracy for the new speaker might be far below the average accuracy. Other major adverse conditions causing mismatches are those due to different microphones, channels, and noise environments.

Many techniques compensating the degradation caused by mismatches have been developed. They are roughly grouped into two categories, namely: (1) *feature compensation* (e.g., [32]), in which the process of feature extraction is modified; and (2) *model adaptation* (e.g., [18, 35]), in which the parameters of recognition models are adjusted. Although combining these two techniques has

been shown effective (e.g., [44]), the discussion is focused on model adaptation in the present study . It is desirable that adaptation improves speech recognition accuracies even when little adaptation data is given and more importantly it yields performance equal to or better than that obtained using *maximum likelihood* (ML) estimation when enough data is available. Few methods, however, achieve both objectives.

The most popular approach to model adaptation is through Bayesian formulation. For example, *maximum a posteriori* (MAP) estimation algorithms (e.g., [30, 18]) have been widely adopted recently and successfully applied to speaker adaptation. In this method the model parameters are regarded as *random variables* whose *joint prior probability density function* (pdf) is assumed. The MAP estimate of the parameter vector is defined as the mode of the posterior pdf given the adaptation data. The improvement obtained with MAP estimation is significantly larger than that obtained with ML estimation, especially when the amount of adaptation data is small. It is well known, since MAP estimates are *asymptotically equivalent* to ML estimates, that the resulting recognition performance is similar to that of speaker-dependent (SD) HMMs when the amount of data becomes large. A quasi-Bayes approach [20] has also been adopted to handle *on-line* MAP adaptation. In these conventional MAP estimation methods, HMM parameters of different speech units are often assumed to be independent. Therefore, each model can be adapted only if the corresponding speech unit has been observed in the current set of adaptation data. The improvement is consequently rather small when the amount of adaptation data is extremely limited. The MAP estimation is briefly explained in Appendix B.

Another category of adaptation techniques, which do not use the MAP framework, are often referred to as *transformation-based* approaches, such as *cepstrum mean normalization* (CMN) [2], *signal bias removal* (SBR) [42], *maximum likelihood linear regression* (MLLR) [35], *spectral interpolation* [50, 51, 52], *vector field smoothing* (VFS) [36], *stochastic matching* (SM) [44], *nonlinear stochastic*

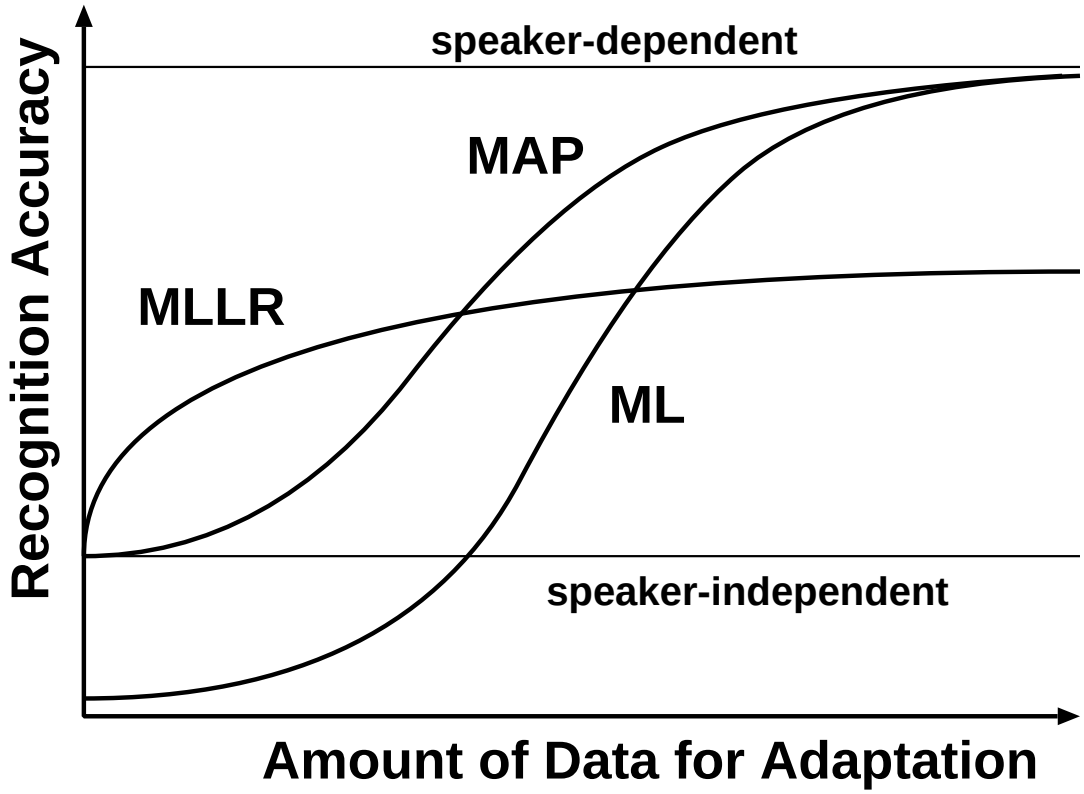

Figure 4.1: Recognition performance of maximum *a posteriori* (MAP) adaptation, maximum likelihood linear regression (MLLR), and maximum likelihood estimation (ML).

matching [65] and *predictive adaptation* [66]. This family of techniques limits the number of free parameters by tying the HMM parameters or by applying some constraints on the parameters in order to improve recognition accuracies with a small amount of data. When the amount of adaptation data exceeds a certain value, however, the recognition accuracy often becomes inferior to that obtained with ML estimation of the model parameters. This is because a model with a small number of free parameters could not fully utilize the potential information embedded in the large amount of data. In Figure 4.1, the difference between the recognition performance of MAP and that of MLLR is shown.

Because the traditional MAP approaches and the transformation-based methods are not capable of either improving recognition accuracy when little data are available or exploiting the information in a large amount of data, several algorithms supplementing those techniques have been developed. The *extended MAP* (EMAP) method [63, 75], and the *quasi-Bayes* technique with *correlated* mean vectors [21] are extensions of the traditional MAP approaches. They increase the recognition rates obtained with a small amount of data by taking into account the *a priori* knowledge in the correlation between the parameters modeling different speech units. For example, the pair-wise correlation between the mean vectors could be used to enhance estimation of the mean parameters of some speech units even if they are not directly observed in the adaptation data and therefore the recognition rates are significantly improved [21]. Although these methods are in theory quite general, they need to impose some approximation in practice because it is difficult to estimate such correlations precisely when the amount of training data is small. In [63, 75], for example, the model parameter space was divided into several subspaces, the ideal number often depends on the amount of adaptation data available. It is also possible to extend the known ML techniques, such as MLLR to incorporate the MAP estimation criterion. The recently proposed *maximum a posteriori linear regression* (MAPLR) [60] algorithm improves MLLR in a way similar to MAP enhancement over ML for HMM parameter estimation.

Combinations of MAP and transformation-based approaches have also been studied intensively ([13, 9, 67, 69]). Notable examples were in combining MLLR and MAP [13] and combining MAP and VFS [67, 69]. Chien *et al.* [9] reported that significantly better recognition accuracy can be obtained by combining MAP and SM with an iterative process. The shortcoming of these combined methods is again the use of fixed *structures*, i.e. fixed ways of parameter tying, in the acoustic space. Therefore they have only been shown useful with adaptation data sizes within a narrow range. To alleviate this problem, a *tree structure* has been used in adjusting the number of layers in a tree and the degree of parameter tying according to the amount of available data (e.g., [53, 54]).

In this study, the nice asymptotic property of MAP estimation for large size adaptation is taken advantage of and the flexible parameter tying strategy in a tree for small sample adaptation and formulate a novel *structural Bayes adaptation* framework that achieves the two desired objectives mentioned earlier. By assuming that the prior knowledge in a tree node can be used to construct prior density needed for MAP estimation of all the parameters in the successive child nodes, a new *structural maximum a posteriori* (SMAP) algorithm [56, 57, 59] is introduced for speaker and environment adaptation.

Three key steps are required in formulating the proposed SMAP approach. They are described in the next three Sections. First, a tree with a uniform structure is needed to characterize the acoustic space represented by the HMM parameters. In this study an information-theoretical criterion is used to cluster all the Gaussian mixture component densities typically used to model state observation densities in HMM. This procedure is discussed in detail in Section 4.2 Next, given all the density clusters used to characterize nodes in a tree, it is needed to find a Gaussian density to summarize all the Gaussian components in the cluster so that the likelihood of a sequence of observation vectors representing the adaptation data can be evaluated at the node level and therefore the MAP estimate at any node in the tree can be computed. In Section 4.3, a summarizing

procedure that simplifies the preparation for SMAP estimation is introduced and proves to be effective for speech recognition. For the third step, the prior density at each tree node needs to be defined. In order to use every observation sample to estimate all the HMM parameters, a *hierarchical prior evolution approximation* is used by assuming that the *hyperparameters* charactering the prior density at each node are evaluated based on the knowledge embedded in the prior density of its parent node. This process is explained in Section 4.4. Once the three key steps are established, the SMAP estimation algorithm is then derived in Section 4.5.

The proposed SMAP approach was evaluated on the RM (Resource Management) task [40]. Training/adaptation and testing utterances by non-native speakers were collected over two different acoustic conditions, a desktop microphone and a telephone handset through dial-up lines. The effectiveness of the SMAP algorithm was demonstrated in a set of supervised and unsupervised adaptation experiments. The ways to combine fast supervised adaptation and on-line unsupervised adaptation to achieve a sufficient recognition accuracy in real applications were also investigated. The experimental results with different adaptation scenarios in these adverse conditions is reported in Section 4.6. Finally, the findings are summarized in Section 4.7.

4.2 Tree structure

The definition of a structure to aid MAP estimation is a key procedure in the proposed structural Bayes approach. In this study a tree structure is adopted because it offers a natural evolution of prior knowledge embedded in the parent-child relationship between nodes at different tree layers (see Figure 4.2). There exists many ways to generate such a tree that models a structure of the acoustic space of interest.

Given the set of all the mixture Gaussian components in the set of HMMs, it is needed to first define a distance measure, $d(m, n)$, between Gaussian components,

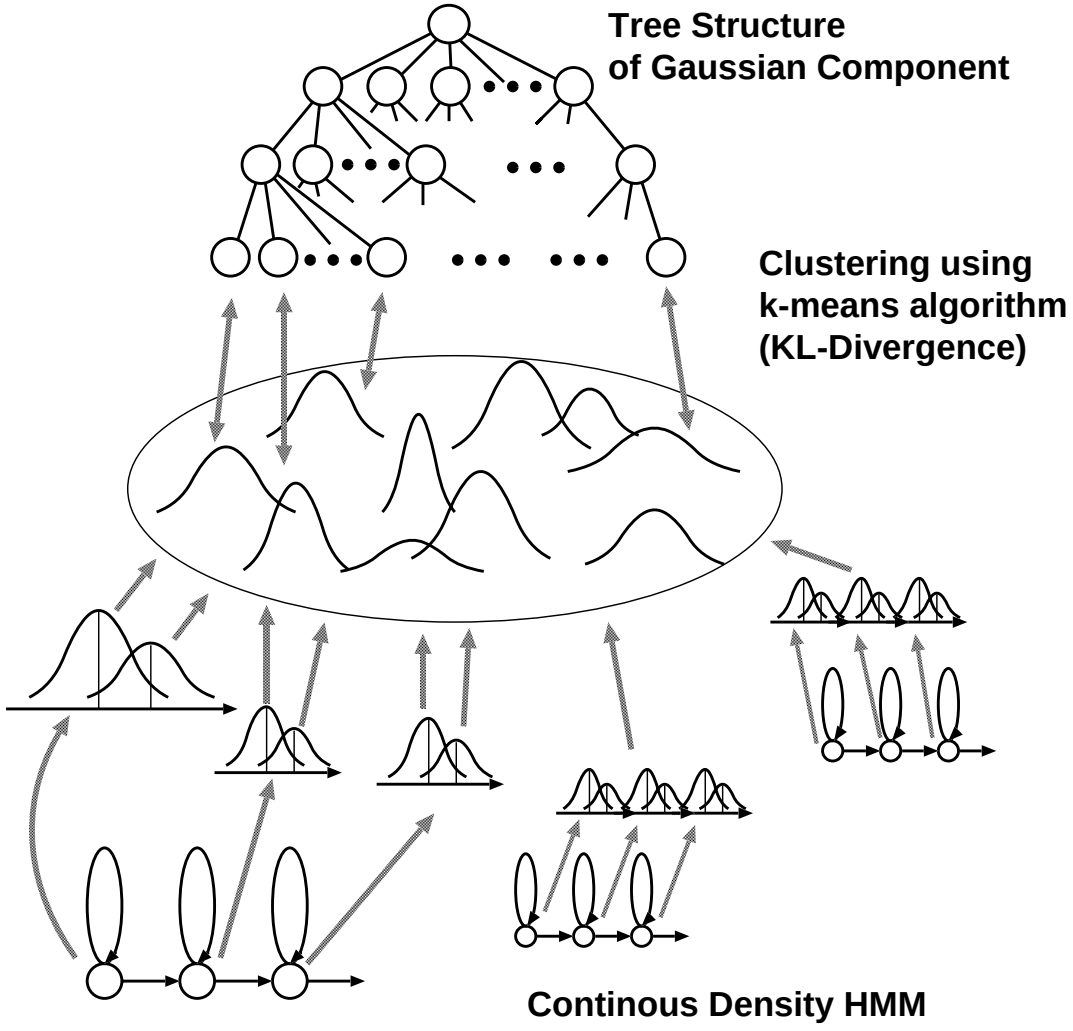

Figure 4.2: Tree structure for Gaussian pdfs in CDHMMs. For simplicity, the case when the dimension is one (scalar) is shown.

$g_m(\cdot)$ and $g_n(\cdot)$, in order to build a tree. Here the distance is defined as the sum of the *Kullback-Leibler divergence* from $g_m(\cdot)$ to $g_n(\cdot)$ and that from $g_n(\cdot)$ to $g_m(\cdot)$ [71]. When diagonal covariance matrices are assumed, the distance $d(m,n)$ is evaluated as follows:

$$\begin{aligned} d(m,n) &= \int g_m(x) \log \frac{g_m(x)}{g_n(x)} dx + \int g_n(x) \log \frac{g_n(x)}{g_m(x)} dx, \\ &= \sum_i [\frac{\sigma_m^2(i) - \sigma_n^2(i) + (\mu_n(i) - \mu_m(i))^2}{\sigma_n^2(i)} \\ &\quad + \frac{\sigma_n^2(i) - \sigma_m^2(i) + (\mu_n(i) - \mu_m(i))^2}{\sigma_m^2(i)}], \end{aligned} \tag{4.1}$$

where $\mu_m(i)$ is the i-th element of the mean vector $\boldsymbol{\mu}_m$ and $\sigma_m^2(i)$ is the i-th diagonal element of the covariance matrix $\boldsymbol{\Sigma}_m$. Next, at each node k in a tree structure, the collection of Gaussian components belonging to node k, $\{g_m^{(k)}(\boldsymbol{X}) = \mathcal{N}(\boldsymbol{X}|\boldsymbol{\mu}_m^{(k)}, \boldsymbol{\Sigma}_m^{(k)}) : m = 1, \ldots, M_k\}$, is approximated by a single Gaussian pdf, where M_k is the number of Gaussian components at node k. This pdf is called a *node* pdf. When it is assumed that the number of data samples from each mixture components are equal, the parameters for the node pdf are calculated as follows:

$$\mu_k(i) = \frac{1}{M_k} \sum_{m=1}^{M_k} E(x_m^{(k)}(i)) = \frac{1}{M_k} \sum_{m=1}^{M_k} \mu_m^{(k)}(i), \tag{4.2}$$

$$\begin{aligned} \sigma_k^2(i) &= \frac{1}{M_k} \sum_{m=1}^{M_k} E((x_m^{(k)}(i) - \mu_k(i))^2) \\ &= \frac{1}{M_k} \left[\sum_{k=1}^{M_k} \sigma_m^{2(k)}(i) + \sum_{m=1}^{M_k} \mu_m^{(k)2}(i) - M_k \mu_k^2(i) \right], \end{aligned} \tag{4.3}$$

where $\mathbf{x}_m^{(k)}$ is a data vector from Gaussian pdf $g_m^{(k)}$.

The following clustering algorithm is used to construct a tree structure for the mixture components in G, where the distance is calculated using Eq. (4.1) and the node pdf is calculated using Eqs. (4.2) and (4.3).

1. The structure of the tree structure is designed; the number of layers and the number of branches from a node in each layer are determined. There is no

clear way to design the structure automatically since the optimal structure may be changed according to the size of models.

2. Set the root node to be node k and the set G to be set G_{now}. Calculate the node pdf for the root node using Eqs. (4.2) and (4.3).

3. If node k has no child nodes, stop clustering. Otherwise, give the initial pdf for each child node using the *minimax* method that is described as follows. Here $g^{(k)}(\cdot)$ is the node pdf for node k, P_k is the number of child nodes of node k, and $g^{(c_p)}(\cdot)$ is the node pdf for child node c_p, $p = 1, \ldots, P_k$.

   (a) Choose among the set G_{now} the mixture component $\hat{m}$ that has the largest distance to $g^{(k)}(\cdot)$ and set it as node pdf for child c_1, i.e., $g^{(c_1)}(\cdot) = g_{\hat{m}}(\cdot)$.

   (b) Choose mixture components for c_p successively from $p = 2$ to $p = P_k$ and set those to the node pdfs for child nodes as follows:

$$\hat{m} = \text{argmax}_m \min_{1 \le i \le p-1} d(m, c_i), \tag{4.4}$$

$$g^{(c_p)}(\cdot) = g_{\hat{m}}(\cdot). \tag{4.5}$$

   In Eq.(4.4), $\hat{m}$ is chosen from the rest of mixture components, which belong to parent node k and not yet assigned any child node.

   (c) The node pdf for each child node c_p and the node pdf for k is interpolated and resulting pdf are set to be the node pdf for c_p as follows:

$$\mu'_{c_p}(i) = (1-\alpha)\mu_k(i) + \alpha\mu_{c_p}(i) \tag{4.6}$$

$$\sigma^{2'}_{c_p}(i) = (1-\alpha)(\sigma^2_k(i) + \mu^2_k(i)) + \alpha(\sigma^2_{c_p}(i) + \mu^2_{c_p}(i)) - \mu'^2_{c_p}(i), \tag{4.7}$$

   where $0 \le \alpha \le 1$.

4. Repeat the following k-means procedure until the grand sum of distances converges.

(a) For each mixture component in G_{now}, calculate the distance from it to each child node pdf by using Eq. (4.1), and assign each mixture component to the nearest child node.

(b) Recalculate the child node pdf by using Eqs. (4.2) and (4.3).

(c) Using Eq. (4.1), calculate the sum of distances from each child node to each of its mixture components and then obtain the grand sum of distances by summing up the sum of distances over all the child nodes.

5. Set each child node to be node k and its corresponding subset of mixture components G_{now}. Go to Step 3.

It is mostly expected that the substantial number of training samples assigned to each mixture component during the training process is largely different from component to component. This is because the phonetic distribution of speech is usually far from uniform. This imbalance is not taken into consideration in this method; it is assumed that the amount of the training samples for each component is the same among all the mixture components. Therefore, this method is expected to be robust against the imbalance of the number of training samples. Additionally, it is also applied when the phonetic distribution of training data and and that of test data are largely different. It should be noted that when the number of training samples is the same for all mixture components this method gives the same result as the clustering method based on maximum-likelihood criterion(e.g., [26]).

4.3 Summarization of Gaussian distributions

In this study, the focus is on adaptation of the parameters of the mixture Gaussian components in continuous-density HMMs (CDHMMs). Let $g_m(\cdot)$ be a normal density function for mixture component m, $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$, where $\boldsymbol{\mu}_m$ is a mean vector, $\boldsymbol{\Sigma}_m$ is a covariance matrix, and x is a D-dimensional observation vector.

Let $G = \{g_m(\cdot) : m = 1, \ldots, M\}$ be the whole set of the mixture components in HMMs, where M is the total number of mixture components in all the states of all the speech units. It is assumed that all the parameters of the general CDHMMs have already been trained by using a sufficient amount of training data from many speakers. Such models are typically used for speaker independent speech recognition.

Let $\boldsymbol{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ denote a given set of adaptation data. We are interested in using $\boldsymbol{X}$ to obtain estimates of the parameter set, $\theta_m = (\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$. Furthermore, we would like to make use of the tree structure established in Section 4.2 to construct prior information in each tree node for SMAP adaptation. Since in the case of CDHMM, we don't know the *membership* of $\mathbf{x}_t$, i.e. we have no precise knowledge which Gaussian component could have generated the observed vector. We usually associate with $\mathbf{x}_t$ a membership function, γ_{mt}, which is the posterior probability of observing $\mathbf{x}_t$ in the Gaussian component $g_m(\cdot)$ given the parameter values θ_m. γ_{mt} is usually computed with the well-known *forward-backward* algorithm (e.g., [41, 18]).

One way to make the problem easier is to go through a summarization process. As the first step of the process, each sample vector $\mathbf{x}_t$ is transformed into a vector $\mathbf{y}_{mt}$ for each mixture component m as follows:

$$\mathbf{y}_{mt} = \boldsymbol{\Sigma}_m^{-1/2}(\mathbf{x}_t - \boldsymbol{\mu}_m), \qquad t = 1, \ldots, T, \; m = 1, \ldots, M. \tag{4.8}$$

When there is no mismatch between the training data and the adaptation data, the pdf for $\boldsymbol{Y}_m = \{\mathbf{y}_{m1}, \ldots, \mathbf{y}_{mT}\}$ is obviously the standard normal distribution $\mathcal{N}(\boldsymbol{Y}|\vec{0}, \mathbf{I})$, where $\vec{0}$ is a zero vector and $\mathbf{I}$ is an identity matrix. When there is a mismatch between them, however, the pdf for $\boldsymbol{Y}$ is different from $\mathcal{N}(\boldsymbol{Y}|\vec{0}, \mathbf{I})$ for the adaptation data.

Here the pdf for $\boldsymbol{Y}$ is assumed to be $\mathcal{N}(\boldsymbol{Y}|\boldsymbol{\nu}, \boldsymbol{\Psi})$, where $\boldsymbol{\nu} \neq \vec{0}$ and $\boldsymbol{\Psi} \neq \mathbf{I}$ represent the shift and rotation needed to compensate for the distortion. It is expected that this pdf for $\boldsymbol{Y}$ better represents the difference between the acoustic characteristics of the training data and those of the adaptation data. This

is called the *normalized pdf*. It is also assumed that the mismatch can be modeled by models simpler than those used for speech recognition. In other words, the number of the normalized pdfs required to model the acoustic difference is assumed smaller than M, the number of the mixture components of the HMMs. Let us consider the case when the whole set of mixture components, G, is divided into subsets $\{G_1, \ldots, G_P\}$, where P is the total number of subsets, which is less than M, the total number of mixture components. One *common* normalized pdf, $h^{(p)}(\cdot) = \mathcal{N}(\boldsymbol{Y}|\boldsymbol{\nu}^{(p)}, \boldsymbol{\Psi}^{(p)})$, is shared by all the mixture components in each subset G_p. In the following explanation, all the mixture components in subset G_p are renumbered as $g_1^{(p)}(\cdot), \ldots, g_m^{(p)}(\cdot), \ldots, g_{M^{(p)}}^{(p)}(\cdot)$, where $M^{(p)}$ is the number of the mixture components in subset G_p. Obviously, $\sum_{p=1}^{P} \sum_{m=1}^{M^{(p)}} 1 = M$. For the mixture component, $g_m^{(p)}(\cdot)$, the observed vector sequence $\boldsymbol{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ is normalized to $\boldsymbol{Y} = \{\mathbf{y}_{m1}^{(p)}, \ldots, \mathbf{y}_{mT}^{(p)}\}$.

The maximum likelihood (ML) estimates for the parameters of the normalized pdfs can be calculated using the *expectation-maximization* (EM) algorithm (e.g., [41]). If the transition probabilities and the weight coefficients are assumed to be fixed, the auxiliary function $Q(\cdot|\cdot)$ for the HMM parameters [41, 18] is given by

$$Q(\Theta|\bar{\Theta}) = \sum_{t=1}^{T} \sum_{p=1}^{P} \sum_{m=1}^{M^{(p)}} \gamma_{mt}^{(p)} \log g_m^{(p)}(\mathbf{x}_t|\boldsymbol{\mu}_m^{(p)}, \boldsymbol{\Sigma}_m^{(p)}), \tag{4.9}$$

where $\Theta = \{\theta_m^{(p)} = (\boldsymbol{\mu}_m^{(p)}, \boldsymbol{\Sigma}_m^{(p)}) : m = 1, \ldots, M^{(p)}, p = 1, \ldots, P\}$ is the new estimate of the HMM parameters and $\bar{\Theta} = \{\bar{\theta}_m^{(p)} = (\bar{\boldsymbol{\mu}}_m^{(p)}, \bar{\boldsymbol{\Sigma}}_m^{(p)}) : m = 1, \ldots, M^{(p)}, p = 1, \ldots, P\}$ is the current estimate of the HMM parameters. The parameter $\gamma_{mt}^{(p)}$ is the posterior probability of observing mixture component $g_m^{(p)}(\cdot)$ at time t. The relation between the original pdf and the normalized pdf is as follows:

$$\begin{aligned} g(\mathbf{x}_t|\boldsymbol{\mu}_m^{(p)}, \boldsymbol{\Sigma}_m^{(p)}) &= \frac{h(\mathbf{y}_{mt}^{(p)}|\boldsymbol{\nu}^{(p)}, \boldsymbol{\Psi}^{(p)})}{|\mathbf{J}_m^{(p)}|} \\ &= \frac{h(\mathbf{y}_{mt}^{(p)}|\boldsymbol{\nu}^{(p)}, \boldsymbol{\Psi}^{(p)})}{|(\bar{\boldsymbol{\Sigma}}_m^{(p)})^{1/2}|}, \end{aligned} \tag{4.10}$$

where $\mathbf{J}_m^{(p)} = (\bar{\boldsymbol{\Sigma}}_m^{(p)})^{1/2}$ is the Jacobian matrix for the normalization transformation

in Eq. (4.8). This relation can be used to modify the auxiliary function as follows:

$$Q(\Theta|\bar{\Theta}) = \sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{m=1}^{M^{(p)}} \gamma_{mt}^{(p)} \log \frac{h(\mathbf{y}_{mt}^{(p)}|\boldsymbol{\nu}^{(p)}, \boldsymbol{\Psi}^{(p)})}{|(\bar{\boldsymbol{\Sigma}}_m^{(p)})^{1/2}|}. \tag{4.11}$$

The ML estimates of the parameters, $(\boldsymbol{\nu}^{(p)}, \boldsymbol{\Psi}^{(p)}) = (\tilde{\boldsymbol{\nu}}^{(p)}, \tilde{\boldsymbol{\Psi}}^{(p)})$, are calculated, by differentiating this equation, as follows:

$$\tilde{\boldsymbol{\nu}}^{(p)} = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M^{(p)}} \gamma_{mt}^{(p)} \mathbf{y}_{mt}^{(p)}}{\sum_{t=1}^{T}\sum_{m=1}^{M^{(p)}} \gamma_{mt}^{(p)}}, \tag{4.12}$$

$$\tilde{\boldsymbol{\Psi}}^{(p)} = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M^{(p)}} \gamma_{mt}^{(p)} (\mathbf{y}_{mt}^{(p)} - \tilde{\boldsymbol{\nu}}^{(p)})(\mathbf{y}_{mt}^{(p)} - \tilde{\boldsymbol{\nu}}^{(p)})^T}{\sum_{t=1}^{T}\sum_{m=1}^{M^{(p)}} \gamma_{mt}^{(p)}}, \tag{4.13}$$

where $(\mathbf{y}_{mt}^{(p)} - \tilde{\boldsymbol{\nu}}^{(p)})^T$ is the transpose of $(\mathbf{y}_{mt}^{(p)} - \tilde{\boldsymbol{\nu}}^{(p)})$. These normalized pdf parameters are used to estimate the corresponding HMM parameters by the following transformations,

$$\tilde{\boldsymbol{\mu}}_m^{(p)} = \bar{\boldsymbol{\mu}}_m^{(p)} + (\bar{\boldsymbol{\Sigma}}_m^{(p)})^{1/2} \tilde{\boldsymbol{\nu}}^{(p)}, \tag{4.14}$$

$$\tilde{\boldsymbol{\Sigma}}_m^{(p)} = \bar{\boldsymbol{\Sigma}}_m^{(p)1/2} \tilde{\boldsymbol{\Psi}}^{(p)} (\bar{\boldsymbol{\Sigma}}_m^{(p)1/2})^T, \tag{4.15}$$

where $\tilde{\boldsymbol{\mu}}_m^{(p)}$ and $\tilde{\boldsymbol{\Sigma}}_m^{(p)}$ are the updated ML estimates of the mean and covariance of the m-th component, respectively.

Let us compare this normalization technique with the *stochastic matching* (SM) algorithm [44] for compensating mismatch during speech recognition. As can be seen from Eqs. (4.14) and (4.15), $(\bar{\boldsymbol{\Sigma}}_m^{(p)})^{1/2} \tilde{\boldsymbol{\nu}}^{(p)}$ corresponds to the bias in SM, where $\bar{\boldsymbol{\Sigma}}_m^{(p)}$ is the covariance for mixture component $g_m^{(p)}(\cdot)$, and $\tilde{\boldsymbol{\Psi}}^{(p)}$ corresponds to the scaling factor in SM when the diagonal covariance is used for $\tilde{\boldsymbol{\Psi}}^{(p)}$. In the proposed method, the bias for each mixture component changes according to the variance. When the variance is large, the bias is also large. Experimental results to compare the two compensation and normalization approaches will be given later.

4.4 Hierarchical prior

The tree structure representation of the set of Gaussian mixture components have been developed in Section 4.2. How to construct a node normalized pdf to

approximate a collection of heterogeneous Gaussian pdfs in the node cluster, and how to obtain maximum likelihood estimate of the mean and the covariance of the normalized pdf have been also shown in Section 4.3. Next, the framework of hierarchical priors is established.

As discussed before we are interested in estimating the parameter set, $\Theta = \{\theta_m = (\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) : m = 1, \ldots, M\}$, based on a small set of adaptation data, $\boldsymbol{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$. Let the normalized Gaussian pdf, $h_k^{(p)}(\cdot)$, at the p-th node of the k-th layer of the tree, be assigned as the parent node of the m-th mixture component $g_m(\cdot)$ by clustering the set, $G = \{g_m(\cdot)\}$. Then, the parameters θ_m can be estimated with the transformations in Eqs. (4.14) and (4.15) using the ML estimates shown in Eqs. (4.12) and (4.13).

Since the normalized pdf is only an approximation to $g_m(\cdot)$ under mismatch conditions, we expect a better estimate of θ_m, which is directly connected to the K-th layer (or *leaf layer*), to be obtained with an estimate, $\tilde{\lambda}_{K-1}^{(p)} = (\tilde{\boldsymbol{\nu}}_{K-1}^{(p)}, \tilde{\boldsymbol{\Psi}}_{K-1}^{(p)})$, at the immediate parent node of $g_m(\cdot)$. To establish a hierarchical prior framework for MAP estimation, we also extend the argument and assume that any normalized pdf, $h_k^{(p)}(\cdot)$, at the k-th layer of the tree is better characterized by some prior information from the immediate parent nodes at the $(k-1)$-st layer. One easy way to accomplish this is to assume the prior density for estimating λ_k is based on some knowledge about λ_{k-1}. An even stronger constraint is to assume a prior density of the form $p(\lambda_k|\hat{\lambda}_{k-1})$, i.e. the hyperparameters for the prior densities are simply derived from some estimate $\hat{\lambda}_{k-1}$ of λ_{k-1}.

Now for each $k = 1, \ldots, K$, with a given estimate $\hat{\lambda}_{k-1}$. the MAP estimate $\hat{\lambda}_k$ is evaluated as follows:

$$\begin{aligned} \hat{\lambda}_k &= \text{argmax}_{\lambda_k} p(\lambda_k|\hat{\lambda}_{k-1}, \boldsymbol{Y}) \\ &= \text{argmax}_{\lambda_k} \frac{p(\boldsymbol{Y}|\lambda_k, \hat{\lambda}_{k-1}) p(\lambda_k|\hat{\lambda}_{k-1})}{p(\boldsymbol{Y})}. \end{aligned} \tag{4.16}$$

Note $\hat{\lambda}_0 = \lambda_0 = (\vec{0}, \mathbf{I})$ is the known parameter assumed at the root of the tree. By further assuming that $p(\boldsymbol{Y}|\lambda_k, \hat{\lambda}_{k-1})$ does not depend on $\hat{\lambda}_{k-1}$ and since $p(\boldsymbol{Y})$

is not a function of λ_k, we have

$$\hat{\lambda}_k = \text{argmax}_{\lambda_k} p(\boldsymbol{Y}|\lambda_k) \cdot p(\lambda_k|\hat{\lambda}_{k-1}), \quad k = 1, \ldots, K. \tag{4.17}$$

It has been already assumed in Section 4.3 that $p(\boldsymbol{Y}|\lambda_k)$ is a normal density. To make MAP estimation in Eq. (4.17) more tractable, it is assumed that the *conjugate* prior density for the random vector λ_k, $p(\lambda_k|\hat{\lambda}_{k-1})$, to be a normal-Wishart density of the form (e.g., [10, 18]),

$$g(\boldsymbol{\nu}_k, \boldsymbol{\Psi}_k|\hat{\boldsymbol{\nu}}_{k-1}, \hat{\boldsymbol{\Psi}}_{k-1}, \xi_k, \tau_k) \propto$$
$$|\boldsymbol{\Psi}_k|^{-(\xi_k - D)/2} \exp\left[-\frac{\tau_k}{2}(\boldsymbol{\nu}_k - \hat{\boldsymbol{\nu}}_{k-1})^T \boldsymbol{\Psi}_k^{-1}(\boldsymbol{\nu}_k - \hat{\boldsymbol{\nu}}_{k-1})\right] \exp\left[-\frac{1}{2}\text{tr}(\hat{\boldsymbol{\Psi}}_{k-1}\boldsymbol{\Psi}_k^{-1})\right], \tag{4.18}$$

for $k = 1, \ldots, K$, with $\tau_k > 0$ and $\xi_k > D - 1$ being the control parameters specified by external constraints with D being the dimension of the observation vector. We now arrive at a familiar MAP solution [18]. By performing this MAP estimation sequentially at each layer k, we obtain the MAP estimate $\hat{\lambda}_K^{(p)}$, at the p-th cluster of the leaf layer K. Then the MAP estimate, $\hat{\theta}_m$, for each Gaussian mixture component can be solved. This novel estimation formulation is described in the following.

4.5 SMAP adaptation using hierarchical priors

Consider for the set G of all the Gaussian mixture component, we have available a *tree structure* like the one shown in Figure 4.3, where K is the total number of layers or the depth of the tree. Each node in the K-th layer (leaf node) corresponds to one Gaussian mixture component in the set of CDHMMs. The root node (the first layer) corresponds the whole set G of the mixture components. Each intermediate node corresponds to a subset of G, and each of its subordinate leaf nodes corresponds to an element of a subset. At each node in the tree, a normalized pdf, which is shared among the mixture components in the corresponding subset of G, is assigned. For each node N, at the p-th cluster of the k-th layer, Eqs. (4.12)

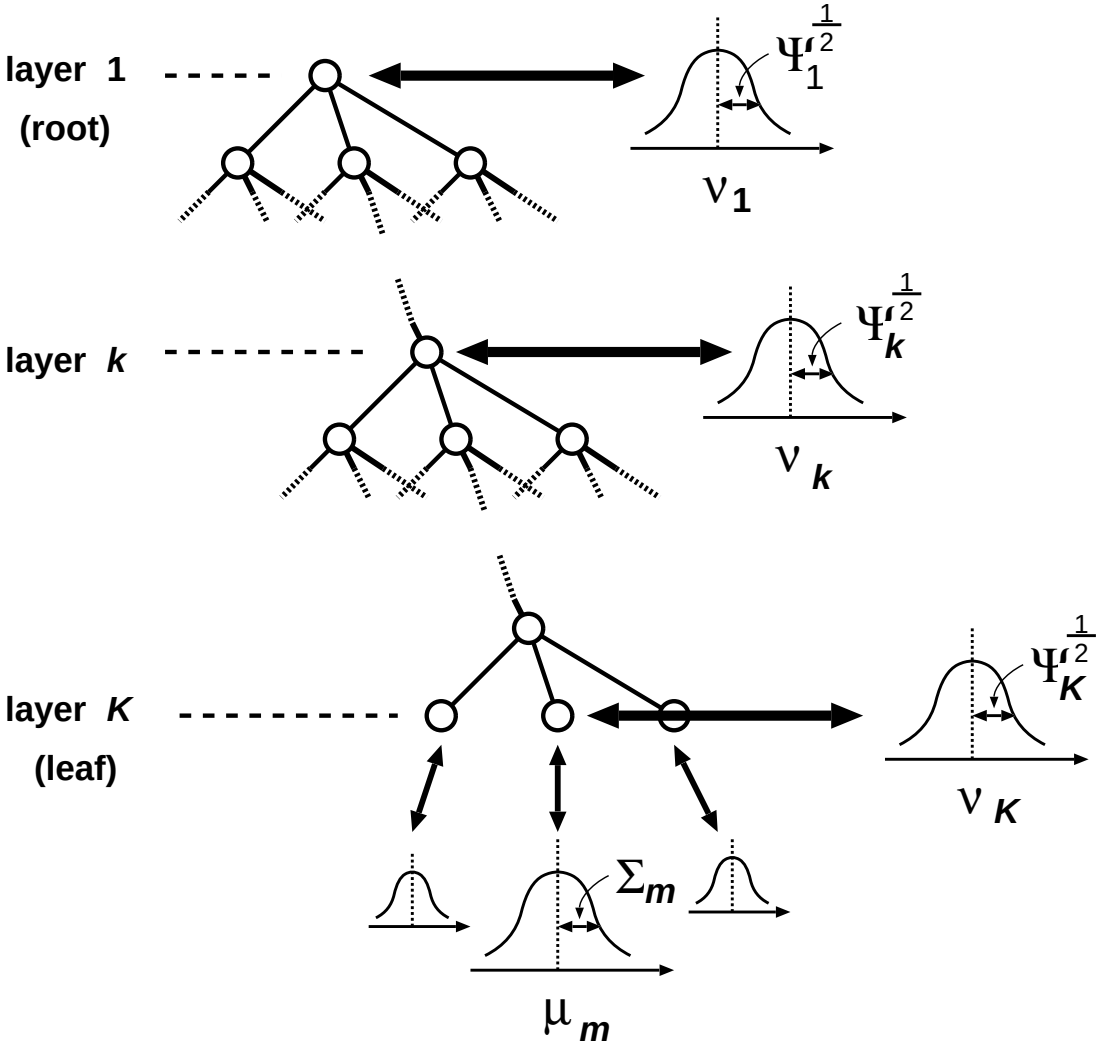

Figure 4.3: SMAP adaptation for Gaussian pdfs in CDHMMs. For simplicity, the case when the dimension is one (scalar) is shown.

and (4.13) are used to calculate the ML estimates of the pdf parameters, $\tilde{\boldsymbol{\nu}}_k^{(p)}$ and $\tilde{\boldsymbol{\Psi}}_k^{(p)}$ (in this case, G_p is the subset related to node N). In the tree structure, one node sequence from the root to a leaf corresponds to all the predecessor nodes that must be traversed to reach a particular mixture component. From now on the focus is on estimation of the parameter set, $\theta_m = (\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$, for a particular m-th mixture component in G. The suffix identifying the mixture component is therefore omitted except when doing so causes confusion. The procedure described below is general and can be used to estimate the parameter sets of all the other mixture components in CDHMMs.

Let the node sequence from the root to the leaf corresponding to the m-th mixture component be $\{N_1, \ldots, N_k, \ldots, N_K\}$, where N_1 is the root node and N_K is the leaf node directly attached to mixture component m. We denote $\lambda_k = (\boldsymbol{\nu}_k, \boldsymbol{\Psi}_k)$ as the Gaussian pdf parameters for node N_k. Now consider the problem of estimating, the parameter set λ_K that maximizes the posterior probabilities, after observing a sequence of feature vectors $\boldsymbol{Y}$. It should be noted that once λ_K is obtained, the parameter set Θ can be obtained immediately by using Eqs. (4.14) and (4.15). In the proposed approach, a set of priors, $\{p(\lambda_k|\hat{\lambda}_{k-1})\}$, are used as *hierarchical priors* for estimating λ_K, where λ_0 is fixed to be $\hat{\lambda}_0 = N(\vec{0}, \mathbf{I})$. Based on the discussion in Section 4.4 the pdf for node N_k, which has the parameter set, λ_k, is assumed to have a hyperparameter, $\hat{\lambda}_{k-1}$, directly extended from its immediate parent node, N_{k-1}.

Since $p(\lambda_k|\boldsymbol{Y}) = \int p(\lambda_k|\lambda_{k-1}, \boldsymbol{Y}) p(\lambda_{k-1}|\boldsymbol{Y})\, d\lambda_{k-1}$, for $k = 1, \ldots, K$, the posterior distribution for λ_K is then expressed as follows:

$$
\begin{aligned}
p(\lambda_K|\boldsymbol{Y}) = & \\
& \int \ldots \int p(\lambda_K|\lambda_{K-1}, \boldsymbol{Y}) \ldots p(\lambda_k|\lambda_{k-1}, \boldsymbol{Y}) \ldots \\
& \ldots p(\lambda_1|\lambda_0, \boldsymbol{Y}) p(\lambda_0|\boldsymbol{Y}) d\lambda_0 d\lambda_1 \ldots d\lambda_{K-1}, \qquad (4.19)
\end{aligned}
$$

with

$$
\int p(\lambda_0|\boldsymbol{Y}) d\lambda_0 = 1, \qquad (4.20)
$$

because λ_0 is assumed to be known.

Because Eq. (4.19) is difficult to maximize directly, a *key* step here is to assume

$$\int p(\lambda_1|\lambda_0,\boldsymbol{Y})p(\lambda_0|\boldsymbol{Y})d\lambda_0 \simeq p(\lambda_1|\hat{\lambda}_0,\boldsymbol{Y}), \tag{4.21}$$

$$\int p(\lambda_{k+1}|\lambda_k,\boldsymbol{Y})p(\lambda_k|\hat{\lambda}_{k-1},\boldsymbol{Y})d\lambda_k \simeq p(\lambda_{k+1}|\hat{\lambda}_k,\boldsymbol{Y}), \quad k=1,\ldots,K-1, \tag{4.22}$$

where $\hat{\lambda}_k$ is the MAP estimate obtained in Eq. (4.17). The posterior distribution for λ_K is thus approximated as follows:

$$p(\lambda_K|\boldsymbol{Y}) \simeq \prod_{k=0}^{K-1} p(\lambda_{k+1}|\hat{\lambda}_k,\boldsymbol{Y}). \tag{4.23}$$

Under these assumptions the MAP estimates in Eq. (4.17) for each node N_k are calculated. First, the ML estimate for each node k is calculated. the auxiliary function $Q(\cdot|\cdot)$ for solving $\hat{\lambda}_k$ is written as follows:

$$Q(\lambda_k|\bar{\lambda}_k) = \sum_{t=1}^{T}\sum_{m\in G_k} \gamma_{mt} \log \frac{h(\mathbf{y}_{mt}|\lambda_k)}{|(\bar{\boldsymbol{\Sigma}}_m)^{1/2}|} + C, \tag{4.24}$$

where C is the part that is independent of λ_k, $\bar{\boldsymbol{\Sigma}}_m$ is known and not a function of λ_k and G_k is a subset of G corresponding to node N_k. We also define $\Gamma_k = \sum_{t=1}^{T}\sum_{m\in G_k}\gamma_{mt}$, with the membership function γ_{mt} being evaluation at the given value of $\bar{\lambda}_k$. Then by excluding the constant term C, the MAP estimates are calculated by maximizing the following auxiliary function with respect to λ_k:

$$\begin{aligned} R(\lambda_k|\bar{\lambda}_k) &= Q(\lambda_k|\bar{\lambda}_k) + \log g(\lambda_k), \\ &= \sum_{t=1}^{T}\sum_{m\in G_k}\Big[\gamma_{mt}(-\frac{1}{2}\log|\boldsymbol{\Psi}_k| - \frac{1}{2}(\mathbf{y}_{mt}-\boldsymbol{\nu}_k)\boldsymbol{\Psi}_k^{-1}(\mathbf{y}_{mt}-\boldsymbol{\nu}_k)^T\Big] + \\ &\quad [-\frac{\xi_k - D}{2}\log|\boldsymbol{\Psi}_k| - \frac{\tau_k}{2}(\boldsymbol{\nu}_k-\hat{\boldsymbol{\nu}}_{k-1})\boldsymbol{\Psi}_k^{-1}(\boldsymbol{\nu}_k-\hat{\boldsymbol{\nu}}_{k-1})^T \\ &\quad -\frac{1}{2}\mathrm{tr}(\hat{\boldsymbol{\Psi}}_{k-1}\boldsymbol{\Psi}_k^{-1}))], \end{aligned} \tag{4.25}$$

where $Q(\lambda_k|\bar{\lambda}_k)$ and $g(\lambda_k)$ are defined in Eq. (4.24) and the relation in Eq. (4.18). By differentiating Eq. (4.25) and setting its result to zero, the MAP estimates,

$(\boldsymbol{\nu}_k, \boldsymbol{\Psi}_k) = (\hat{\boldsymbol{\nu}}_k, \hat{\boldsymbol{\Psi}}_k)$, are calculated as follows: for $k = 1, \ldots, K$,

$$\hat{\boldsymbol{\nu}}_k = \frac{\Gamma_k \tilde{\boldsymbol{\nu}}_k + \tau_k \hat{\boldsymbol{\nu}}_{k-1}}{\Gamma_k + \tau_k}, \tag{4.26}$$

$$\hat{\boldsymbol{\Psi}}_k = \frac{\hat{\boldsymbol{\Psi}}_{k-1} + \Gamma_k \tilde{\boldsymbol{\Psi}}_k + \frac{\tau_k \Gamma_k}{\tau_k + \Gamma_k}(\tilde{\boldsymbol{\nu}}_k - \hat{\boldsymbol{\nu}}_{k-1})(\tilde{\boldsymbol{\nu}}_k - \hat{\boldsymbol{\nu}}_{k-1})^T}{(\xi_k - D) + \Gamma_k}, \tag{4.27}$$

with $\hat{\boldsymbol{\nu}}_0 = \vec{0}$ and $\hat{\boldsymbol{\Psi}}_0 = \mathbf{I}$ at the root node and $\tilde{\boldsymbol{\nu}}_k$ and $\tilde{\boldsymbol{\Psi}}_k$ are ML estimates shown in Eqs. (4.12) and (4.13). The mean $\hat{\boldsymbol{\nu}}_K$ and the variance $\hat{\boldsymbol{\Psi}}_K$ for the leaf node N_K are obtained by applying Eqs. (4.26) and (4.27) successively from the root node to the leaf node. To obtain the approximate MAP estimate, these $\hat{\boldsymbol{\nu}}_K$ and $\hat{\boldsymbol{\Psi}}_K$ values in Eqs. (4.26) and (4.27) are first assigned as $\hat{\boldsymbol{\nu}}^{(p)}$ and $\hat{\boldsymbol{\Psi}}^{(p)}$ according to their corresponding cluster membership and then used to replace the $\tilde{\boldsymbol{\nu}}^{(p)}$ and $\tilde{\boldsymbol{\Psi}}^{(p)}$ values shown in Eqs. (4.14) and (4.15), i.e.,

$$\hat{\boldsymbol{\mu}}_m^{(p)} = \bar{\boldsymbol{\mu}}_m^{(p)} + (\bar{\boldsymbol{\Sigma}}_m^{(p)})^{1/2} \hat{\boldsymbol{\nu}}^{(p)}, \tag{4.28}$$

$$\hat{\boldsymbol{\Sigma}}_m^{(p)} = \bar{\boldsymbol{\Sigma}}_m^{(p)1/2} \hat{\boldsymbol{\Psi}}^{(p)} (\bar{\boldsymbol{\Sigma}}_m^{(p)1/2})^T, \tag{4.29}$$

where $\hat{\boldsymbol{\mu}}_m^{(p)}$ and $\hat{\boldsymbol{\Sigma}}_m^{(p)}$ are the updated MAP estimates of the mean and covariance of the m-th component, respectively.

It should be noted that the prior parameters, τ_k and ξ_k, for all the nodes in the tree need to be specified. Since the SMAP framework provides no specific ways to calculate these parameters, optimal values should be determined empirically. One simple way to do this is to use the same τ and ξ for all the nodes and optimize their values by using preliminary recognition experiments (see Section 4.5.B).

Equation (4.26) can be rewritten for the leaf node as follows:

$$\hat{\boldsymbol{\nu}}_K = \sum_{k=1}^{K} w_k \tilde{\boldsymbol{\nu}}_k, \tag{4.30}$$

where the weighting factor w_k is

$$w_k = \frac{\Gamma_k}{\Gamma_k + \tau_k} \prod_{i=k+1}^{K} \frac{\tau_i}{\Gamma_i + \tau_i}. \tag{4.31}$$

The mean vector estimated using the SMAP method can be considered as a weighted sum of the ML estimates at the different layers of the tree. Two important characteristics of the weight, w_k, are highlighted in the following:

1. The weight w_k at node N_k becomes larger as the amount of data at that node, Γ_k, becomes larger.

2. The weight w_k at node N_k decreases as k becomes smaller.

These properties are desirable for adaptation. When the amount of data is small, the ML-estimated parameters in the upper layers, which represent global transformation, are mainly responsible for the resulting pdf. And when the amount of data is large, the parameters in the lower layers, which represent localized transformation, predominate.

It should be noted that Furui [17] has already developed an unsupervised adaptation method that utilizes a hierarchical structure for vector quantization. It is also important to note that similar hierarchical structures have already been used in the acoustic modeling ([3, 31, 74, 54]). Although this SMAP approach is not the first using tree-based adaptation (e.g., [38]), the method described here is thought to be theoretically well-defined in terms of taking advantage of both the Bayesian framework and the tree construction principle. It demonstrates that this framework and this principle work well together as will be clear in the next section.

4.6 Experiments

The proposed method was experimented with the 991-word DARPA resource management (RM) task [40]. New adaptation and testing data from five non-native male speakers (labeled as A,B,C,D, and E) were recorded simultaneously under two acoustic conditions: (1) a close-talking microphone (MIC); and (2) a telephone handset over a dial-up line (TEL). The data for adaptation consisted of 300 utterances from each speaker in each of the two channels. For testing, 75 utterances from each of the two channels were collected from each speaker. In the following experiments all the 75 utterances were always used in a particular condition for testing.

The speech signal was first down-sampled from 16 kHz to 8 kHz and the analysis frames were 30-ms wide with a 20-ms overlap. For each frame a 38-dimensional feature vector [31] was extracted by using a tenth-order LPC analysis. The feature vector for each frame consists of a 12-dimensional cepstral vector, a 12-dimensional delta-cepstrum vector, a 12-dimensional delta-delta-cepstrum vector, a delta log energy feature and a delta-delta log energy feature. For recognition, a set of 1769 context-dependent phone HMMs [31] was used. All units except the one for background silence had three states, each with a maximum of 16 mixture components. This gives a total of about 5,000 states and about 80,000 distinct Gaussian densities. Forty seven context-independent phones were used to create all the context-dependent units. For all the experiments, the RM word pair grammar was used, which gives a perplexity of about 60. A diagonal covariance was assumed for each mixture Gaussian component.

Speaker-independent HMMs were trained using the NIST/RM SI-109 training set consisting of 3990 utterances from 109 native American talkers (31 females and 78 males), each providing 30 or 40 utterances. These models were then adapted, using a MAP adaptation algorithm [18], with the data from the 78 male talkers in order to create speaker-independent male models. These speaker-independent male models served as initial seed models for further adaptation. The tree structures used in the experiments were constructed using the parameters of the speaker-independent male models. The background noise model was excluded in tree construction.

4.6.1 Summarization

The effectiveness of the summarization technique using the normalized pdfs is examined first. In this experiment, one single normalized pdf was applied to all the mixture components and its covariance was assumed as the identity matrix, i.e. no mismatch in scaling was considered. This normalization technique was compared with two other methods: stochastic matching (SM) [44] and a method

Table 4.1: Recognition rates (%) for MIC data when one utterance was used for normalization. SI is the result of a speaker-independent recognition experiment.

SI	Normalized pdf	SM	Tied-Shift
75.2	81.2	81.7	81.7

that uses a shift in the feature vector space (Tied-Shift) [50, 51, 36, 52, 54]. In SM the i-th element of the mean vector for the m-th component is adapted as follows:

$$\tilde{\mu}_m(i) = \mu_m(i) + \frac{\sum_{t=1}^{T}\sum_{l=1}^{M}\gamma_{lt}\frac{x_t(i)-\mu_l(i)}{\sigma_l^2(i)}}{\sum_{t=1}^{T}\sum_{l=1}^{M}\frac{\gamma_{lt}}{\sigma_l^2(i)}}. \tag{4.32}$$

And in Tied-Shift the mean vector for the m-th component is adapted as follows:

$$\tilde{\mu}_m(i) = \mu_m(i) + \frac{\sum_{t=1}^{T}\sum_{l=1}^{M}\gamma_{lt}(x_t(i)-\mu_l(i))}{\sum_{t=1}^{T}\sum_{l=1}^{M}\gamma_{lt}}. \tag{4.33}$$

In contrast, using a single normalized pdf, the mean vector for the m-th component is adapted as follows:

$$\tilde{\mu}_m(i) = \mu_m(i) + \sigma_m(i)\cdot\frac{\sum_{t=1}^{T}\sum_{l=1}^{M}\gamma_{lt}\frac{x_t(i)-\mu_l(i)}{\sigma_l(i)}}{\sum_{t=1}^{T}\sum_{l=1}^{M}\gamma_{lt}}. \tag{4.34}$$

The recognition experiments for the five speakers were carried out using the data collected through the microphone (MIC). Supervised adaptation in which only one utterance was used for adaptation was carried out. The results of the experiments, averaged over five speakers, were listed in Table 4.1. It shows that the simplified normalization technique gave similar improvement over the SI recognition accuracy compared with the other two methods. This normalization procedure will be used throughout the rest of this chapter.

4.6.2 Tree clustering

It is difficult to choose the optimal tree structure for adaptation. To determine a good experimental condition, several structures were tested in a set of supervised

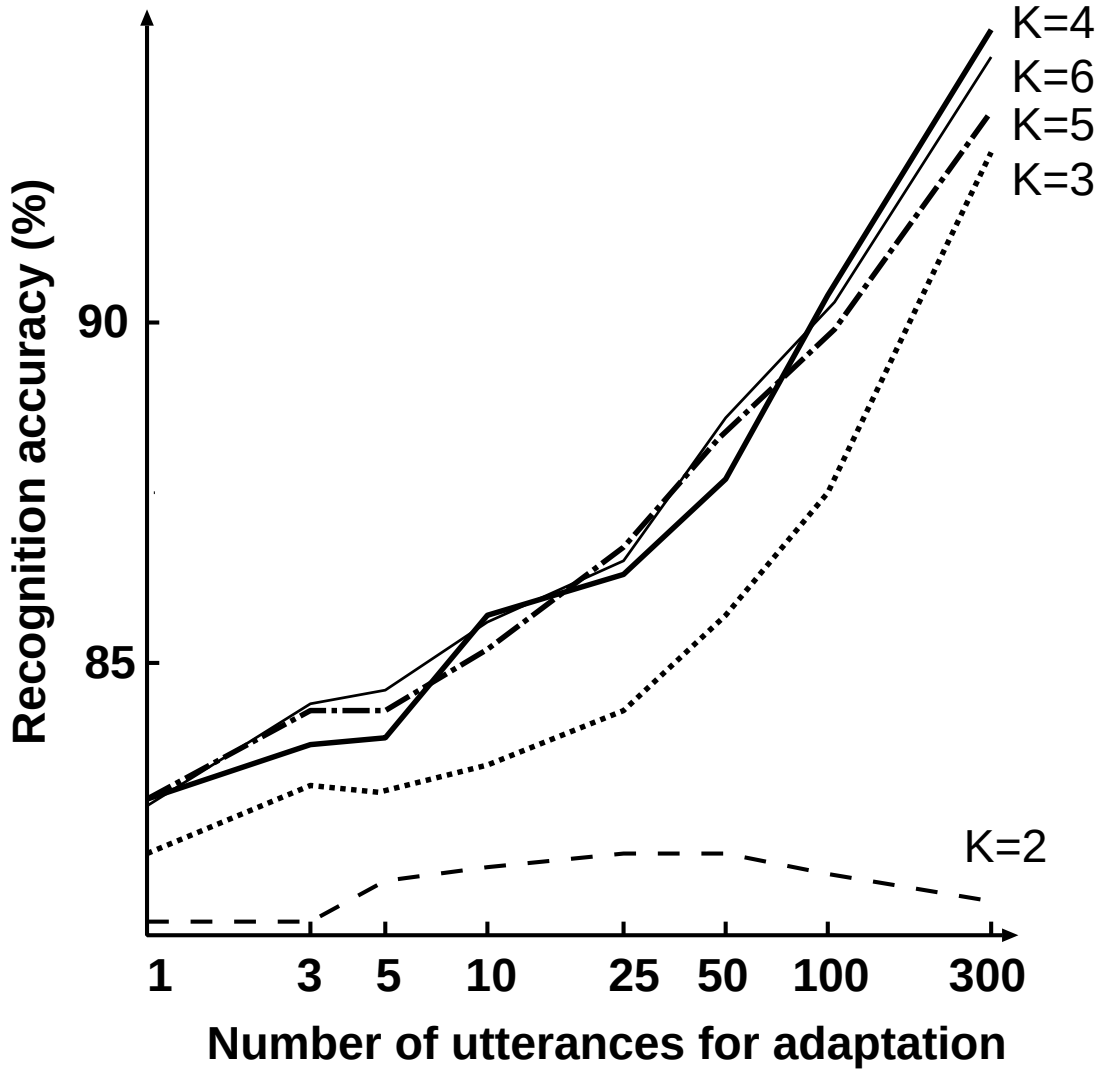

Figure 4.4: Recognition results obtained using different tree structures.

adaptation experiments. In each experiment the number of branches from each node was fixed to ten, and the depth of the tree was changed from one layer ($K = 2$ with only one common normalized pdf) to five layers ($K = 6$ with 11,111 normalized pdfs). The weight α used in tree clustering in Eqs. (4.6) and (4.7) is fixed at a value of 0.1 for all the experiments. Only the mean vectors were modified and the variances remained unchanged. Recognition accuracies obtained in the adaptation recognition experiments (averaged over five speakers) are shown in Figure 4.4 as a function of the amount of adaptation data. The result obtained with only one normalized pdf ($K = 2$) was obviously the worst. The result obtained with the two-layer tree ($K = 3$) was much better than that obtained with one normalized pdf ($K = 2$). The result obtained with the three-layer tree ($K = 4$) was better than that obtained with the two-layer tree ($K = 3$), and the results were similar for $K = 4,5$, and 6. The three-layer ($K = 4$) tree was therefore used in the following experiments. In all experiments the control parameters τ_k and ξ_k were shared by all the nodes in the tree. Using $\tau_k = 2$ and $\xi_k = D + 1$ gave the best results for the three-layer ($K = 4$ with 111 clusters) tree in some preliminary experiments. These control parameters were therefore used in all the following experiments.

Next, how the weights in Eq. (4.31) changed according to the amount of data was investigated. Those changes for each layer in the four-layer ($K = 5$) tree were examined. Figure 4.5 shows the distribution of the weight among the layers in the tree for four different numbers of utterances: 1, 5, 25 and 100. When the amount of data was small, the ML estimates of the upper layers (near the root node) were the ones mainly used. As the amount of data became larger, those of the lower layers (near the leaf nodes) predominated.

Next how the acoustic features of the speech units were distributed in the tree structure was examined . All the units were classified into nine phonemic classes [15] according to the feature of the central phone label in a triphone unit and counted the number of mixture components in each node. Only the mixture

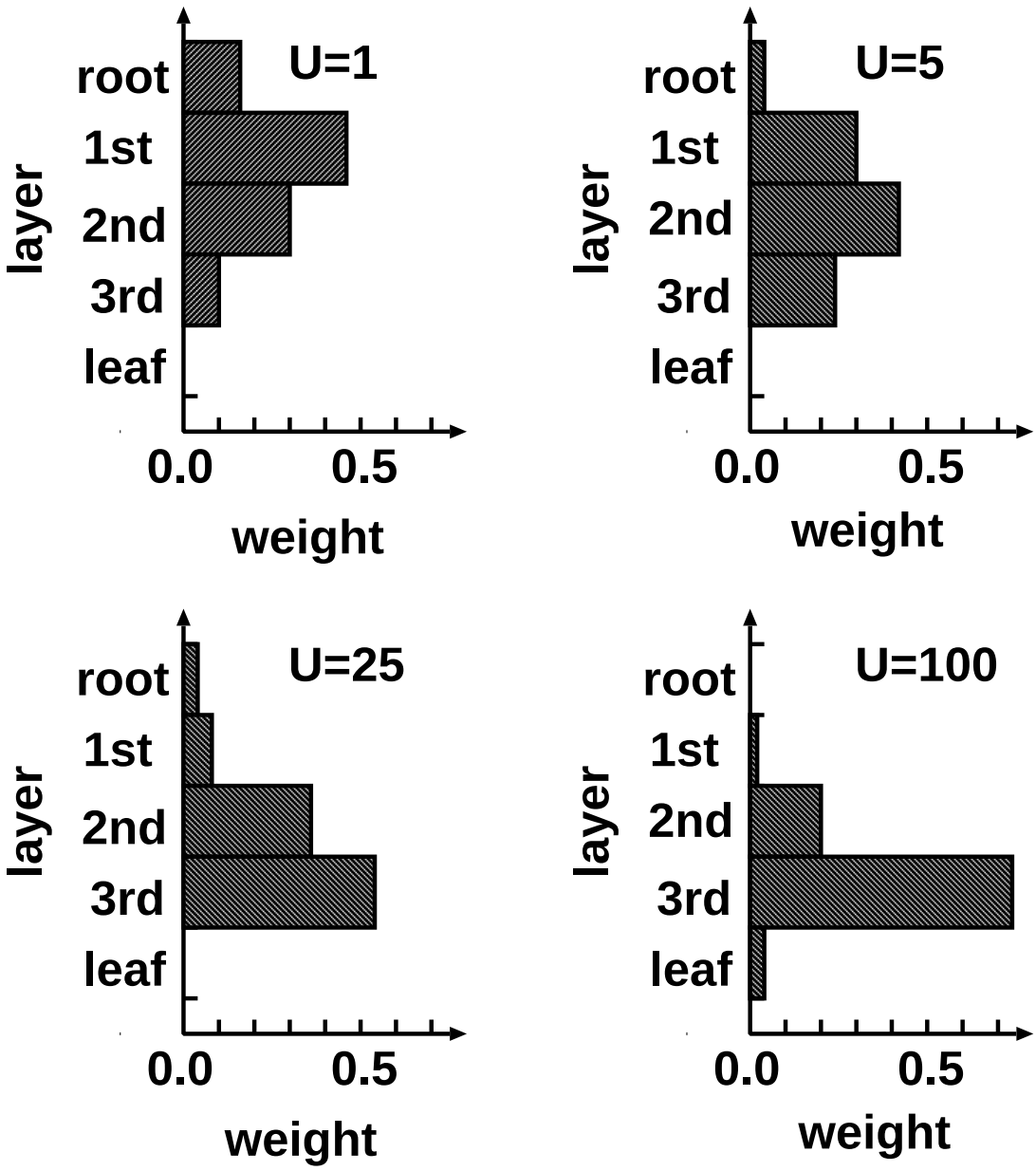

Figure 4.5: The weight for each layer in the tree. In each layer, the weights for all the leaf nodes are averaged.

Table 4.2: Phone distribution (%) in the first layer ($K = 2$) of the tree.

Node No.	1	2	3	4	5	6	7	8	9	10
Front vowels	72.4		0.1	20.9	0.9	3.7	0.4	0.6	0.6	0.5
Central vowels	49.7		0.6	6.5	4.1	23.2	0.7	6.8	6.5	1.8
Back vowels	92.2	0.1		1.3		2.5		1.5	2.2	0.3
Diphthongs	98.6			0.7				0.2	0.2	0.2
Fricatives	4.9	2.9	2.7	1.2		0.8	82.6	0.5	1.2	3.3
Stops	2.8	63.2	10.5	4.0	0.5	1.1	12.6	0.9	0.5	3.8
Nasals	51.6	0.1	0.2	2.4	0.1	3.2	0.2	0.4	0.5	41.4
Affricates		7.0		1.0			92.0			
Glides, Semivowels	59.3			3.7	1.8	2.0		21.6	9.2	2.4

Table 4.3: Phone distribution (%) in the second layer ($K = 3$) of the tree.

Node No.	1	2	3	4	5	6	7	8	9	10
Front vowels	68.6	0.5	7.6	0.1	0.0	0.4		21.7	0.8	0.3
Central vowels	10.4	5.9	44.6	0.1	0.8	5.6	0.1	19.2	10.5	2.9
Back vowels	14.5	0.5	66.8	0.1	7.6	4.2		0.9	0.2	5.2
Diphthongs	0.5		93.7		0.2	0.2	1.2	4.0	0.2	
Nasals	29.9	6.7	0.8	1.6				1.1		59.8
Glides, Semivowels	11.5	0.6	18.2	0.2	41.9	0.6	11.8	0.4	10.4	4.4

components in the middle state of each unit were selected because the other two may be influenced by their nearby phones. The distribution in the nodes in the second layer (the child nodes of the root-node) is shown by the data listed in Table 4.2. The numbers in each row are the percentage of mixture components that were classified into the various nodes. For the affricates, for example, 92% of the corresponding mixture components were clustered into node No. 7. It can be seen that fricatives, affricates, and stops were distinguished from the other features, whereas all the vowels, nasals, glides, and semi-vowels were mostly in the first node (Node No. 1). Therefore, the child nodes of Node No. 1 were examined next. The results were listed in Table 4.3. This time, front vowels were distinguished from the other vowels (central vowels, back vowels, and diphthongs), and glides, semi-vowels, and nasals were separated from vowels. These results show that the clustering using only the distance in the acoustic space results in a mixture component grouping that is phonologically meaningful.

4.6.3 Supervised adaptation experiments

To verify the effectiveness of the SMAP method, it was compared with conventional MAP estimation (MAP) [18] and with simple bias estimation using a tree structure without MAP estimation (TREE) [53]. In the experiment labeled as MAP estimation, no structure in the acoustic space was assumed and each parameter of HMMs was estimated separately. In the TREE experiment, one node in the tree was selected for each mixture component by using a threshold data amount, and the ML estimates for the parameters at that node were used to modify the parameter of the corresponding mixture component (Figure 4.6).

The recognition results averaged over all five non-native speakers, for the two acoustic conditions, MIC and TEL, are shown in Figures 4.7 and 4.8, respectively. As can be seen, the baseline performance of SI models is much lower for TEL than for MIC. This is because the combined microphone and channel distortion was larger in the TEL data than the situation in the MIC data in which the

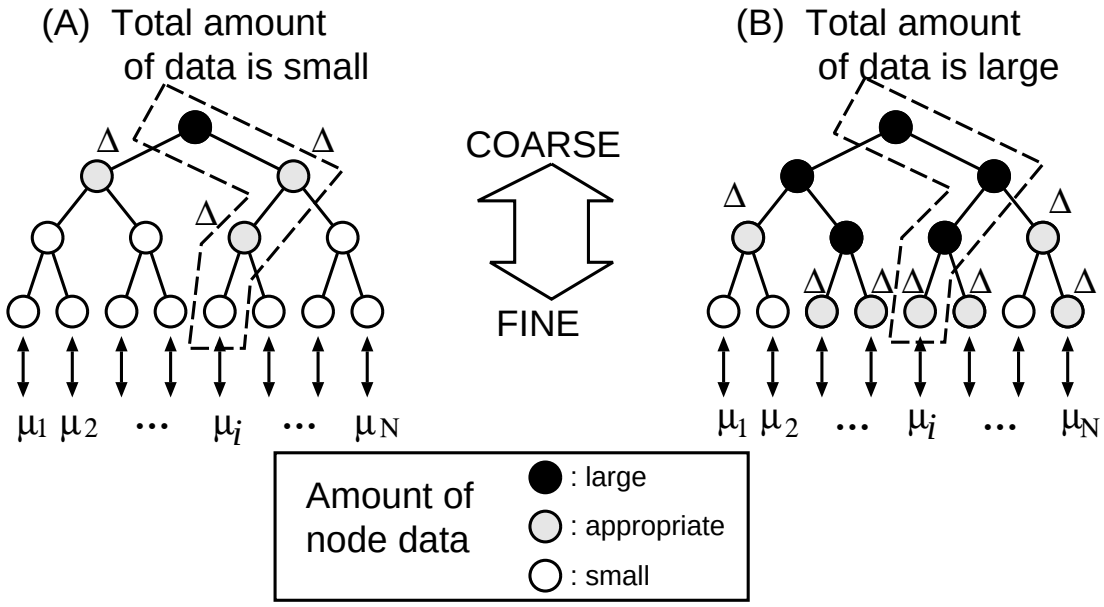

Figure 4.6: Speaker adaptation adaptation using autonomous model complexity control (TREE adaptation).

microphone difference is the main source of distortion on top of the speaker distortion due to non-native pronunciations and accents. In each figure two sets of results are shown for the SMAP method: mean-only adaptation (solid curve) in which the variances of the normalized pdfs were assumed to be the identity matrix $\mathbf{I}$ and only the means were estimated, and the other with adaptation for both means and variances (dotted curve). These figures clearly show that the SMAP method outperformed the MAP and TREE methods at almost every data point. The recognition rates for the SMAP method were much higher than those for MAP when the amount of data was small. With three adaptation utterances for TEL data, for example, the error rate reduction from the SI performance was 56%. This was much larger than the reduction obtained with the MAP method, which was only 1.7%. The recognition rates for the SMAP method became the same as those for the MAP method when the amount of data became large. The SMAP method also showed better recognition accuracy than that obtained with the TREE method not only when the amount of data was large but also when the amount of data was small. This is probably because parameter estimation

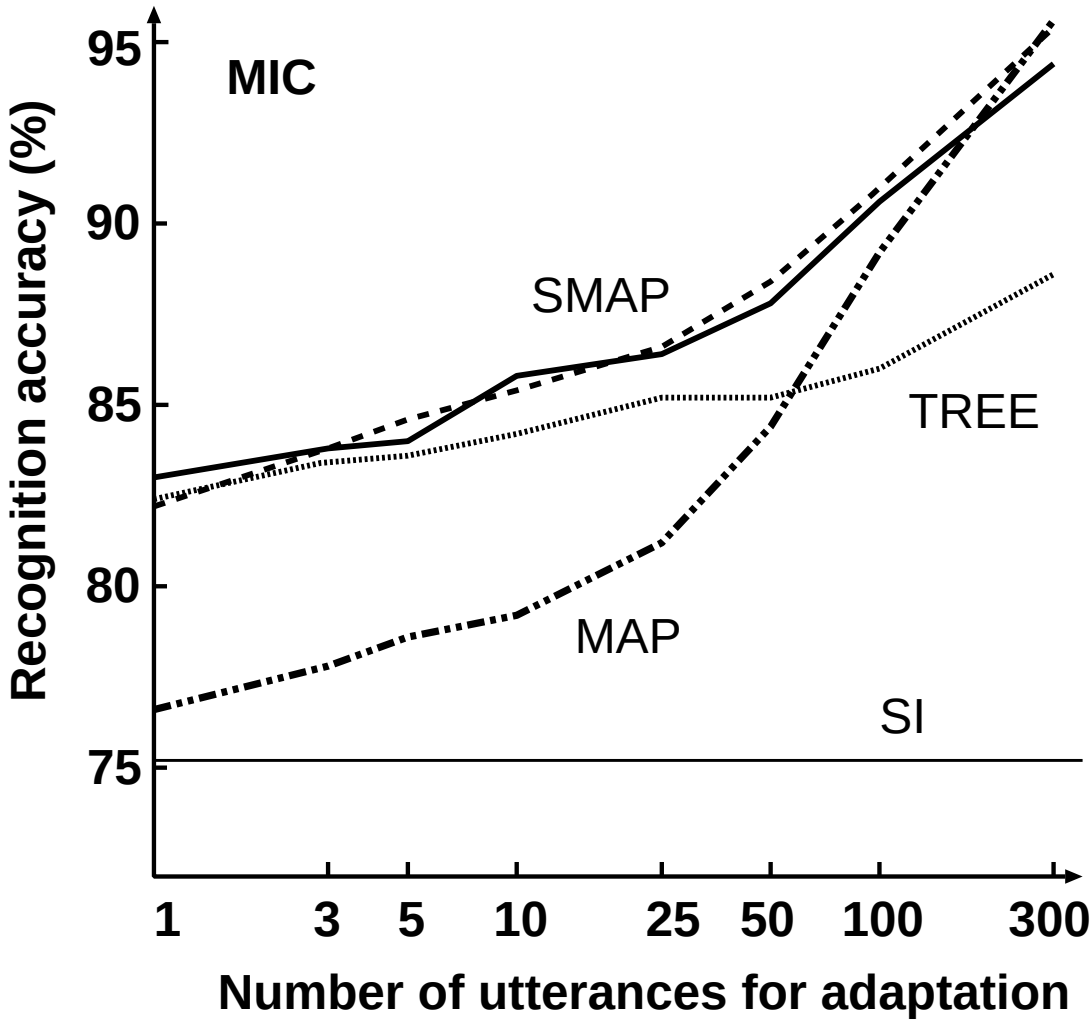

Figure 4.7: Recognition rates obtained with supervised adaptation when the MIC data were used. Two kinds of experiments were done with the SMAP method: mean-only adaptation (solid curve), and adaptation for both means and variances (dotted curve).

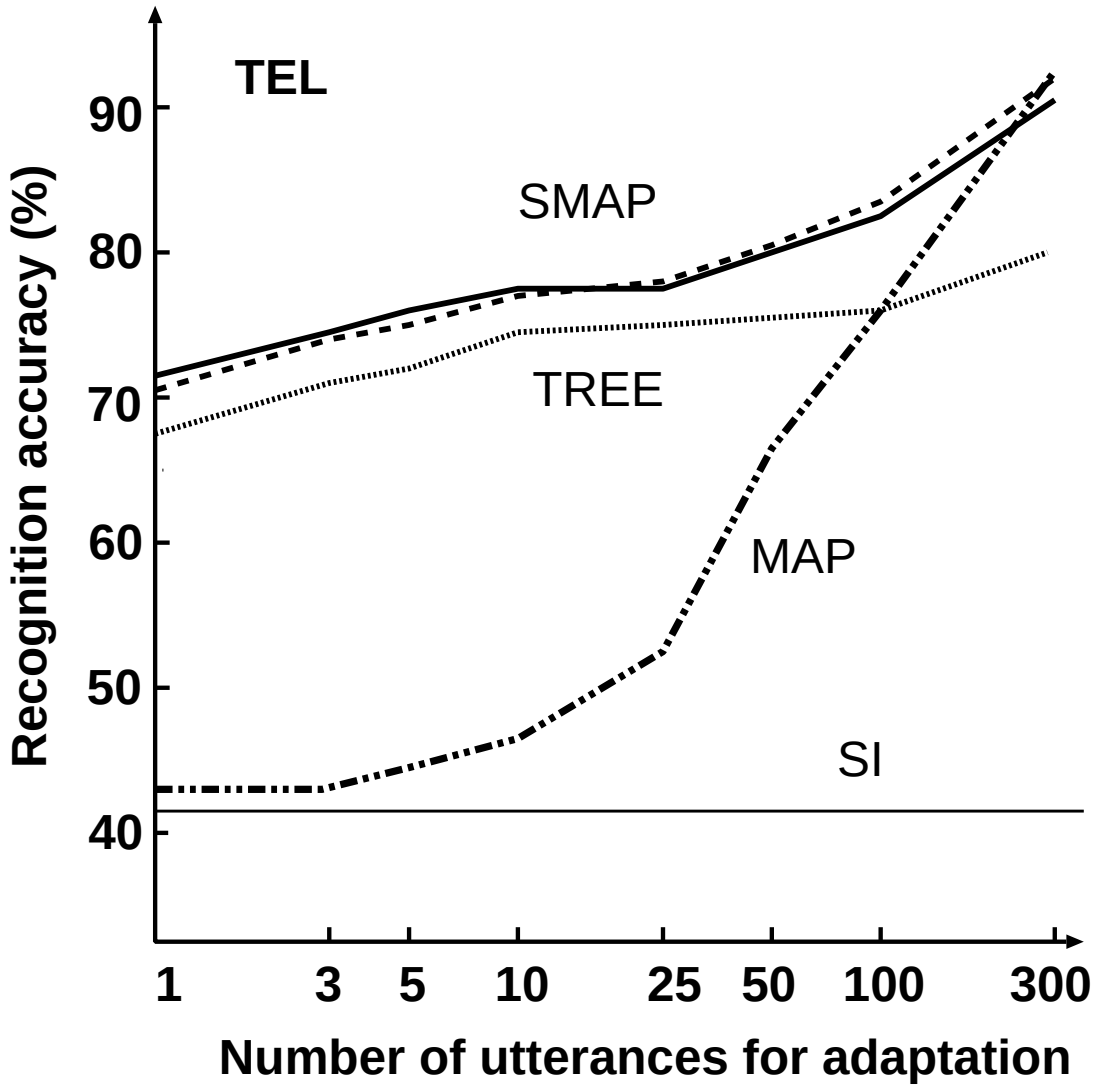

Figure 4.8: Recognition rates obtained with supervised adaptation when the TEL data were used. Two kinds of experiments were done with the SMAP method: mean-only adaptation (solid curve), and adaptation for both means and variances (dotted curve).

Table 4.4: Recognition rates (%) of each speaker obtained when using the SMAP method on MIC data.

No. of Utter.	A	B	C	D	E	Ave.
SI	74.8	51.2	74.9	85.7	89.3	75.2
1	83.0	68.5	88.6	82.9	88.1	82.2
3	82.9	70.1	89.2	86.8	89.6	83.7
5	84.8	69.7	89.3	88.7	90.1	84.5
10	83.8	72.8	90.8	89.2	90.4	85.4
25	86.2	76.3	88.3	90.5	91.3	86.5
50	88.3	78.4	91.6	92.2	91.7	88.4
100	90.1	84.5	93.2	94.1	91.7	90.7
300	95.0	92.8	96.4	97.6	94.9	95.3

was more robust than that in TREE, since a weighted sum of parameters from more than one layer was used. The mean-only adaptation result was better than adaptation of both means and variances when the amount of data was extremely small, but became worse as the amount of data became larger. This indicates that when the data amount was insufficient it is difficult to estimate both the mean and the variance as expected.

The results for each speaker, obtained when both means and variances were adapted are listed in Tables 4.4 and 4.5. The recognition rates were improved for all sizes of adaptation data and for all the speakers when TEL data were used as shown in Table 4.5. The improvement was especially more pronounced for those speakers with lower accuracy when SI models were used. With the MIC data, the recognition rates for speakers D and E were slightly less than the SI result when only one utterance was used but they soon became higher when more than three utterances were used.

Table 4.5: Recognition rates (%) of each speaker obtained when using the SMAP method on TEL data.

No. of Utter.	A	B	C	D	E	Ave.
SI	48.8	18.6	50.8	35.6	52.6	41.3
1	67.9	58.9	75.2	77.8	73.7	70.7
3	76.3	58.4	75.2	81.1	78.8	74.0
5	77.6	61.7	75.2	83.6	76.6	74.9
10	80.3	63.2	78.7	83.2	79.4	77.0
25	81.2	65.8	79.3	82.1	80.8	77.8
50	83.2	72.1	80.5	83.8	84.1	80.7
100	84.5	79.7	83.6	87.2	83.6	83.7
300	91.4	90.2	91.7	95.6	90.4	91.9

Table 4.6: Recognition rates (%) obtained with unsupervised adaptation for MIC data.

	A	B	C	D	E	Ave.
SI	74.8	51.2	74.9	85.7	89.3	75.2
SMAP	81.2	67.3	78.7	88.4	89.0	80.9

4.6.4 Unsupervised adaptation experiments

Unsupervised adaptation, in which no supervising information is available, is desirable in actual system operation. An on-line unsupervised adaptation scenario incorporating SMAP was therefore designed and evaluated [57, 59]. During unsupervised adaptation and testing, the parameters were estimated on a per utterance basis; only one utterance was used for unsupervised adaptation. First the test utterances were decoded by using the initial HMMs and then the parameters were estimated assuming this decoded word string is correct. In these

Table 4.7: Recognition rates (%) obtained with unsupervised adaptation for TEL data.

	A	B	C	D	E	Ave.
SI	48.8	18.6	50.8	35.6	52.6	41.3
SMAP	67.3	48.5	57.7	43.1	70.1	57.3

experiments, the variances for the normalized pdfs were fixed to be the identity matrices. The recognition results are listed in Tables 4.6 and 4.7. The error rate reduction was 23% for MIC and 27% for TEL. It should be noted that the effect of the SMAP method was larger for the speakers with lower SI recognition rates. For speaker B, for example, the error reduction rate was 33% for MIC and 37% for TEL.

The recognition rates in Table 4.6 are still rather low for any actual system usage. In the next experiment, ways to combine batch supervised adaptation typically used for fast enrollment of new speakers with on-line unsupervised adaptation was examined in order to raise recognition rates to a level high enough for practical usage. This combined adaptation process was carried out in two steps:

Step 1. Supervised adaptation using a set of adaptation data to generate seed models;

Step 2. Unsupervised adaptation using the test data, based on the above models.

The recognition rates (averaged over the five speakers) are listed in Table 4.8. In this experiment, the number of utterances used in Step 1 varied from 1 to 300. The supervised adaptation was carried out using the MIC or TEL adaptation data (labeled as SUP in Table 4.8) and unsupervised adaptation and recognition were also done for MIC or TEL test data (labeled as TEST in Table 4.8). In all cases, only one utterance was used for unsupervised adaptation (Step 2). In this table, the values listed in the column labeled S1 are the recognition rates

Table 4.8: Recognition rates (%) for combining supervised and unsupervised adaptation.

SUP	MIC				TEL			
TEST	MIC		TEL		MIC		TEL	
No. of Utter.	S1	S2	S1	S2	S1	S2	S1	S2
0(SI)	75.2	80.9	41.3	57.3	75.2	80.9	41.3	57.3
1	83.0	83.5	41.4	57.8	74.8	81.6	71.5	71.7
3	83.8	83.9	44.3	62.2	77.5	82.8	74.4	75.2
5	83.9	84.2	47.0	63.0	78.9	83.9	76.0	76.3
10	85.7	86.1	51.7	67.2	79.1	84.3	77.3	77.2
25	86.3	86.4	54.0	69.4	80.3	85.1	77.5	78.3
50	87.7	87.8	58.1	73.3	82.2	85.8	80.0	80.0
100	90.4	90.4	62.1	75.5	83.8	87.6	82.6	83.2
300	94.3	94.4	70.8	84.6	87.7	91.0	90.6	90.6

obtained with Step 1 only and the values listed in the column labeled S2 are the rates obtained after Step 2. Although this adaptation was only slightly effective when the acoustic conditions for the SUP data and that for the TEST data were similar, its effectiveness when acoustic conditions were different was clearly shown. For example, when the MIC data set was used for SUP and the TEL data set was used for TEST, the combined method (Step 1 and Step 2) required only three utterances for supervised adaptation to achieve 60% recognition accuracy, while the supervised adaptation (Step 1 only) needed 100 utterances. It was proved that this combined adaptation is especially effective when there are mismatches other than the speaker differences in the current adaptation scenario. It should also be noted that there was no degradation in the recognition performance of the combined strategy when the acoustic conditions for SUP and TEST were identical, i.e., when there was no mismatch between the adaptation data and the testing data.

4.7 Discussion and summary

The SMAP method for adaptation of HMM parameters enhances the performance of the conventional MAP method when the amount of data is small by utilizing a hierarchical structure in the model parameter space. Its effectiveness was confirmed in a set of recognition experiments using the speech data from non-native speakers collected through two different channels (MIC and TEL). In supervised adaptation, for example, with three utterances for TEL data, the error rate reduction was 56%. This was much better than the 1.7% reduction obtained with the MAP method. The SMAP method was also shown to be effective in unsupervised adaptation: with only one utterance for TEL data, the error rate reduction was 27%. In addition, the combination of supervised adaptation and on-line unsupervised adaptation greatly reduced the amount of data required for supervised adaptation. To obtain a recognition accuracy of 60% for the TEL data when using the MIC data for supervised adaptation, the combined method

required only three utterances whereas 100 utterances were needed to obtain 60% accuracy when using supervised adaptation only. It should also be noted that the SMAP method yields as high a recognition accuracy as the MAP method when the amount of data is sufficiently large.

Mismatches between training and testing conditions are caused by many differences, such as those between speakers, microphones, channels and noise levels. Many methods of compensating such mismatches have been developed but most of them focus on one or two differences. In an actual operational environment, however, more than one difference often contributes to the mismatch. The significance of each difference is usually unknown. In such cases, it is almost impossible to distinguish the contribution of one difference from that of another by using only the speech data. Furthermore, it is difficult to choose an effective combination of techniques, each of which is designed to compensate the effect of a particular difference. The SMAP method, on the other hand, compensates the mismatch as a whole; there is no need to specify any particular difference responsible for the mismatch. It is therefore more robust than the other methods when these differences are not known.

Recently, the idea of rapid adaptation was studied intensively at the 1998 Johns Hopkins University Summer Workshop and the results obtained were summarized in [14]. Tree structure dependency for speaker adaptation was one of the techniques explored during the workshop as a means of reducing the requirement for a large adaptation set [27]. The findings also support the proposed SMAP algorithm, especially in the area of unsupervised speaker adaptation.

This SMAP approach is quite general in its framework and can be easily applied to other adaptation methods. For example, SMAPLR, in which SMAP is applied to maximum *a posteriori* linear regression (MAPLR), was recently proposed and proved to be significantly better than MAPLR when the amount of adaptation data is extremely small [61]. The importance and effectiveness of this SMAP approach was also emphasized in recent surveys of speaker adaptation

studies [72, 33].

The SMAP method described here uses a tree structure in the model parameter space. While many kinds of tree structures can be used for SMAP estimation, it is important to choose one which represents the similarity of the normalized pdfs of the mixture components well. Good results were obtained when the Kullback-Leibler divergence between mixture components was used as a measure of similarity in constructing the tree structure, but many other similarity measures can be used. Other structures reflecting the relationship between acoustic model parameters are also worth investigating.

-

Chapter 5

Conclusion

This thesis has proposed a structural approach to robustness against data insufficiency. In this approach, a tree-structured model set is prepared and one node set (a model) is selected by using information-theoretic criteria. Three key factors in this approach are the design of the root layer and the leaf layer, the method of constructing the tree structure, and the node selection framework. This approach was applied to two problems in speech recognition: acoustic modeling and speaker adaptation.

For acoustic modeling, a phonetic decision tree, in which the root is a monophone and the leaves are triphones, was constructed and the optimal model was selected by using the MDL criterion. This approach achieved recognition more accurate than that obtained when a conventional approach was used and it had a much lower overall computational cost in an evaluation made through a series of recognition experiments.

For speaker adaptation, a tree of Gaussian pdfs was constructed by using Kullback-Leibler divergence. Its leaves correspond to all the mixture components in CDHMMs and the root node corresponds to the pdf shared among all of them. The optimal model was selected by using SMAP estimation. It reduced the error rate by half using only three utterances for adaptation and it yielded the same accuracy as conventional MAP and ML estimation when the amount of data was

sufficiently large in the evaluational experiments.

5.1 Contribution of the thesis

The contribution of this thesis for speech recognition studies is the following:

- It has proved the importance of the described structural approach in statistical pattern recognition. This approach can be applied to many problems incurred due to data insufficiency.
- It has proved that the use of information criteria is an effective approach to dealing with acoustic modeling problems in speech recognition. Among the information criteria, the MDL criterion was focused on and its effectiveness was extensively explored.
- It has shown several examples of tree-structured models, which performed well in the structural approach. While many other models may possibly be used, those described in this thesis demonstated particularly good performance.
- It offers excellent methods for acoustic modeling and speaker adaptation; MDL acoustic modeling and SMAP adaptation. Both have been extensively studied by many researchers since they were first published.

5.2 Future research directions

Although I believe the investigations reported in this thesis represent some significant progress, further research is clearly still needed.

- The MDL-based modeling can be applied not only to the clustering method using a phonetic decision tree but also to other clustering methods for acoustic modeling. It can also be easily applied to the language modeling in

speech recognition, and some studies relevant to this point have appeared recently (e.g., [62]).

- The MDL criterion can be used to compare various classes of probabilistic models, since it utilizes only the model complexity and the likelihood of the probabilistic model for the data. For example, it can be used to decide what class of models can best be used for the output pdfs of HMMs. Furthermore, it can be used to carry out clustering across more than one parameter set. For example, it enables the similtanuous clustering of parameters for transtition probabilities and those for output probabilities.

- The SMAP adaptation is quite general in its framework, and can easily be applied to language adaptation. The n-gram language model, for example, can be used to construct a tree structure in which each leaf node represents an n-gram and an (n-1)-gram is assigned to the parent node. In a trigram model(e.g., [24]), a four-layer tree is constructed; for each word the root node corresponds to uniform distribution for all the vocabulary, the parent node in the first layer corresponds to the unigram, the parent node in the second layer corresponds to the bigram, and the leaf node corresponds to the trigram.

- While this structural approach is not directly aimed at increasing the recognition accuracy under ideal conditions, it will be helpful for research in this direction. This approach keeps computational costs low because it makes it possible to easily obtain a model whose size is appropriate for the given amount of training data.

- It should be noted that noisy data are not dealt with in this thesis. It is well known that environmental noise and channel distortion can seriously degrade speech recognition. In such cases, the likelihood of data becomes more important in model selection. While one method described in this thesis used the MDL criterion, in which the likelihood is taken into account,

further evaluation using actual noisy data is still needed.

- This approach utilizes the embedded structure of the parameter space. In the acoustic modeling, it is assumed that the structure of the parameter space can be represented by phonetic decision trees. And in the speaker adaptation, it is assumed that the Kullback-Leibler divergence can be used as the measure of the distance between the Gaussian pdfs. While these assumptions were justified by the evaluation, it may be possible to obtain more accurate representations of the embedded structure.

Bibliography

[1] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automatic Control*, vol. 19, pp. 716-723, 1974.

[2] B. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *J. Acoust. Soc. Am.*, vol. 55, pp. 1304-1312, 1974.

[3] L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo, and M. A. Picheny, "Decision trees for phonological rules in continuous speech," in *Proc. ICASSP-91*, Toronto, pp. 185-188, 1991.

[4] J. O. Berger, *Optimal Statistical Decisions*, in Springer Series in Statistics, Springer-Verlag, 1980.

[5] P. F. Brown, V. J. D. Pietra, P. V. deSouza, J. C. Lai, and L. R. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp.467-479, 1993.

[6] S. S. Chen and P. S. Gopalakrishnan, "Clustering via the Bayesian information criterion with applications in speech recognition," in *Proc. ICASSP-98*, Seattle, pp. 645-648, 1998.

[7] S. S. Chen, E. M. Eide, M. J. F. Gales, R. A. Gopinath, D. Kanevsky, and P. A. Olsen, "Recent improvements to IBM's speech recognition system for automatic transcription of broadcast news," in *Proc. ICASSP-99*, Phoenix, 1999.

[8] S. S. Chen and R. A. Gopinath, "Model selection in acoustic modeling," in *Proc. EuroSpeech-99*, Budapest, pp. 1087-1090, 1999.

[9] J.-T. Chien, C.-H. Lee, and H.-C. Wang, "Improved Bayesian learning of hidden Markov models for speaker adaptation," in *Proc. ICASSP-97*, Munich, pp. 1027-1039, 1997.

[10] M. H. DeGroot, *Statistical Decision Theory and Bayesian Analysis*, McGraw-Hill, 1970.

[11] A. P. Dempster, N. M.Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.*, vol. 39, no. 1, pp. 1-38, 1977.

[12] V. V. Digalakis, P. Monaco, H. Murveit, "Genons: generalized mixture tying in continuous hidden Markov model-based speech recognizers," *IEEE Trans. Speech Audio Processing*, vol. 4, no. 4, pp. 281-289, 1996.

[13] V. V. Digalakis and L. G. Neumeyer, "Speaker adaptation using combined transformation and Bayesian methods," *IEEE Trans. Speech Audio Processing*, vol. 4, no. 4, pp. 294-300, 1996.

[14] V. V. Digalakis, S. Berkowitz, E. L. Bocchieri, C. Boulis, W. J. Byrne, H. Collier, A. Corduneanu, A. Kannan, S. P. Khudanpur, and A. Sankar, "Rapid speech recognizer adaptation to new speakers," in *Proc. ICASSP-99*, Phoenix, pp. 765-768, 1999.

[15] J. L. Flanagan, *Speech Analysis, Synthesis, and Perception, 2nd Edition*, Springer-Verlag, 1972.

[16] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 34, pp. 52-59, 1986.

[17] S. Furui, "Unsupervised speaker adaptation method based on hierarchical spectral clustering," in *Proc. ICASSP-89*, Glasgow, pp. 286-289, 1989.

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

[19] I. J. Good, "The population frequencies of species and the estimation of population parameters," *Biometrica*, vol. 40, pp. 237-264, 1953.

[20] Q. Huo and C.-H. Lee, "On-line adaptive learning of the continuous density hidden Markov model based on approximate recursive Bayes estimate," *IEEE Trans. Speech Audio Processing*, vol. 5, no. 2, pp. 161-172, 1997.

[21] Q. Huo and C.-H. Lee, "On-line adaptive learning of the correlated continuous-density hidden Markov model for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 6, no. 4, pp. 386-397, 1998.

[22] M.-Y. Hwang, X. Huang, and F. Alleva, "Predicting unseen triphones with senones," *Proc. ICASSP-93*, Minneapolis, pp. II-311-314, 1993.

[23] F. Jelinek, "Self-organized language modeling for speech recognition," in *Readings in Speech Recognition*, A. Waibel and K. F. Lee, eds., Morgan-Kaufmann, pp. 450-506, 1990.

[24] F. Jelinik, *Statistical method for speech recognition*, MIT Press, 1998.

[25] H. Jiang, K. Hirose, and Q. Huo, "Robust speech recognition based on Viterbi Bayesian predictive classification," in *Proc. ICASSP-97*, Munich, pp. 1551-1554, 1997.

[26] A. Kannan, M. Ostendorf and J. Rohlicek, "Maximum likelihood clustering of Gaussians for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 2, no. 3, pp. 453-454, 1994.

[27] A. Kannan and S. P. Khudanpur, "Tree-structured models of parameter dependence for rapid adaptation in large vocabulary conversational speech recognition," in *Proc. ICASSP-99*, Phoenix, pp. 769-772, 1999.

[28] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, no. 3, pp. 400-401, 1987.

[29] R. Kuhn and R. De Mori, "A cache-based natural language model for speech recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, pp. 570-583, 1990.

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

[31] C.-H. Lee, E. Giachin, L. R. Rabiner, R. Pieraccini, and A. E. Rosenberg, "Improved acoustic modeling for large vocabulary continuous speech recognition," *Computer Speech and Language*, vol. 6, no. 2, pp. 103-207, 1992.

[32] C.-H. Lee, "On stochastic feature and model compensation approaches to robust speech recognition," *Speech Communication*, vol. 25, pp. 29-47, 1998.

[33] C.-H. Lee and Q. Huo, "On adaptive decision rules and decision parameter adaptation for automatic speech recognition," in *Proc. of IEEE*, vol. 88, no. 8, pp. 1241-1269, 2000.

[34] K.-F.Lee, S. Hayamizu, H.-W. Hon, C. Huang, J. Swartz, and R. Weide, "Allophone clustering for continuous speech recognition," in *Proc. ICASSP-90*, Albuquerque, pp. 749-753, 1990.

[35] 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.

[36] K. Ohkura, M. Sugiyama, and S. Sagayama, "Speaker adaptation based on transfer vector field smoothing with continuous mixture density HMMs," in *Proc. ICSLP-92*, Alberta, pp. 369-372, 1992.

[37] M. Ostendorf and H. Singer, "HMM topology design using maximum likelihood successive state splitting," *Computer Speech and Language*, vol. 11, pp. 17-41, 1997.

[38] D.-B. Paul, "Extensions to phone-state decision-tree clustering: single-tree and tagged clustering," in *ICASSP-97*, Munich, pp. 1487-1490, 1997.

[39] F. Pereira, N. Tishby, L. Lee, "Distributional clustering of English words," in *Proc. ACL-93*, Columbus, Ohio, pp.183-190, 1990.

[40] P. Price, W. Fisher, J. Bernstein, and D. Pallett, "A database for continuous speech recognition in a 1000-word domain," in *Proc. ICASSP-88*, New York City, pp. 651-654, 1988.

[41] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[42] M. Rahim and B.-H. Juang, "Signal bias removal by maximum likelihood estimation for robust telephone speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, no. 1, pp. 19-30, 1996.

[43] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Trans. Information Technology*, vol. 30, no. 4, pp. 629-636, 1984.

[44] A. Sankar and C-.H. Lee, "A maximum likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, no. 3, pp. 190-202, 1996.

[45] G. Schwartz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461-464, 1978.

[46] R. M. Schwartz, Y. Chow, S. Roucos, M. Kransner, and J. Makhoul, "Improved hidden Markov modeling of phonemes for continuous speech recognition," in *Proc. ICASSP84*, San Diego, 35.6, 1984.

[47] R. Shibata, "Selection of the order of an autoregressive model by Akaike's information criterion," *Biometrica*, vol. 63, pp. 117-126, 1976.

[48] R. Shibata, "Asymptotically efficient selection of the order of the model for estimating parameters of a linear process," *Annals of Statistics*, vol. 8, pp. 147-164, 1980.

[49] R. Shibata, "Optimal selection of regression variables, " *Biometrica*, vol. 68, pp. 45-54, 1981.

[50] K. Shinoda, K. Iso, and T. Watanabe, "Speaker adaptation for demi-syllable based speech recognition using continuous HMM," in *Proc. of ICSLP-90*, Kobe, pp. 261-264, 1990.

[51] K. Shinoda, K. Iso, and T. Watanabe, "Speaker adaptation for demi-syllable-based continuous-density HMM," in *Proc. ICASSP-91,* Toronto, pp. 857-860, 1991.

[52] K. Shinoda and T. Watanabe, "Unsupervised speaker adaptation for speech recognition using demi-syllable HMM," in *Proc. ICSLP-94*, Yokohama, pp. 435-438, 1994.

[53] K. Shinoda and T. Watanabe, "Speaker adaptation with autonomous control using tree structure," in *Proc. EuroSpeech-95*, Madrid, pp. 1143-1146, 1995.

[54] K. Shinoda and T. Watanabe, "Speaker adaptation with autonomous model complexity control by MDL principle," in *Proc. ICASSP-96*, Atlanta, pp.717-720, 1996.

[55] K. Shinoda and T. Watanabe, "Acoustic modeling based on the MDL criterion for speech recognition," in *Proc. EuroSpeech-97*, Rhodes, vol. 1, pp. 99-102, 1997.

[56] K. Shinoda and C.-H. Lee, "Structural MAP speaker adaptation using hierarchical priors," *Proc. IEEE Workshop on Speech Recognition and Understanding*, Santa Barbara, pp. 381-387, 1997.

[57] K. Shinoda and C.-H. Lee, "Unsupervised adaptation using structural Bayes approach," in *Proc. ICASSP-98*, Seattle, pp. II 793-796, 1998.

[58] K. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition," *J. Acoust. Soc. Jpn.(E)*, vol. 21, no. 2, pp.79-86, 2000.

[59] K. Shinoda and C.-H. Lee, "A structural Bayes approach to speaker adaptation", *IEEE Trans. Speech Audio Processing* (to appear March 2001).

[60] O. Siohan, C. Chesta, and C.-H. Lee, "Hidden Markov model adaptation using maximum *a posteriori* linear regression," *Proc. Workshop on Robust Methods for Speech Recognition in Adverse Conditions*, Tampere, Finland, pp.147-150, 1999.

[61] O. Siohan, T. A. Myrvoll, and C.-H.Lee, "Structural maximum *a posteriori* linear regression for fast HMM adaptation," *Proc. ISCA ITRW ASR2000 Workshop*, Paris, 2000.

[62] M. Siu and M. Ostendorf, "Variable n-grams and extensions for conversational speech language modeling," *IEEE Trans. Speech Audio Processing*, vol. 8, no. 1, pp. 63-75, 2000.

[63] R. M. Stern and M. J. Lasry, "Dynamic speaker adaptation for feature-based isolated word recognition," *IEEE Trans. Speech Audio Processing*, vol. 35, no. 6, pp. 751-763, 1987.

[64] M. Sugiyama and H. Ogawa, "A new information criterion for the selection of subspace models," in *Proc. ESANN*, Bruges, pp. 69-74, 2000.

[65] A. C. Surendran, C.-H. Lee, and M. Rahim, "Non-linear compensation for stochastic matching," *IEEE Trans. Speech Audio Processing*, vol. 7, no. 6, pp. 643-655, 1999.

[66] A. C. Surendran and C.-H. Lee, "Bayesian predictive approach to adaptation of HMMs," in *Proc. Workshop Robust Methods for Speech Recognition in Adverse Conditions*, Tampere, Finland, pp.155-158, 1999.

[67] J. Takahashi and S. Sagayama, "Vector field smoothed Bayesian learning for incremental speaker adaptation," in *Proc. ICASSP-95*, Detroit, pp. 696-699, 1995.

[68] J. Takami and S. Sagayama, "A successive state splitting algorithm for efficient allophone modeling," in *Proc. ICASSP-92*, San Francisco, pp. I-573-576, 1992.

[69] M. Tonomura, T. Kosaka, and S. Matsunaga, "Speaker adaptation based on transfer vector field-smoothing using maximum *a posteriori* probability estimation," in *Proc. ICASSP-95*, Detroit, pp. 688-691, 1995.

[70] A. Tritschler and R. Gopinath, "Improved speaker segmentation and segments clustering using the Bayesian information criterion," in *EuroSpeech-99*, Budapest, pp. 679-682, 1999.

[71] T. Watanabe, K. Shinoda, K. Takagi, and E. Yamada, "Speech recognition using tree-structured probability density function," in *Proc. ICSLP-94*, Yokohama, pp. 223-226, 1994.

[72] P. C. Woodland, "Speaker adaptation: techniques and challenges," in *Proc. IEEE Automatic Speech and Understanding Workshop '99*, Keystone, vol. 1, pp. Page 85-90, 1999.

[73] S. J. Young, "The general use of tying in phoneme-based HMM speech recognizers," in *Proc. ICASSP92*, San Francisco, pp. 569-572, 1992.

[74] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modeling," in *Proc. Human Language Technology*, pp. 307-312, 1994.

[75] G. Zavaliagkos, R. Schwartz, and J. McDonough, "Maximum *a posteriori* adaptation for large-scale HMM recognizers," in *Proc. ICASSP-95*, Detroit, pp. 725-728, 1995.

Appendix A

Derivation of Description Length

The description length is defined as the cord length required for encoding a compound information source [43]. The MDL criterion states that the probabilistic model which minimizes the description length is the optimal model for given data. Let us assume a set of models $\{1, \ldots, i, \ldots, I\}$ and data $\mathbf{x}^N = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are given. The description length $DL(i)$ for model i is the sum of three parts: the data description length $DL1(i)$, the parameter description length $DL2(i)$, and the model description length $DL3(i)$,

$$DL(i) = DL1(i) + DL2(i) + DL3(i). \tag{A.1}$$

$DL1$ is the description length given the data and the model, $DL2$ is the description length required for coding the parameter set for the model, and $DL3$ is the description length for model selection.

In calculating $DL3$, it is usually assumed that the probability distribution over the model set is uniform,

$$P(i) = \frac{1}{I}, \qquad i = 1, \ldots, I. \tag{A.2}$$

Then, $DL3(i)$ is simply calculated as follows,

$$DL3(i) = \log I. \tag{A.3}$$

From now on, we focus on DL for each model. The suffix i identifying the model is therefore omitted.

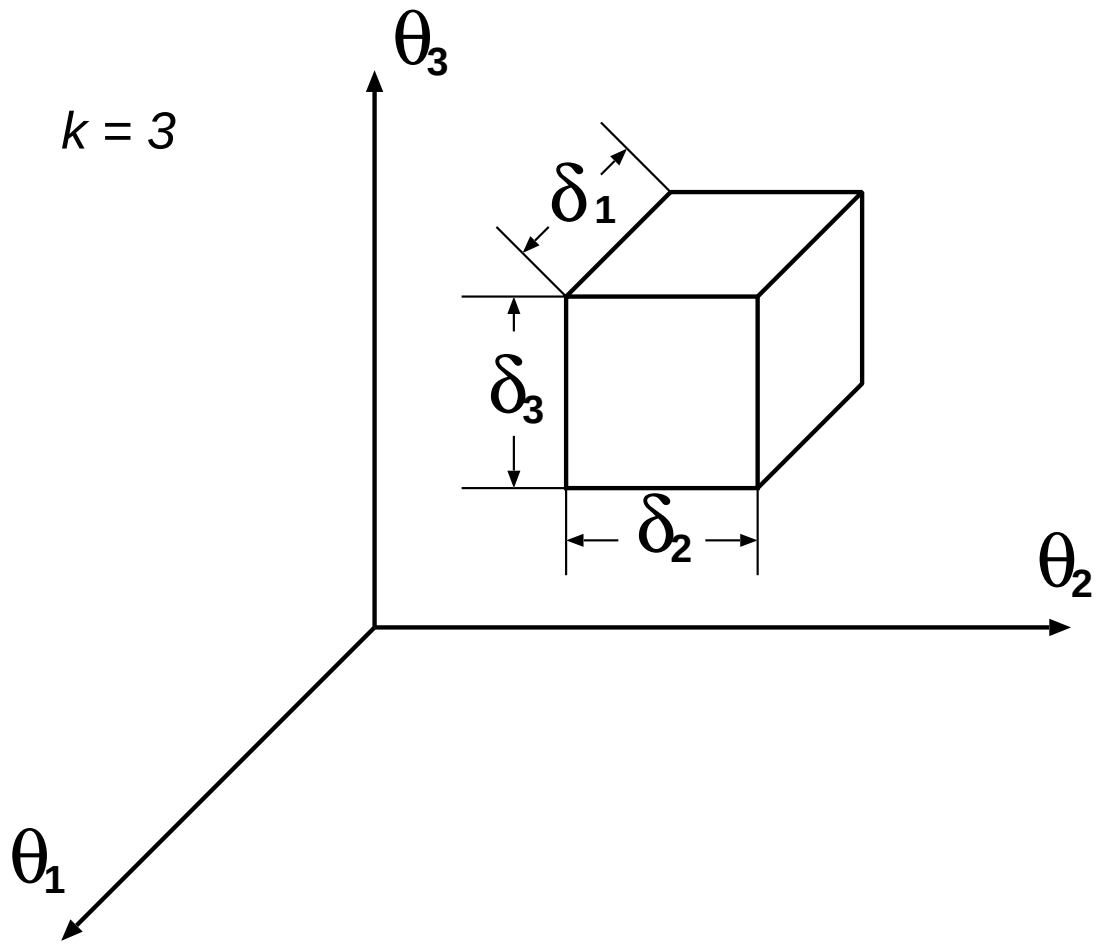

Figure A.1: Quantization of parameter space.

Next, let us calculate $DL1$ for a model with parameter set θ. $DL1$ is the negative of the log-likelihood of the model with respect to given data, and is minimum when the maximum likelihood estimate for the parameter set, $\hat{\theta}$, is used.

$$DL1 = -\log P_{\hat{\theta}}(\mathbf{x}^N). \tag{A.4}$$

We next consider the minimization of the sum of $DL1$ and $DL2$ by choosing the parameter set. Assuming that the dimension of the parameter space is K, then θ is a vector with K real-valued components:

$$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)^T. \tag{A.5}$$

Since each element of the parameter vector $\boldsymbol{\theta}$ is a real number, an infinite cord length is needed to encode the parameter vector. One possible way to deal with this difficulty is to discretize the parameter space. The parameter space can be discretized by defining a cell as a k-dimensional rectangular solid having length δ_k on the axis θ_k (see Figure A.1). The parameter vector $\hat{\boldsymbol{\theta}}$ is mapped to a representative $\tilde{\boldsymbol{\theta}}$ in the same cell. If the volume of the parameter space is V, then we have $V/(\delta_i \cdots \delta_k)$ cells. The sum of $DL1$ and $DL2$ is then,

$$DL1 + DL2 = \min_{\boldsymbol{\delta}} D(\boldsymbol{\delta}). \tag{A.6}$$

$$D(\boldsymbol{\delta}) = -\log P_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}^N) + \log \frac{V}{\delta_1 \cdots \delta_K}, \tag{A.7}$$

where $\boldsymbol{\delta} = \{\delta_1, \ldots, \delta_K\}$.

The minimization process of $D(\boldsymbol{\delta})$ over $\boldsymbol{\delta}$ is carried out as follows. We first make Taylor's expansion of the first term,

$$-\log P_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}^N) = -\log P_{\hat{\boldsymbol{\theta}}}(\mathbf{x}^N) + \left.\frac{\partial(-\log P_{\boldsymbol{\theta}}(\mathbf{x}^N))}{\partial \boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}} \cdot \boldsymbol{\delta} + \frac{1}{2} N \cdot \boldsymbol{\delta}^T \cdot I_N(\hat{\boldsymbol{\theta}}) \cdot \boldsymbol{\delta} + O(N \cdot \delta^3), \tag{A.8}$$

$$I_N(\hat{\boldsymbol{\theta}}) = \left.\frac{\partial^2(-\frac{1}{N}\log P_{\boldsymbol{\theta}}(\mathbf{x}^N))}{\partial^2 \boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}}. \tag{A.9}$$

The second term of (A.8) equals 0 because $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimate. Under certain suitable conditions, $I_N(\hat{\boldsymbol{\theta}})$ is converged to a K-dimensional matrix of constants, $I(\hat{\boldsymbol{\theta}})$, known as the Fisher information matrix, when $N \to \infty$. Then,

$$D(\boldsymbol{\delta}) \simeq -\log P_{\hat{\boldsymbol{\theta}}}(\mathbf{x}^N) + \frac{1}{2} \cdot \boldsymbol{\delta}^T \cdot I(\hat{\boldsymbol{\theta}}) \cdot \boldsymbol{\delta} + \log \frac{V}{\delta_1 \cdots \delta_K}. \tag{A.10}$$

Differentiating this formula with each δ_k and setting a result equal to 0 for each, we obtain the following equations,

$$(N \cdot I(\hat{\boldsymbol{\theta}}) \cdot \boldsymbol{\delta})_k - \frac{1}{\delta_k} = 0, \qquad k = 1, \ldots, K. \tag{A.11}$$

Suppose that the eigenvalues of $I(\hat{\boldsymbol{\theta}})$ are $\lambda_1, \ldots, \lambda_K$, and the eigenvectors are $(u_1, \ldots, u_K)$. If we only consider the case when the axes of a cell are in parallel

with the eigenvectors, then (A.11) becomes,

$$N \cdot \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_K \end{pmatrix} \begin{pmatrix} \delta_1 \\ \vdots \\ \delta_K \end{pmatrix} = \begin{pmatrix} \frac{1}{\delta_1} \\ \vdots \\ \frac{1}{\delta_K} \end{pmatrix}. \tag{A.12}$$

Hence, we have

$$\delta_k = \frac{1}{\sqrt{N \cdot \lambda_k}}, \tag{A.13}$$

and then,

$$N \cdot \boldsymbol{\delta}^T \cdot I(\hat{\boldsymbol{\theta}}) \cdot \boldsymbol{\delta} = K. \tag{A.14}$$

Moreover, since $\lambda_1 \cdots \lambda_k = |I(\hat{\boldsymbol{\theta}})|$, we have

$$\frac{1}{\delta_1 \cdots \delta_K} = \sqrt{N}^K \cdot \sqrt{|I(\hat{\boldsymbol{\theta}})|}. \tag{A.15}$$

Using (A.10), (A.14), and (A.15), (A.6) becomes,

$$\begin{aligned} DL1 + DL2 &\simeq -\log P_{\hat{\boldsymbol{\theta}}}(\mathbf{x}^N) + \log(V \cdot \sqrt{n}^K \cdot \sqrt{|I(\hat{\boldsymbol{\theta}})|}) + \frac{K}{2} + O(\frac{1}{\sqrt{N}}) \\ &= -\log P_{\hat{\boldsymbol{\theta}}}(\mathbf{x}^N) + \frac{K}{2}\log N + O(1). \end{aligned} \tag{A.16}$$

Finally, the total description length is,

$$\begin{aligned} DL(i) &= DL1(i) + DL2(i) + DL3(i) \\ &\simeq -\log P_{\hat{\boldsymbol{\theta}}^{(i)}}(\mathbf{x}^N) + \frac{K^{(i)}}{2}\log N + \log I, \end{aligned} \tag{A.17}$$

where $\hat{\boldsymbol{\theta}}^{(i)}$ is the maximum likelihood estimate of the parameter set of model i, and $K^{(i)}$ is the dimension of model i.

Appendix B

Maximum A Posteriori Estimation

We consider the case where the parametric form of the probabilistic density function (pdf) $p(x)$, where x is a k-component vector-valued random variable, is the multivariate Gaussian pdf,

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{k}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right], \tag{B.1}$$

while neither the mean vector $\boldsymbol{\mu}$ nor the variance $\boldsymbol{\Sigma}$ are known. Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a set of observed samples, which are assumed to be independent and identically distributed (i.i.d.). Our goal is to estimate the parameter set $\theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ by using the observation samples $\mathcal{X}$.

Maximum likelihood (ML) estimation is often used for this purpose. In the ML estimation, the parameter set which maximizes the following likelihood function is chosen,

$$f(\mathcal{X}|\theta) = \prod_{n=1}^{N} p(\mathbf{x}_n|\theta). \tag{B.2}$$

The resulting maximum likelihood estimate, $\tilde{\theta} = \{\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}\}$, is calculated as follows,

$$\tilde{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n, \tag{B.3}$$

$$\tilde{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{x}_n - \tilde{\boldsymbol{\mu}})(\mathbf{x}_n - \tilde{\boldsymbol{\mu}})^T. \tag{B.4}$$

In the maximum *a posteriori* (MAP) estimation [10, 18], it is assumed that the parameter set θ is a random vector in the parameter space and it has a prior distribution $p(\theta)$. Let $p(\theta|\mathcal{X})$ be the posterior pdf that is obtained after the observation of $\mathcal{X}$. Then, using Bayes' rule,

$$\begin{aligned} p(\theta|\mathcal{X}) &= \frac{p(\mathcal{X}|\theta)p(\theta)}{\int p(\mathcal{X}|\theta)p(\theta)d\theta} \\ &= C\prod_{n=1}^{N} p(\mathbf{x}_n|\theta)p(\theta), \end{aligned} \tag{B.5}$$

where C is a scale factor that depends on $\mathcal{X}$ but is independent of θ. The MAP estimate $\hat{\theta}$ is defined as the mode of the posterior pdf,

$$\begin{aligned} \hat{\theta} &= \text{argmax}_\theta p(\theta|\mathcal{X}) \\ &= \text{argmax}_\theta \prod_{n=1}^{N} p(\mathbf{x}_n|\theta)p(\theta). \end{aligned} \tag{B.6}$$

The choice of the prior pdf is a key issue in MAP estimation. Mainly from the viewpoint of tractability, the conjugate prior pdf is often used; when using it, the resulting posterior pdf is in the same family as the one that the prior pdf belongs to. One such pdf for the multivariate Gaussian pdf $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the normal-Wishart density of the form,

$$\begin{aligned} &g(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \alpha, \tau) \propto \\ &\quad |\boldsymbol{\Sigma}|^{-\frac{\alpha-k}{2}} \exp\left[\frac{\tau}{2}(\boldsymbol{\mu}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_0)\right] \exp\left[-\frac{1}{2}\text{tr}(\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}^{-1})\right], \end{aligned} \tag{B.7}$$

where $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \alpha, \tau)$ are the prior density parameters such that $\alpha > k-1$, $\tau > 0$, $\boldsymbol{\mu}_0$ is a vector of dimension k, and $\boldsymbol{\Sigma}_0$ is a $k \times k$ positive definite matrix.

Then, the MAP estimate $\hat{\theta} = \{\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\}$ is the one that maximizes the following function,

$$g(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathcal{X}) = \prod_{n=1}^{N} p(\mathbf{x}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma})g(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{B.8}$$

After simple calculations, we get

$$\hat{\boldsymbol{\mu}} = \frac{\tau\boldsymbol{\mu}_0 + \sum_{n=1}^{N}\mathbf{x}_n}{\tau + N}, \tag{B.9}$$

$$\hat{\boldsymbol{\Sigma}} = \frac{\boldsymbol{\Sigma}_0 + \sum_{n=1}^{N}(\mathbf{x}_n - \hat{\boldsymbol{\mu}})(\mathbf{x}_n - \hat{\boldsymbol{\mu}})^T + \tau(\boldsymbol{\mu}_0 - \hat{\boldsymbol{\mu}})(\boldsymbol{\mu}_0 - \hat{\boldsymbol{\mu}})^T}{(\alpha - k) + N}. \tag{B.10}$$

It should be noted that as the number of samples, N, increases, the MAP estimate $\hat{\theta} = \{\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\}$ approaches the ML estimate $\tilde{\theta} = \{\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}\}$.

Publication List

Reviewed Journal Papers

1. K. Shinoda, K. Iso, and T. Watanabe, "Speaker adaptation using spectral interpolation for speaker adaptation," *IEICE Trans. Fundamentals*, vol. J77-A, no. 2, pp. 120-127, 1994 (in Japanese).

2. K. Shinoda and T. Watanabe, "Speaker adaptation using autonomous model complexity control for speech recognition," *IEICE Trans. Inf., Syst.*, vol. J79-D-II, no. 12, pp. 2054-2061, 1996 (in Japanese).

3. K. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition," *J. Acoust. Soc. Jpn.(E)*, vol. 21, no. 2, pp.79-86, 2000.

4. T. Emori and K. Shinoda, "Vocal tract length normalization using rapid maximum-likelihood estimation for speech recognition," *IEICE Trans. Inf., Syst.*, vol. J83-D-II, no. 11, 2000 (in Japanese).

5. K. Shinoda and C.-H. Lee, "A structural Bayes approach to speaker adaptation", *IEEE Trans. Speech Audio Processing* (to appear March 2001).

Reviewed Conference Papers

1. K. Shinoda, K. Iso, and T. Watanabe, "Speaker adaptation for demi-syllable based speech recognition using continuous HMM," in *Proc. of ICSLP-90*, Kobe, pp. 261-264, 1990.

2. K. Shinoda, K. Iso, and T. Watanabe, "Speaker Adaptation for Demi-Syllable-Based Continuous-Density HMM," in *Proc. ICASSP-91,* Toronto, pp. 857-860, 1991.

3. K. Shinoda and T. Watanabe, "Unsupervised speaker adaptation for speech recognition using demi-syllable HMM," in *Proc. ICSLP-94*, Yokohama, pp. 435-438, 1994.

4. T. Watanabe, K. Shinoda, K. Takagi, and E. Yamada, "Speech recognition using tree-structured probability density function," in *Proc. ICSLP-94*, Yokohama, pp. 223-226, 1994.

5. T. Watanabe, K. Shinoda, K. Takagi, and K. Iso, "High speed speech recognition using tree-structured probability density function," in *Proc. ICASSP-95*, Detroit, pp. 556-559, 1995.

6. K. Shinoda and T. Watanabe, "Speaker adaptation with autonomous control using tree structure," in *Proc. EuroSpeech-95*, Madrid, pp. 1143-1146, 1995.

7. K. Shinoda and T. Watanabe, "Speaker adaptation with autonomous model complexity control by MDL principle," in *Proc. ICASSP-96*, Atlanta, pp.717-720, 1996.

8. K. Takagi, K. Shinoda, H. Hattori and T. Watanabe, "Unsupervised and incremental speaker adaptation under adverse environmental conditions," in *Proc. ICSLP-96*, Atlanta, pp. 2079-2082, 1996.

9. K. Shinoda and T. Watanabe, "Acoustic modeling based on the MDL criterion for speech recognition," in *Proc. EuroSpeech-97*, Rhodes, vol. 1, pp. 99-102, 1997.

10. K. Shinoda and C.-H. Lee, "Structural MAP speaker adaptation using hierarchical priors," *Proc. IEEE Workshop on Speech Recognition and Understanding*, Santa Barbara, pp. 381-387, 1997.

11. K. Shinoda and C.-H. Lee, "Unsupervised adaptation using structural Bayes approach," in *Proc. ICASSP-98*, Seattle, pp. II 793-796, 1998.